Università degli Studi di Bologna

Dipartimento di Elettronica Informatica e Sistemistica

Dottorato di Ricerca in Ingegneria Elettronica, Informatica
e delle Telecomunicazioni - XIX Ciclo

SSD: ING-INF/03 - Telecomunicazioni

Joint Source and Channel Coding

(Codifica Congiunta di Sorgente/Canale)

Relatore:
Chiar.mo Prof. Ing. Marco Chiani

Tesi di:
Ing. Matteo Mazzotti

Coordinatore:
Chiar.mo Prof. Ing. Paolo Bassi

Anno Accademico 2006/2007

To my family
and Ornella

Contents

List of Tables

List of Figures

Acronyms

AEP asymptotic equipartition property

App application layer

AWGN additive white Gaussian noise

BC broadcast channel

BCJR Bahl-Cocke-Jelinek-Raviv

BER bit error rate

BSC binary symmetric channel

BP belief propagation

BPSK binary phase shift keying

CC convolutional code

c.d.f. cumulative distribution function

CIF common intermediate format

CRC cyclic redundancy code

CSI channel state information

DCT discrete cosine transform

DPC dirty paper coding

DRI decision reliability information

DTC dirty tape coding

DVB Digital Video Broadcasting

EEP equal error protection

ETSI European Telecommunication Standards Institute

FEC forward error correction

FER frame error rate

FFT fast Fourier transform

FPT-UEP fixed-length packet transcoding unequal error protection

FSM finite state machine

GOP group of pictures

GOV group of video objects

HVS human visual systems

JSCC joint source and channel coding

JSCC/D joint source and channel co-decoding

KKT Karush-Kuhn-Tucker

IFFT inverse fast Fourier transform

i.i.d. independent, identically distributed

IP internet protocol

IPv6 internet protocol version six

IRA irregular repeat-accumulate

ISI inter-symbol interference

LDPC low density parity check

LDPCC low density parity check code

LHS left hand side

LLR log-likelihood ratio

MAP maximum a posteriori

MIMO multiple-input/multiple-output

ML maximum-likelihood

MLSE maximum-likelihood sequence estimation

MMSE minimum mean square error

MSE mean square error

NIR noise-to-interference ratio

NoSI noisy side information

NSI network state information

OFDM orthogonal frequency division multiplexing

p.d.f. probability density function

Phy physical layer

PLR packet loss rate

PMF probability mass function

PSNR peak signal-to-noise ratio

P-UEP proportional unequal error protection

QAM quadrature amplitude modulation

QoS quality of service

QPSK quadrature phase shift keying

R-CSI reduced channel state information

RCPC rate compatible punctured convolutional

RCPCC rate compatible punctured convolutional code

RCPRSC rate compatible punctured recursive systematic convolutional

RCPRSCC rate compatible punctured recursive systematic convolutional code

RHS right hand side

RMSE root mean square error

RoHC robust header compression

RTP real time protocol

RTT round trip time

r.v. random variable

SAI source a-posteriori information

SC start code

SI side information

SNR signal-to-noise ratio

SRI source a-priori information

SSI source significance information

SSIM structural similarity index

UDP user datagram protocol

UEP unequal error protection

VA Viterbi algorithm

VO video object

VOL video object layer

VOP video object plane

VP video packet

VQ vector quantizer

WCDMA wide-band code division multiple access

WDP writing on dirty paper

WLAN wireless local area network

Chapter 1

Introduction

O tempora! O mores!

In the recent years the historical separation between the application and the physical world, supported by Shannon's famous *Separation Theorem*, has been overcome. The spreading of applications highly demanding in terms of resources, like multimedia communications, is leading to joint-design approaches, capable of maximizing the final quality of the connections.

Particularly with wireless video communications, the constraints on the maximum allowed delays become critical elements, as well as the high sensitivity of the compressed bitstream to errors occurred along the radio channel. Our work, following the already known joint source and channel co-decoding (JSCC/D) approach, aims at developing strategies where the compression, protection and transmission parameters are jointly determined to yield the best end-to-end system performance. This joint approach requires the transmission of side information about the source and the channel state through the network nodes and the protocol layers within the nodes. Thus a realistic performance evaluation of such techniques should take these aspects into account, although this issue has been in general neglected in the literature.

In our work the definition and management of the side information exchanged among the chain blocks are realistically addressed, based on the key concepts of *Network Transparency* and *Cross Layer design*, which may be summarized by the phrase *awareness vs unawareness*. All the entities involved in the JSCC approach must share some sort of awareness, intended with a double meaning: awareness of the nature of the data they are transmitting and awareness of the state of the communication channel they are using. On the contrary, all the entities between the source encoder and the decoder which are not directly involved in the JSCC strategy should be completely unaware (or transparent) of the side information they are transmitting.

Within the *Phoenix* project, we address the problem of JSCC introducing two novel blocks, called Application JSCC Controller and Physical JSCC Controller. Basically, they constitute the units which collect the signalling information about the network and radio channel states, as well as the quality feedback from the source decoder, and decide the optimal transmission strategy.

Dirty Paper and Dirty Tape Coding are channel coding techniques capable of maximize the throughput of broadcast multi-user scenarios. Thus they constitute important elements which may help the JSCC Controllers in the efficient exploitation of the radio resources available. Moreover, they are strictly connected to the problem of information embedding and watermarking, which may be good candidates to build the distortion metrics implemented by the JSCC adaptive algorithms.

In the Chapter 1 we introduce the problem of JSCC and the *Phoenix* project. The model and the adaptive algorithms developed for the App and Phy JSCC Controller are reported in Chapter 2 and 3, respectively. Chapter 4 shows some results obtained with the proposed techniques in realistic communication scenarios. Chapter 5 introduces the problem of Dirty Paper and Dirty Tape Coding. A simple scheme for Dirty Paper Coding is then presented in Chapter 6, while in Chapter 7 we address the problem of binary Dirty Paper/Tape Coding with additional noisy side information available at the decoder.

Chapter 2

Joint Source and Channel Coding

Traditionally, the source coding and the dual operation of channel coding have been investigated and developed separately. This approach is mainly due to the famous *Separation Theorem* by Shannon [74]. Considering stationary and memoryless channels, he proved that an optimal coding approach can separately cope with the problem of reducing the redundancy in a source stream and with the problem of adding a proper protection to the compressed sequence. Clearly a separated scheme is a sub-case of a more general joint-coding strategy, but the remarkable result by Shannon showed that we do not incur in a loss of optimality if we restrict our attention to the first case.

The *separation theorem* relies on some hypothesis which are not always satisfied in real communication scenarios. First of all, the optimality expressed by Shannon is to be intended in an asymptotical sense, i.e. with infinitely long sequences coming from the source and entering the encoders. Real data transmissions must always respect fixed delay and complexity constraints. In the second place the bitstream generated by real source encoders is not uniformly significant: certain bits are inevitably more important than other, and so a uniform protection offered by the channel encoder may become a sub-optimal solution. This somehow simplified approach led to the development of extremely efficient source coding standards (e.g. MPEG-4 and H.264 for video sources) and to the design of powerful channel codes (e.g. Turbo and low density parity check (LDPC) codes), but, recently, the separation paradigm is being overcome. The innovative concepts of cross-layer design and network transparency are more and more spreading among researchers and engineers.

In particular, the evolution of wireless communications is towards a more integrated and global system, meeting the requirements of both users and industrial world, by offering convergence of different technologies and making efficient use of the existing and emerging ones. In order to meet this goal, an efficient and flexible allocation of the resources is necessary, based on co-operative optimization of the communication system components in the different layers. It is in this context that the *Phoenix* project was born.

2.1 The joint coding approach

Historically, the two dual encoding operations of source and channel coding have been kept separated from each other, following Shannon's well-known separation theorem [74] which states that source coding and channel coding can, asymptotically with the length of the source data, be designed separately without any loss of performance for the overall system. However, most of modern popular applications, such as audio/video streaming, do not meet such ideal hypotheses [55]. They indeed often require transmitting data with real-time constraints, operate on sources whose encoded data bit error sensitivity varies significantly and are designed to be as simple and low-power consuming as possible. For instance, variable-length codes (VLC), which are classically used in source coding for their compression capabilities, are often associated with Forward Error Correction (FEC) techniques which combat the effects of a real transmission channel (fading, noise, interference, etc.). It has however been shown that separation does not necessarily lead to the less complex solution [75], nor is always applicable [76]. In particular, joint source and channel coding (JSCC) can offer better solutions in wireless communications, offering a complexity reduction of the overall system while maintaining a good performance. Quite recently, JSCC techniques that include a co-ordination between source and channel encoders were investigated, and techniques were developed [77][78][79] that improve both encoding and decoding processes while keeping the overall complexity at an acceptable level [80].

2.2 The *Phoenix* project

The *Phoenix* project is a IST-FP6 European project [62], started in January 2004 and ended in December 2006. The aim of the PHOENIX project was to develop a scheme offering the possibility to let the application world (source coding, ciphering) and the transmission world (channel coding, modulation) to talk to each other over an internet protocol version six (IPv6) protocol stack (network world), so that they can jointly develop an end-to-end optimized wireless communication link. To reach this goal, the three

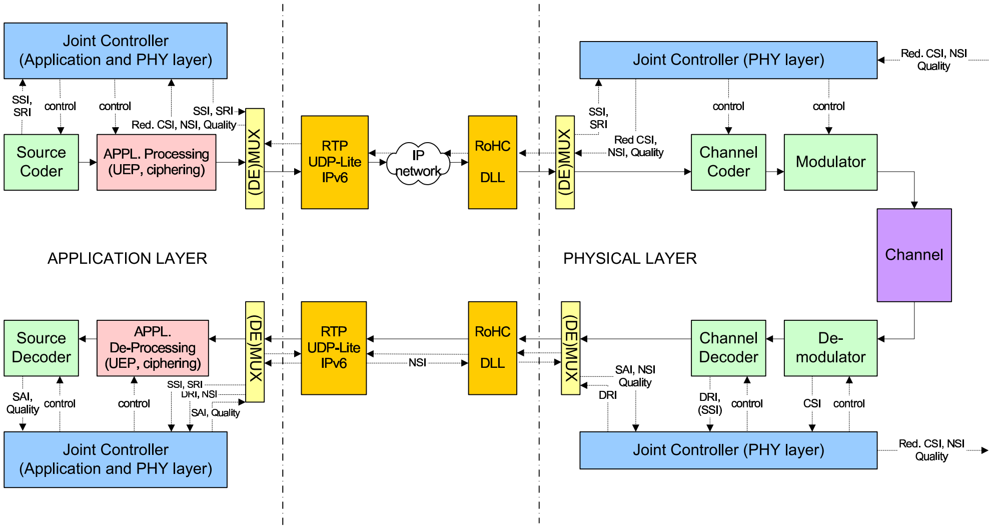


Figure 2.1: The communication chain proposed by the Phoenix consortium.

following main axes have been pursued:

- development of innovative schemes to enable end-to-end joint optimization over wireless links: flexible channel coding and modulation schemes, adaptation and development of source coding schemes with respect to their ability for JSCC/D,
- establishment of efficient and adaptive optimization strategies jointly controlling the coding blocks and realistically taking into account the system limitations,
- integration of those techniques in a global network architecture including the development of a cross-layer design approach which will allow to apply the optimization strategies in any kind internet protocol (IP)-based network.

To guarantee the quality of service (QoS) requested by the end-user, an end-to-end management of the connection must be undertaken, that will guarantee the exchanges of the QoS parameters between the different sections of the end-to-end link and allow for optimized high data rate transmissions. The optimization technique relies on the blocks called application layer (App) and physical layer (Phy) JSCC Controller.

The *Phoenix* consortium was made of 7 European partners:

- Thales (France),
- CNIT/University of Bologna (Italy),

- Cefriel (Italy),
- University of Budapest (Hungary),
- VTT (Finland),
- Wind (Italy),
- University of Southampton (United Kingdom).

Fig. 2.1 presents the overall architecture considered by the project, as it was defined in [63]. The considered transmission chain contains all the main blocks normally present in a video communication scenario. In particular, three sections have been identified: the application layer, the network and the physical layer.

2.3 Network Transparency and Cross-layer design

Network transparency is a key concept in developing JSCC/D techniques [64]. The goal of network transparency is twofold: to allow the communication between the different entities of the network involved the joint optimization approach and to make the correspondent communication transparent to all the non JSCC-aware nodes. Thus, it enhances the network communication flexibility, while permitting to a generic network element to be non-aware of non-standard functionalities, like, for our specific purposes, the JSCC controls and signalling information. Clearly, this kind of flexibility is a fundamental requirement in order to open the way towards a wide deployment of joint source and channel co-decoding techniques in the next generation wireless networks. The basic mechanisms of network transparency are two: cross-layer communications and network transparent communications. In fact, the entities communicating through the network can be different nodes but also different layers within the same node. The recent paradigm of cross-layer design is to be intended in this way: it does not imply the integration of all the protocol layers, nor their elimination. A cross-layer design constitutes the overcoming of a sometime excessively rigid protocol separation, in the direction of an holistic approach to wireless networking.

The mechanisms to transmit the control/signalling information may be different and they can be more or less appropriate, according to the specific type of information considered. For example, side information that are strictly coupled with the multimedia data are more likely to be in-band delivered, while for the others an out-of-band protocol may be preferable; furthermore, proper extensions to already existing standard protocols are to be employed to provide efficient communication between different nodes, while inter-layer signalling should require the development of new APIs.

2.4 Side information flow

The basic concept of a JSCC technique is the joint optimization of the source coder, channel coder and modulator at the transmitting side, as well as of the demodulator, channel decoder and source decoder at the receiving side. To achieve this goal it is required to exchange several side information across both different network entities and through the stack of layers within each entity. The signals among the blocks involved in the communication task can be divided in control signals, side information and feedback information [77].

Several information can be useful to dynamically configure communication chain, run-time adapting the configuration of the blocks on the basis of the actual condition of the wireless channel(s), the available network resources in the wired infrastructure, the source and channel codecs options, the modulation schemes, the terminal capabilities and others, in order to maximize the QoS perceived by the end-user (or, alternatively, in order to decrease the employed transmission resources).

This side information-aided approach is already known in the literature, see e.g. [77], but often the hypothesis underlying these techniques do not consider how to really implement the suggested solutions. Within the *Phoenix* project, we have tried to identify realistic information exchanging mechanisms, easily implementable in the current or in the the next generation architectures. Here we present a list of control/signalling information that have been considered in the framework of the *Phoenix* project [65][64].

The structure, size and rate of each type of information depend on its nature and has been taken into account during the project development.

As already anticipated, some controls and signals travel along the same path of the coded video stream, while others on the reverse one. An underlying hypothesis is that the two paths are exactly the same, at least for what concerns the wireless hops; however, this is normally true, considering that the routing protocols commonly used (e.g. OSPF, RIP, IS-IS) are based on the lowest cost metric that do not change with the direction between two end-points. More likely the paths are coincident when considering data flow in the same direction. Some side information must be even synchronized with the video contents they refer. Anyway, we usually assume that this hypothesis holds for both the direct and the reverse link, so that also the feedback information can be considered available to all the crossed nodes.

The side information are:

- Source Significance Information (SSI). They are generated by the source coder and reflect the importance of the different parts composing the media stream. In particular the SSI should correspond to the data significance as it is perceived by the human visual systems (HVS). The limits of the human visual perception are the basis of the lossy coding effectiveness, allowing high data compression with minimal distortion; nevertheless, some basic video information are very important for the decoding task and a successful delivering of them to the end-user is necessary. source significance information (SSI) are strictly related to the data stream and so they must be somehow synchronized with it, determining an overall overhead roughly proportional to the volume of video data sent.

  The SSI allow for discriminating different classes (or data partitions) within the same stream. As a consequence an unbalanced level of protection among the different priority layers becomes preferable every time the available resources are not able to guarantee a sufficiently reliable connection with a traditional approach. This is particularly true when one or more error prone channels, e.g. the wireless links, are present along the data path. In Chapter 4 we will show how this task can be performed through proper channel codes, e.g. rate compatible punctured convolutional (RCPC) and LDPC codes. Through an unequal error protection (UEP) policy it is possible to increase the protection of critical data by allocating more redundancy with respect to less important information.

  Within the *Phoenix* project several UEP schemes have been tested, working both at the application layer or at the physical layer. While the first solution has the advantage to be more complaint to existing transmission standards, like UMTS, 802.11 or WiMax, it has the main drawback of a bit rate increment produced at the application layer. Thus we have focused our attention to the latter case, of UEP performed at the physical layer. A UEP strategy becomes even more effective if combined with the error resilience tools provided by the most recent source coding standards, like MPEG-4 or H.264. Even error concealment strategies provide better result if, at least, the most important information are correctly delivered to the source decoder.

- Channel State Information (CSI). They constitute information on the conditions of the wireless channel(s), fed-back from the radio receiver to the radio transmitter. They are basically constituted by the average signal-to-noise ratio (SNR), the estimated coherence time, etc. In a multiple-input/multiple-output (MIMO) architecture they often include the feedback information about the channel matrix, while for a orthogonal frequency division multiplexing (OFDM) modulation it may contain the SNRs of all the subcarriers. The rate of CSI transmission depends on the channel characteristics and a good trade-off between the accuracy of the channel knowledge and the volume of feedback information should be addressed. Anyway, differently from the reduced channel state information (R-CSI), the communication of the channel state information (CSI) affects only the radio terminals, so that a good knowledge of the channel state is a reasonable assumption also for the transmitter. This hypothesis is particularly important for the operation of the Phy JSCC Controller, because, as it will be shown Chapter 4, it usually needs detailed CSI. CSI goes in the reverse path with respect to the video data packets, hence it is not strictly synchronized with them.

- Reduced CSI (R-CSI). The application layer controller may use R-CSI from the feedback channel. The R-CSI are not necessarily the same as the CSI in physical layer. For example, channel conditions averaged over longer time intervals or in some cases only bit error rate (BER) information may be needed by the application controllers in order to adjust error resilience and UEP mode. In general, the R-CSI rate should be much lower than the video data rate and it may be considered almost negligible in terms of additional overhead.

- Decision Reliability Information (DRI). Beside the received video data flow, the stack layers at the receiver side may exchange information about the reliability of the stream. Usually DRI are constituted by soft information, like the log-likelihood ratios (LLRs) coming from the channel decoder and provided to the source decoder. In particular, the *Thales* within the *Phoenix* consortium suggested and developed a soft H.264 decoder capable of taking advantage of the LLRs coming from the radio receiver [69]. Clearly, the overhead introduce by the decision reliability information (DRI) can easily exceed the encoded video rate. Thus, the trade-off between the benefits they introduce and their cost must be accurately evaluated. Clearly, DRI must be strictly synchronized with the video stream.

Control signal	Suitable Mechanism	Results
SSI	IPv6 hop-by-hop or solutions as in [57]	Overhead of few Kbits/s (depending on the source coding rate); high synchronization with the video data.
CSI	ICMPv6	Overhead of less than 10 Kbyte/sec for CSI update frequencies up to 10 ms; slight synchronization with the video data.
NSI	ICMPv6	Low overhead with suitable frequency of 100 ms (less than 1 Kbyte/sec).
DRI/SAI	IPv6 packets	Very high bandwidth consuming (even higher than the video data flow of a fixed multiplying factor). Probably better to send these control signals only when the wireless receiver is also the data traffic destination.
SRI	IPv6 hop-by-hop or Destination	Overhead of few Kbytes/sec (depending on the source coding rate) and high synchronization with the video data.

Table 2.1: Control and signal information transmission mechanisms and overheads

- Source A-priori Information (SRI). They are further information produced by the the source encoder. The SRI contains information from the source known a-priori, like some statistical or deterministic properties of the produced video stream. Thus the App and Phy JSCC Controllers and the other entities involved in the JSCC approach can use the SRI to adjust QoS.
  source a-priori information (SRI) is synchronized with the associated video stream and the amount of introduced overhead depends heavily on the nature of the transmitted information.

- Source A-posteriori Information (SAI). These information result from the analysis

of the source decoding process; hence they are generated by the destination terminal and they follow the reverse data path. SAI can be exploited, for instance, by the channel decoder to improve its correcting performance.
As already stated, SAI travel from the destination terminal to transmitting nodes and so it so not strictly synchronized with the video packets. The related overhead varies according to the nature of the feedback information.

- Network State Information (NSI). NSI are feedback information about the availability of network resources along the direct data path and possibly also in the reverse direction. Such information can be represented by delay, jitter and packet loss rate (PLR). Thanks to these information the App and Phy JSCC Controllers know the load conditions of the network and possible congestion events. Within the *Phoenix* project we have actually assumed that the resource bottleneck is on the wireless interfaces, more than inside the IPv6 network. Anyway, also NSI can be effectively exploited by the controllers to tune the amount of the bit rate generated by the source coder.
  NSI flow towards the source terminal and they are not synchronized with the video stream. Their refreshment rate should be frequent enough to track the variations within the network state, without introducing excessive overhead.

- Video quality. It is constituted by a feedback information about the quality of the decoded video stream, coming from the source decoder and directed to the App JSCC Controller. It is a piece of information, which can help the controllers in determining the configuration of all the chain blocks in order to maximize the end-to-end QoS. In the following chapter we will describe in more details the problem of the quality metrics used to compute the video quality feedback.

In Table 2.1 we have reported some of the mechanisms defined by the *Phoenix* consortium [89][64] for the side information exchanges.

2.5 The Joint Source and Channel Co-Decoding Controllers

When dealing with JSCC strategies, two different approaches are possible:

- a distributed approach, where it is assumed that all the blocks know a sufficient amount of information to decide the best configuration parameters,

- a scheme where the *intelligence* is concentrated only in a few blocks, which should collect state information from all the chain, decide the optimal communication strategy and send the proper control signals to the other chain blocks.

Within the *Phoenix* project we opted for the second solution, realized through the introduction of the JSCC controllers [66][67].
The controllers play a key role in the system, being responsible of the optimization of the whole system chain. The task of the controllers is to be aware of the system state in different layers, exchange this state information with other system layers and jointly optimize different transmission parameters according to the system state in various layers. The system model includes state information about the source stream sensitivity, wireless channel state (CSI), network status (NSI), decisions reliability (DRI) and source a-priori and a-posteriori information (SRI and SAI). The parameters that may be adjusted according to the state information include source data rate or error resilience tuning, channel coding and modulation parameters and probability information that allow soft video stream decoding at the receiving terminal.
The controllers are located in logically and, often, also physically distinct points of the chain:

- the App JSCC Controller is at the application layer and it is basically designed to cope with long-term transmission conditions and impairments (both at the radio link level and in the network),
- the Phy JSCC Controller is at the physical layer and it is capable of faster reactions to changes in the wireless channel conditions.

A typical video communication session opens with a preliminary "handshaking" phase, during which further information is exchanged among the system blocks. In the session set-up, in fact, information about the characteristics of the system, such as, e.g., the type of channel encoders available and the achievable channel coding rates, the modulators and modulation modes available, is collected by the controllers. Figure 4.7 gives an example of session set-up, highlighting the information to be exchanged in this preliminary phase. During this phase also the user requirements are specified, according to the communication scenario considered.

After establishing the resources available within the various blocks of the chain, the App JSCC Controller sets the starting configuration for all the JSCC-aware nodes of the chain. The model that the *Phoenix* consortium has suggested and developed is based on a

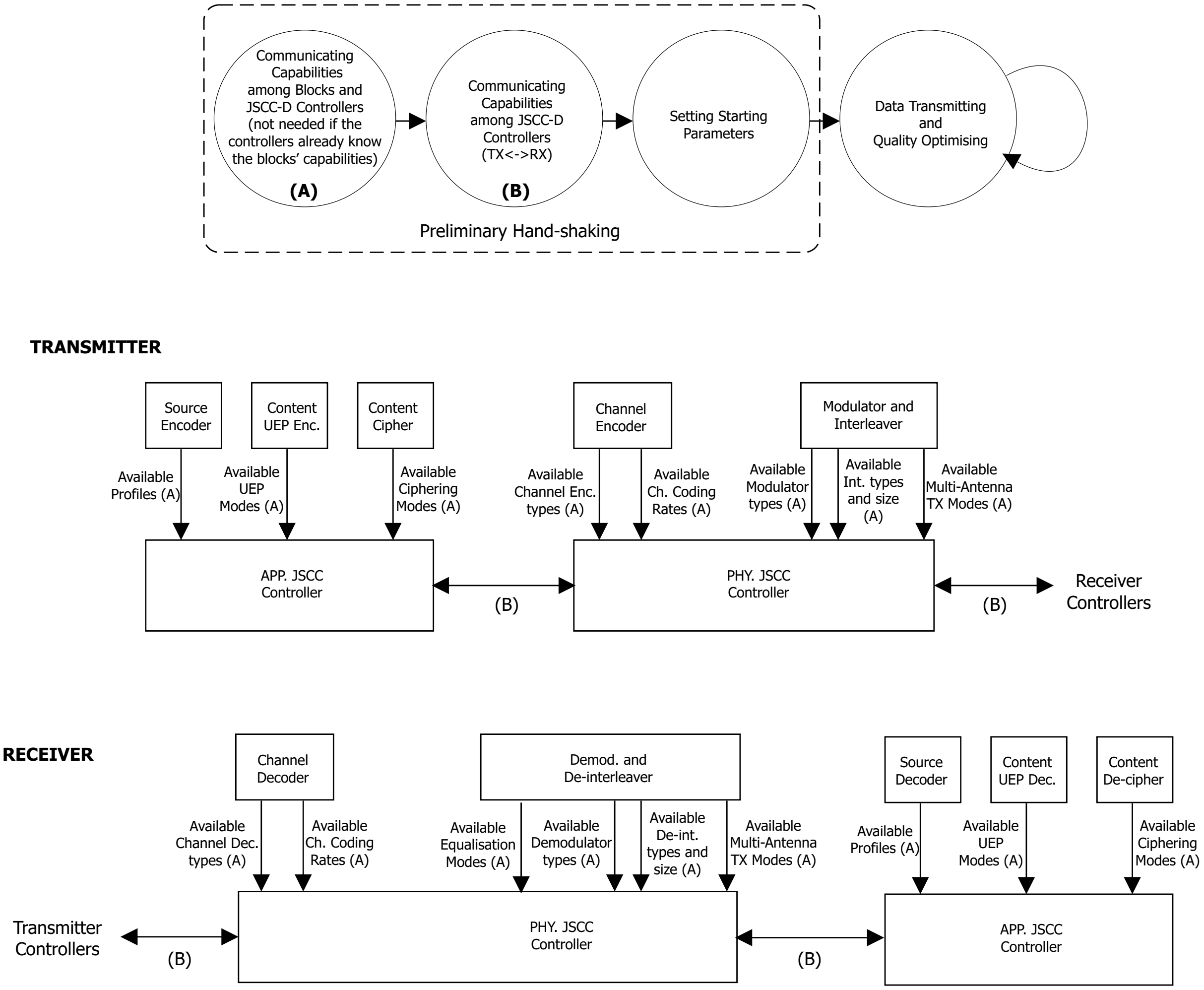

Communicating Capabilities among Blocks and JSCC-D Controllers (not needed if the controllers already know the blocks' capabilities)
(A)
Communicating Capabilities among JSCC-D Controllers (TX<->RX)
(B)
Setting Starting Parameters
Data Transmitting and Quality Optimising
Preliminary Hand-shaking
TRANSMITTER
Source Encoder
Content UEP Enc.
Content Cipher
Channel Encoder
Modulator and Interleaver
Available Profiles (A)
Available UEP Modes (A)
Available Ciphering Modes (A)
Available Channel Enc. types (A)
Available Ch. Coding Rates (A)
Available Modulator types (A)
Available Int. types and size (A)
Available Multi-Antenna TX Modes (A)
APP. JSCC Controller
(B)
PHY. JSCC Controller
(B)
Receiver Controllers
RECEIVER
Channel Decoder
Demod. and De-interleaver
Source Decoder
Content UEP Dec.
Content De-cipher
Available Channel Dec. types (A)
Available Ch. Coding Rates (A)
Available Equalisation Modes (A)
Available Demodulator types (A)
Available De-int. types and size (A)
Available Multi-Antenna TX Modes (A)
Available Profiles (A)
Available UEP Modes (A)
Available Ciphering Modes (A)
Transmitter Controllers
(B)
PHY. JSCC Controller
(B)
APP. JSCC Controller

Figure 2.2: Transmission system block diagram.

master-slave scheme: the App JSCC Controller collects the low-rate feedback information (NSI, R-CSI, quality information) and, according to the video sequence characteristics, sets the main parameters of the chain in order to optimize the end-to-end video quality. In particular, as reported in Fig. 2.3, it specifies some constraints to the Phy JSCC Controller, which, acting as the slave, tries to optimize the transmission over the radio channel while satisfying the constraint. In Fig. we have indicated with S_i^{APP} the set of parameters determined by the App JSCC Controller during the i^{th} time step and with $S_{k,i}^{PHY}$ the set of parameters set by the Phy JSCC Controller for the k^{th} packet transmitted during the i^{th} time interval. The Phy JSCC Controller knows more accurate CSI and thus can react more precisely and more quickly.

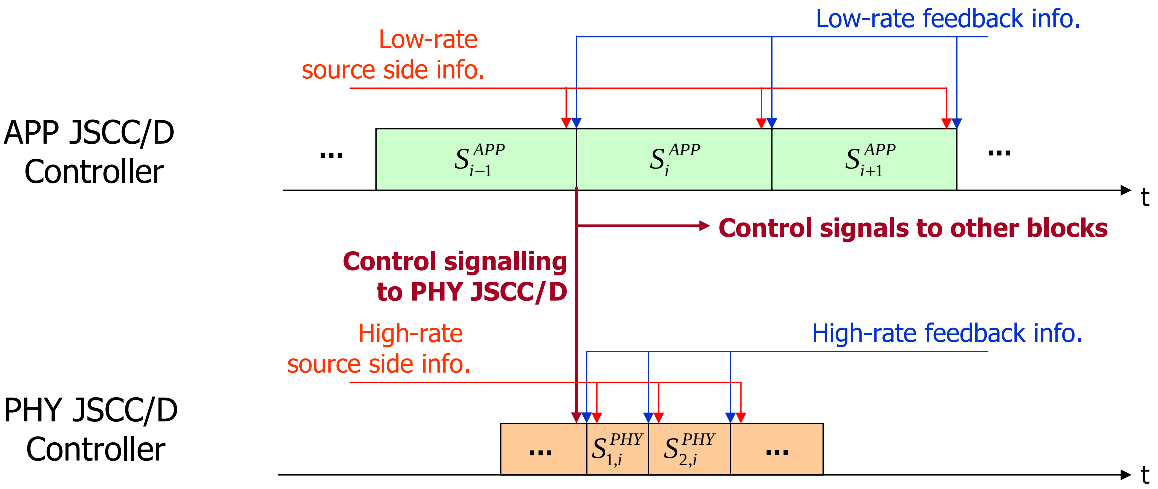


Figure 2.3: Scheme representing the JSCC signalling.

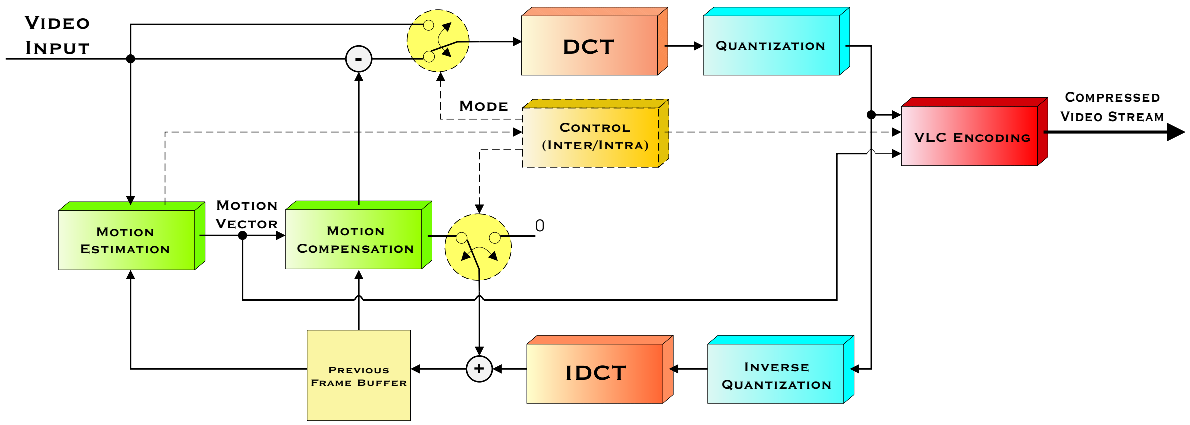


Figure 2.4: Encoder MPEG.

2.6 The source coder

As source coding, we have mainly considered the MPEG-4 standard and only recently we started moving to H.264. This choice was principally motivated by one practical reason: to obtain the results of our joint coding techniques we needed to insert in the implemented simulation chain a source co-decoder which were robust against packet loss and bit errors occurred during the transmission through the radio channel. In other words, we needed a source co-decoder capable of recognizing a corrupted syntax and continuing working according to a *best effort* policy. At the moment there are not many co-decoders satisfying these requirements: usually the available software starts from the assumption that the received data are correct and consider only the possibility of some packet loss. For this reason we opted for a particular MPEG-4 co-decoding software developed during the European project MoMuSys [59], with some changes and improvements made by *Philips Research*, and specifically designed to deal with corrupted bitstreams.

2.7 MPEG-4 coding

MPEG-4 is one of the most recent MPEG ISO/IEC standard for video compression [48][49][28][50]. In Fig. 2.5 we have schematically illustrated an MPEG encoder. The main coding steps are easily identifiable from the figure:

- block decomposition;
- motion estimation/compensation;

- discrete cosine transform (DCT);
- quantization;
- run-length coding and entropy coding.

The MPEG-4 standard utilizes the concept of object-based coding, allowing interactivity, and layered coding. The MPEG-4 bitstream is basically structured in video objects (VOs), video object layers (VOLs), i.e. the information related to an object in a scalability layer, video object planes (VOPs), i.e. the instance of an object in a frame and, optionally, group of video objectss (GOVs) and video packets (VPs). Just like most video compression standards, it extensively relies on prediction and entropy coding and it is consequently very sensitive to channel errors.

With the goal of transmission over error prone channels, some error resilience tools have been added to the MPEG-4 standard [51]: Reversible Variable Length Codes (RVLC), Header Extension Codes (HEC), resync markers and data partitioning help in adding robustness to the MPEG-4 bitstream. With the use of Resync markers, the MPEG-4 bitstream results composed of packets which are almost of the same length and separated by start codes, unique words recognizable from any sequence of variable length codewords, but not robust to channel errors. The *data partitioning* tool allows the separation of data with different significance within the packet.

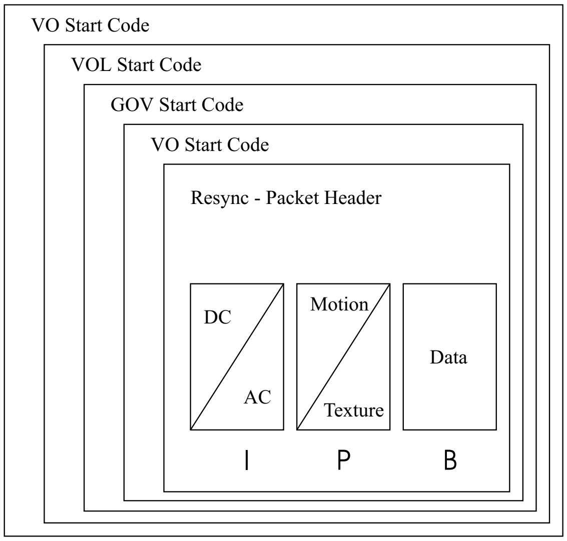


Figure 2.5: MPEG-4 hierarchical structure.

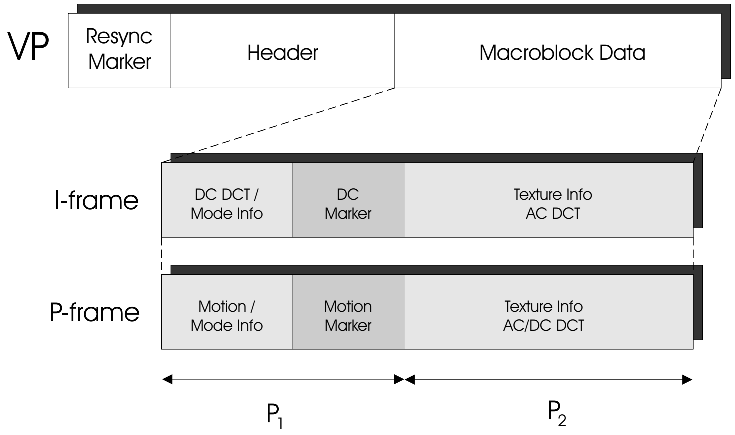


Figure 2.6: MPEG-4 data partitioning.

Thanks to the tools of re-synchronization and data partitioning, the bitstream results logically organized in an hierarchical structure [56], as reported in Fig. 2.5. The elementary packets transmitted are called VPs and they contain the coded description of the various blocks and macro-blocks of pixels in which each frame has been decomposed. A proper number of VPs forms a VOP, that, for sequences containing a single VO, coincides with a frame. Each VOP of the sequence is coded either in *intra* mode (frame I), i.e. without any reference to previous images, or in *inter* mode, i.e. differentially predicted from the previous frame (P frame) or from a combination of the previous and the following frames (B frame). A group of VOPs, in turn, constitutes a GOV. Finally, at a logical upper level, there are the VOLs and the VOs.

Regardless of the error resilience tools, MPEG-4 video transmission over wireless channels is still critical: for this reason, studies aimed at efficiently transmitting MPEG-4 video over wireless channels are currently being performed [58].

If properly exploited, error resilience tools can produce a further improvement of the received video quality. In particular, the data partitioning tool can be usefully exploited with the purpose of performing UEP: information bits contained in each packet are split into three partitions, each of which has a different sensitivity to channel errors. As shown in Fig. 2.6, for I frames partitions consist of a header, DC DCT coefficients and AC DCT coefficients, separated by a marker. As far as P frames are concerned, partitions consist of a header, a motion partition, containing data for the motion compensation, thus very sensitive to channel errors, and a texture partition, separated by a marker. As it will be shown in Chapter 4, the data partitioning tool may thus be exploited to perform unequal error protection, both at channel coding and at modulation level.

Chapter 3

Application JSCC Controller

The tasks performed by the App JSCC Controller at the transmitter include tuning the video coding parameters according to the system state and application requirements. In particular, it contributes to initialize the video transmission by selecting configuration parameters in accordance to the delay requirements, the target image quality, and the resources available at the radio link and through the IPv6 network [66][67]. During the video transmission, the task of the controller is to monitor the quality of the communication, checking the periodically received feedback information. As a result, the data rate of the video stream is continuously adapted to the network and wireless channel conditions. The main goals of the App JSCC Controller are to prevent network congestion, support fair resource sharing among different users and allow to give some guidance to an efficient exploitation of the wireless resources. In addition, the controller may also adjust the error resilience tools applied to the video stream, according to the quality of the overall communication. In fact, these tools, completely useless with error free connections, lead to sensible improvements in the perceived final quality.

3.1 Finite state machine model

The App JSCC Controller has been modelled as a finite state machine (FSM) [84]: according to the input information collected from the system, the state (represented by source frame rate, quantization parameters, group of pictures (GOP) length,...) may be modified at each controller time step. The controlling step duration can be chosen according to the scenario. In the selection of the controller step, the channel coherence time should be considered, since the channel conditions should possibly be constant over one controlling step. In the case of wireless channel affected by fading and shadowing, we consider thus a controlling step lower than the shadowing channel coherence time. On the other side, the time step had to be long enough to allow source adaptation frame by frame. If f_r is the selected frame rate, $T_{ch,sh}$ is the channel coherence time with respect to the slow fading (shadowing) component and T_a is the App JSCC Controller time step, our adaptation technique requires that the following inequality is satisfied:

$$\frac{1}{f_r} < T_s \ll T_{ch,sh}. \tag{3.1}$$

According to these observations, we found that a time step $T_a = 1\ s$ is usually enough to obtain good results, without drastically increasing the side information flow through the layer stack. This FSM model can also be applied for source coding different from MPEG-4: for instance, within the Phoenix project we have considered the case of H.264 co-decoders controlled by a similar FSM.

At the beginning of each iteration cycle (roughly corresponding to one or two GOPs), the controller decides the next operating state, which is defined by a fixed set of configuration parameters for the different blocks of the chain. According to these values, the App JSCC Controller updates the configuration parameters for the different blocks of the communication chain. The choice of the new state is based on the history of the previous states and on the feedback information coming from the receiver side, clearly relevant to the previous cycle.

The main feedback information required by the adapting algorithm are:

- quality: peak signal-to-noise ratio (PSNR) or other quality metrics, e.g. based on structural distortion [52] or other solution not dependent on the knowledge of the original sequence;
- reduced CSI: average SNR measured one controller step, channel coherence time;
- NSI: number of lost packets, average jitter, round trip time (RTT).

The most important configuration parameters set by the App JSCC Controller and modifiable at each simulation step are:

- video encoder frame rate;
- quantization parameters;
- GOP size;

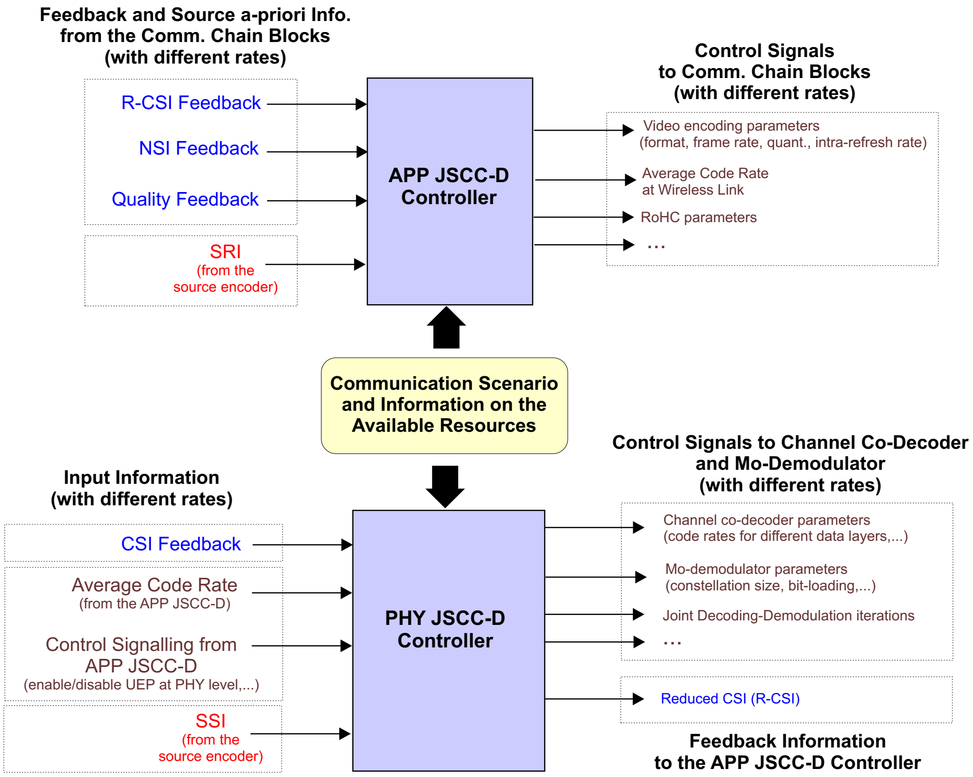

Feedback and Source a-priori Info. from the Comm. Chain Blocks (with different rates)
R-CSI Feedback
NSI Feedback
Quality Feedback
SRI (from the source encoder)
APP JSCC-D Controller
Control Signals to Comm. Chain Blocks (with different rates)
Video encoding parameters (format, frame rate, quant., intra-refresh rate)
Average Code Rate at Wireless Link
RoHC parameters
...
Communication Scenario and Information on the Available Resources
Input Information (with different rates)
CSI Feedback
Average Code Rate (from the APP JSCC-D)
Control Signalling from APP JSCC-D (enable/disable UEP at PHY level,...)
SSI (from the source encoder)
PHY JSCC-D Controller
Control Signals to Channel Co-Decoder and Mo-Demodulator (with different rates)
Channel co-decoder parameters (code rates for different data layers,...)
Mo-demodulator parameters (constellation size, bit-loading,...)
Joint Decoding-Demodulation iterations
...
Reduced CSI (R-CSI)
Feedback Information to the APP JSCC-D Controller

Figure 3.1: Input and output signals for the JSCC Controllers.

- average channel code rate, computed from the source encoding parameters and the knowledge of the available bandwidth.

A simple scheme summarizing the input and the output signals for the App JSCC Controller is reported in Fig. 3.1.

In order to reduce the total number of possible configurations and to avoid switching continuously from one given set of parameters to another, which would not be efficient in terms of compression for any source encoder, the controller takes into account in its practical implementation only a limited set of possibilities for these parameters. In particular we consider frame rates of 30, 15 and 7.5 fps, QCIF/CIF spatial resolutions and GOP lengths assuming values of 8, 15 and 30 frames. Different but limited sets of quantization parameters (qI,qP) have been determined for the various sequences considered. Furthermore, some constraints on these configuration parameters must be satisfied, so that the overall number of controller states is reduced. For instance, we impose that the video data in each time step is independently decodable by the source decoder: e.g., with a frame rate of 15 fps, only GOPs of 8 and 15 frames can be selected.

Let us denote with B_r the average bit rate output from the source encoder. If R_S is the average source coding rate in [bit/pixel], we have $B_r = MQf_rR_S$, where M and Q are the frame width and height, respectively, and f_r is the frame rate in [frame/s]. Given the bit rate of the chosen state, the average code rate to apply for data protection along the radio channel is evaluated considering the total bandwidth available for the communication, B_{max}. This parameter is specified by the Phy JSCC Controller to the App JSCC Controller during the preliminary handshaking phase and it is exploited to compute the constraint on the average code rate which must be satisfied by the Phy JSCC Controller optimization task (see Chapter 4 for more details):

$$R_C = \frac{B_{max}}{B_r} \tag{3.2}$$

That Rc target information is used either for embedded unequal error protection at the application level [67][54] or provided directly to the physical layer controller. If physical layer UEP is adopted, Rc will constitute the basis of the optimal code rate selection algorithm. Of course the knowledge of the bit rate is only approximated and it is based on proper rate/source parameters models or average values evaluated in previous controller steps. An example of the source characterizations we realized is reported in section 3.3. As an example, for the first and second scenarios five different states have been chosen for the App JSCC Controller with the *foreman* test sequence. State 1 corresponds to the lowest source data rate (lowest video quality) and highest robustness, whereas state 5 corresponds to the highest source data rate (highest video quality) and lowest robustness. Thus, increasing the state number means increasing the robustness transmission at cost of a loss in the error free received video quality. Fig. 3.2 depicts the finite state machine describing the App-JSCC controller with the allowed transitions among states [84]. More precisely, the number of possible parameter sets is seven, since state 3 and state 5 have two different options for the GOP length. The choice of the GOP length is made according to the channel conditions: for average channel SNR below a prefixed threshold the lower one is chosen, whereas for higher values of SNR the higher value is preferred.

3.2 Adaptive Algorithm

Several adaptive algorithms have been developed and tested during the *Phoenix* project. Here we present an algorithm which has been developed and tested with MPEG-4 coded sequences. It basically takes into account the trend of the video quality feedback from the source decoder. Typically, a low video quality value associated to a negative quality

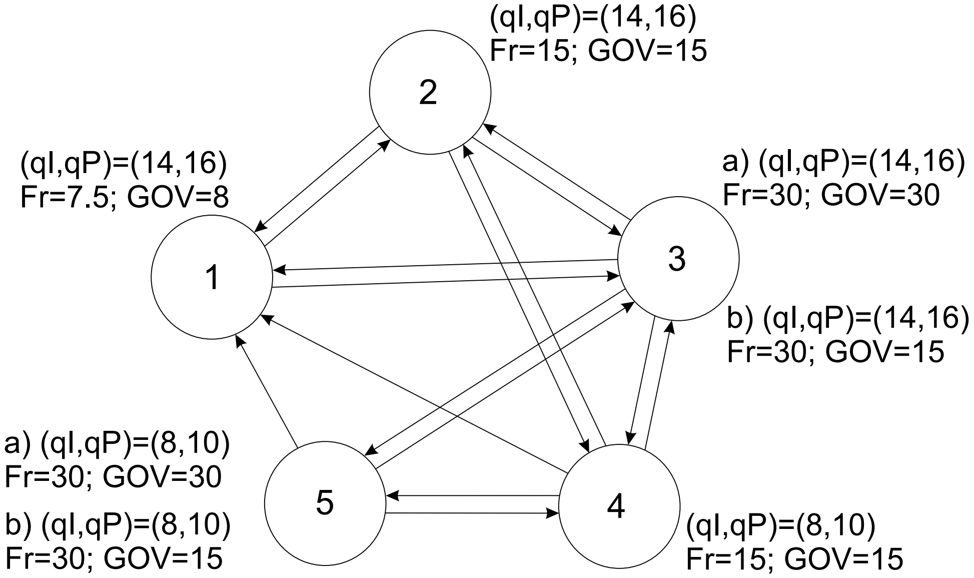


Figure 3.2: Example of FSM model and allowed transitions among states.

trend will cause a transition to a state characterized by higher robustness. When there is a network congestion, indicated by an high value for the PLR feedback in the NSI, the controller turns immediately to state 1, characterized by the lowest source bit rate, in order to reduce as much as possible the amount of data which have to flow through the IPv6 network. This choice contributes to a rapid resolution of the possible network congestion which may have determined the packet loss events.

The adaptation algorithm has been schematically reported in Fig. 3.3. Different approaches have been developed within the *Phoenix* project by our partners (often specific for different source coders) and validated through the communication chain. In Fig. 3.3 we have adopted the following notation:

- s_j is the j^{th} configuration parameter of the communication chain;
- $S_i^{APP} = \{s_j\}$ represents the set of parameters determined by the App JSCC Controller during the i^{th} time step;
- QI_i, $RCSI_i$ and NSI_i are the quality feedback information, the reduced CSI and the NSI received at the beginning of the i^{th} time step and so relevant to the $(i-t)^{th}$ cycle;
- Thr_{PLR} and Thr_{QI} are proper threshold values;

- cr_{SNR} is used to keep track of critical values of R-CSI.

Basically, the App JSCC Controller is based on an adaptive algorithm with memory, based on several thresholds values. In particular, cr_{SNR} store the channel conditions corresponding to a sudden drop in terms of quality, preventing the FSM from selecting states characterized by a low protection level: a state increment, corresponding to a higher error-free quality, will be allowed only after an SNR improvement. The memory approach here described may be easily extended to include some learning mechanisms helping in determining the best working state, given all the feedback information. These improvements are currently under investigation. Apart from the PLR_i, the future state is determined through $f_1(Q_1)$ (just for the second cycle) and $f_j(QI_i, state_i)$ with $j = 2, 3, 4$, which, basically, determine the state number increment/decrement based on the last received QI_i.

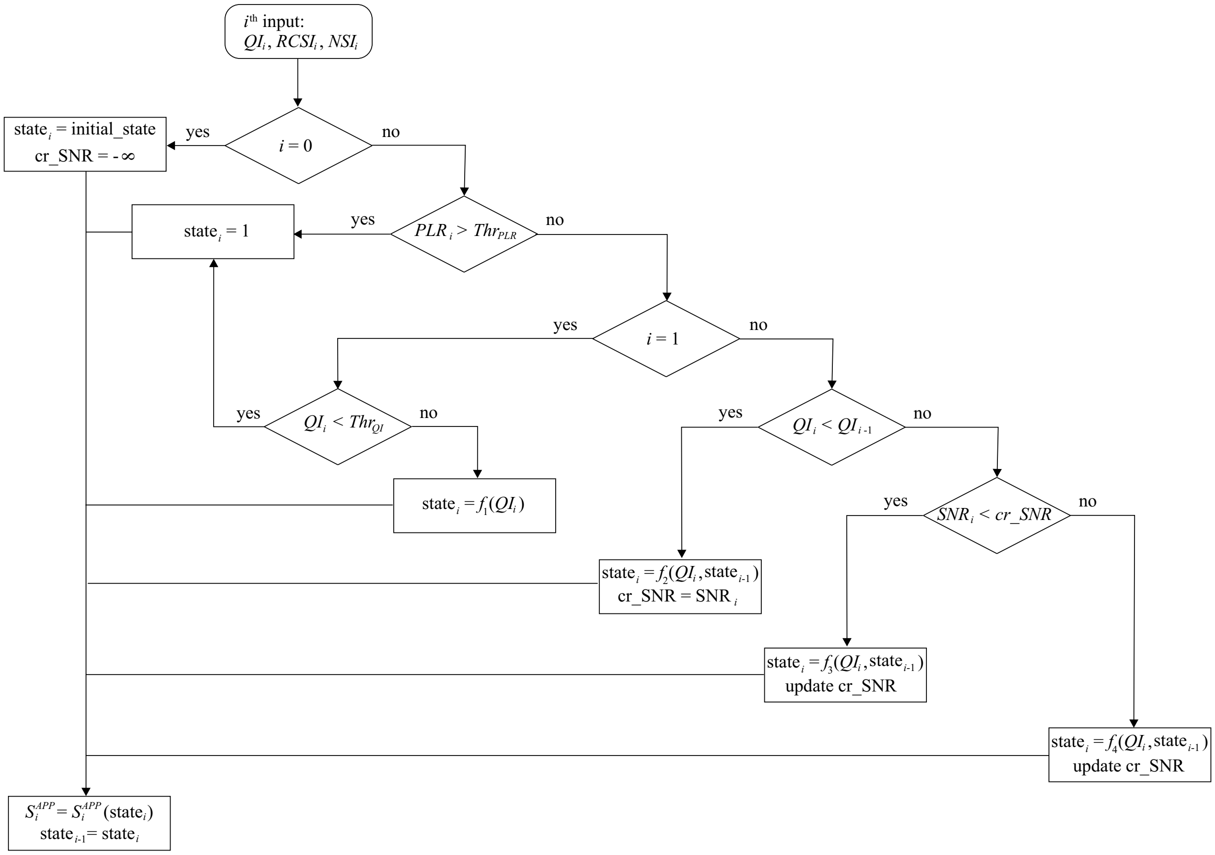


Figure 3.3: An App JSCC Controller adaptive algorithm based on FSM.

3.3 Source characterization

In order to define the states for the App JSCC Controller we have performed several simulative campaign with video sequences characterized by different format, frame rate and motion/texture features. As an example, in Figures 3.6-3.7 we have reported the PSNR and bit rate surfaces for varying quantization scales qI and qP. The original video sequence is the *Foreman* sequence in common intermediate format (CIF) format at 30 fps. We have taken into account three different frame rates (30, 15 and 7.5 fps) and three GOP sizes (30, 15 and 8 frames). The bit rate has been measured in two points of the communication chain: after the source encoder and before the channel encoder. In the latter case we have distinguished between a scheme implementing robust header compression (RoHC) and a traditional scheme.

Starting from these surfaces, we have identified several states characterized by different quality and bit rate. The criteria which led us in the state selection are basically:

- the states have to span a wide range of target qualities and bit rates;
- the states can be ordered in terms of increasing quality, so that the App JSCC Controller knows that state $i + 1$, when available, is always preferable to state i, under ideal transmission conditions.

The number of identified states changes according to the video format and the statistical characteristics of the sequence, but the algorithm applied by the App JSCC Controller is basically the same.

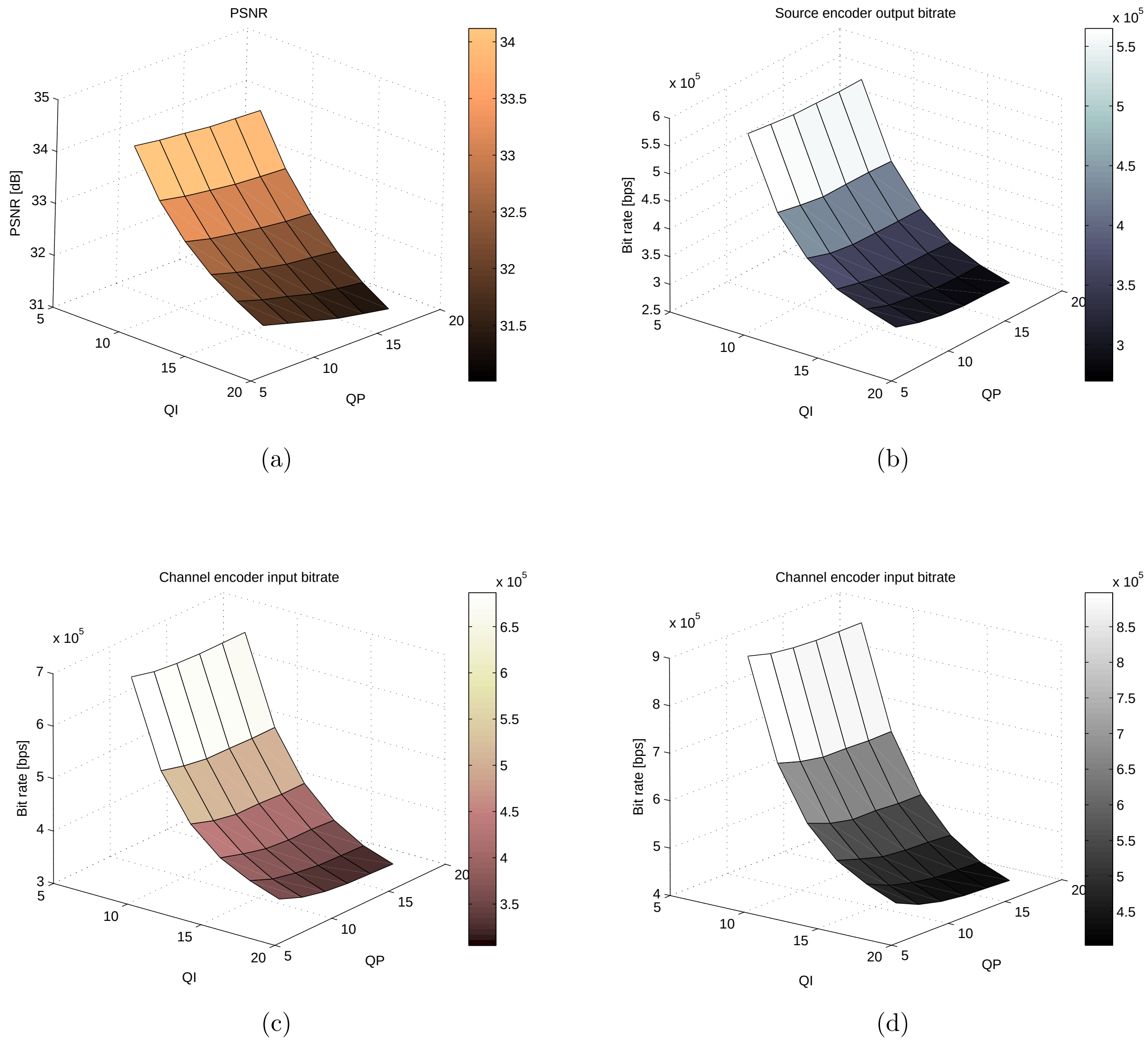
The sequences we have considered to validate our joint coding techniques are:

- *foreman* in CIF and QCIF format;
- *akiyo* in QCIF format;
- *phoenix* in CIF format.

Moreover, as it will be described in Chapter 4, some echocardiographic sequences have been considered when dealing with bio-medical applications.

3.4 Quality metrics

The evaluation techniques of video quality can be classified in two main families:



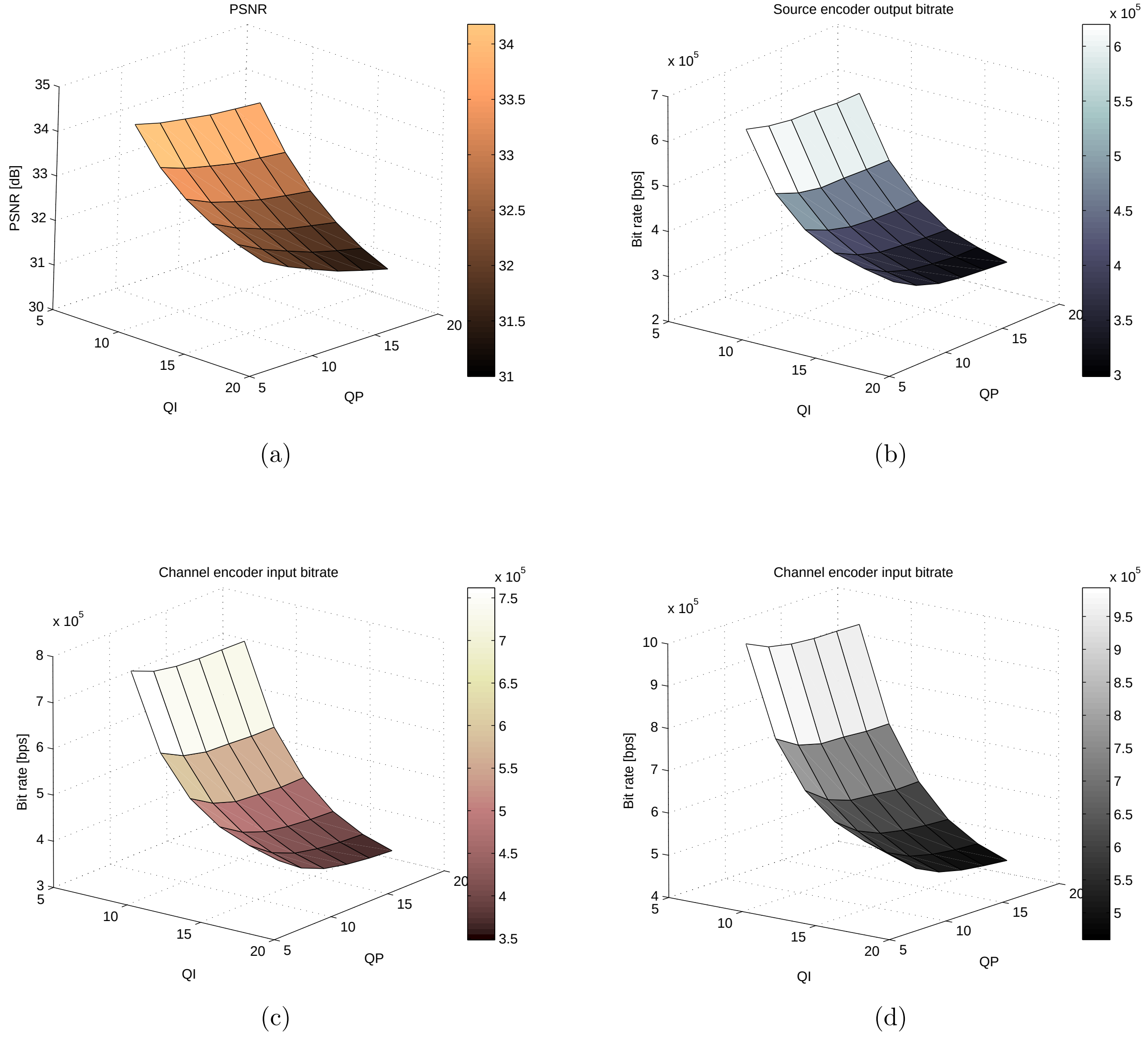
(a)

(b)

(c)

(d)

Figure 3.4: Foreman sequence characterization (CIF, 30fps, GOP=30). [a] PSNR(dB); [b] bitrate output of source encoder; [c] bitrate input of channel encoder (with RoHC); [d] bitrate input of channel encoder (without RoHC).



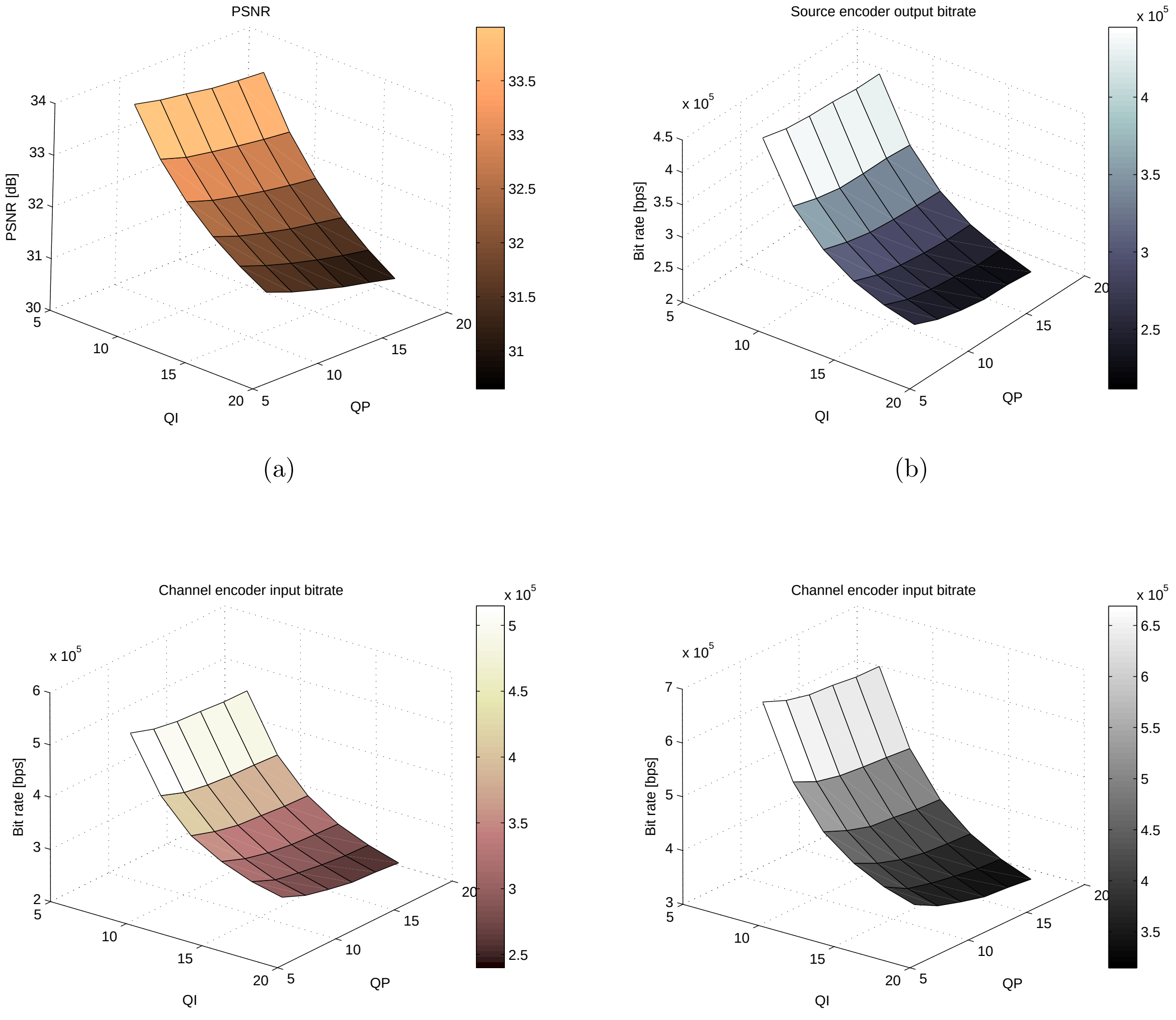
(a)

(b)

(c)

(d)

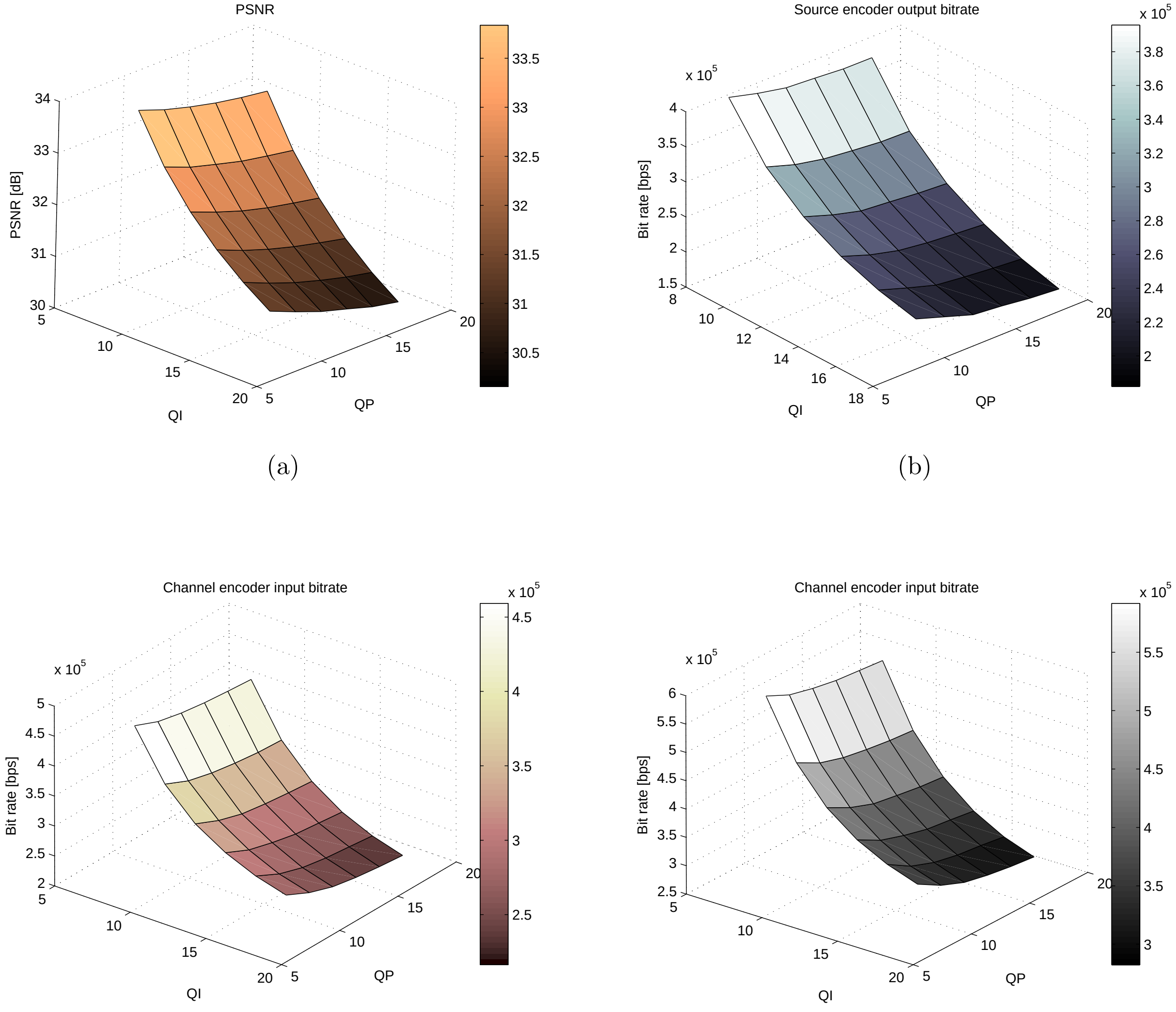
Figure 3.5: Foreman sequence characterization (CIF, 30fps, GOP=15). [a] PSNR(dB); [b] bitrate output of source encoder; [c] bitrate input of channel encoder (with RoHC); [d] bitrate input of channel encoder (without RoHC).



(a) (b)

(c) (d)

Figure 3.6: Foreman sequence characterization (CIF, 15fps, GOP=15). [a] PSNR(dB); [b] bitrate output of source encoder; [c] bitrate input of channel encoder (with RoHC); [d] bitrate input of channel encoder (without RoHC).



(a)

(b)

(c)

(d)

Figure 3.7: Foreman sequence characterization (CIF, 7.5fps, GOP=8). [a] PSNR(dB); [b] bitrate output of source encoder; [c] bitrate input of channel encoder (with RoHC); [d] bitrate input of channel encoder (without RoHC).

- subjective methods: relying on the subjective opinion expressed by a sample of people; while these methods are extremely expensive in terms of time and money, they constitute the most accurate quality estimation technique, because they provide the quality actually perceived by the HVS. The subjective quality tests have been standardized and they are often used to compare different source encoding techniques;

- objective methods: based on the objective computation of a given distortion metric; they can be easily implemented and used, but often they do not reflect the video quality as perceived by the HVS. They are implemented every time an automatic quality assessment technique is required.

Clearly, for our goals, an objective video quality metric is preferable: the source decoder must automatically compute the quality of the received sequence and send the correspondent feedback information to the App JSCC Controller, at the transmitter side. However, at least in addition to this objective estimation, the system may also include some kind of subjective assessment-driven control: e.g. the final user may be able to force the request for a better video quality if not satisfied with the current performance.
The objective quality metrics can be further divided in:

- full reference techniques, requiring the original (reference) sequence to evaluate the distortion metric;

- reduced (or partial) reference techniques, requiring some information about the original sequence;

- no-reference (or blind) techniques, capable of roughly estimating the quality of the decoded images without any knowledge of the original sequence[53].

In particular, in our work we have considered two full reference metrics: the PSNR and the structural similarity index (SSIM). The PSNR (in dB) is defined as

$$PSNR = 20log_{10}\left(\frac{255}{RMSE}\right), \tag{3.3}$$

where RMSE is the square root of the mean square error:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}[F^*(i,j) - F(i,j)]^2}{M \times Q} \tag{3.4}$$

being $F(i,j)$ and $F^{*}(i,j)$ the luminance of the pixel (i,j) in the source and the reconstructed images, containing $M \times Q$ pixels each.

The SSIM is another full reference quality assessment technique, aiming at measuring the structural distortion introduced in the images [52]. Its computation is more complex than the PSNR, but often it gives results more adherent to the quality perceived by the HVS. In the presentation of our results, we often considered both the cited metrics.

3.4.1 Quality feedback

The App JSCC Controller may receive as feedback the video quality information in different forms. In the first version of the *Phoenix* demonstrator, only PSNR was considered as quality information. Anyway, a full-reference video quality metric like the PSNR cannot be used as real feedback when such quality information has to be evaluated on-the-fly, without any reference to the original video sequence since it is not available at the receiver side. For this reason, reduced-reference or no-reference video quality metrics should be used as feedback information by the App JSCC controller. Anyway, at least theoretically, the controller mechanisms do not depend on the quality metric adopted as feedback information. As a confirmation of this statement, in section 5.4 we report the results obtained with different quality feedback.

In particular the feedback quality assessment metrics considered are:

- PSNR;
- SSIM [52];
- reduced reference metric, based on the structure similarity principle, developed at the University of Bologna;
- no-reference metric [53], specifically developed for single-image coding.

It has been observed that the no-reference metric considered only provides a rough approximation of the quality evaluation of the received video in the *Phoenix* system and fails in particular to take into account error propagation. Anyway, given the simplicity and the possibility to evaluate it without any transmitted redundancy, it is considered as possible feedback information. Clearly, a realistic communication scheme would require a reduced reference or, even better, a no-reference quality metric. These considerations motivated our work in the field of dirty paper coding (DPC) and dirty tape coding (DTC), which has been reported in the last chapters of this dissertation. In particular, DPC and DTC are strictly connected to the problem of information embedding and watermarking: inserting invisible known watermarks in the frames of the video sequence permits to estimate the distortion occurred to the watermarks themselves, and so also to the host frames. It is opinion of the author that this technique may constitute a good solution to the problem of distortion measurement at the receiver.

Chapter 4

Physical JSCC Controller

The task of the physical layer controller is basically to adapt channel coding and modulation parameters to channel conditions and video stream characteristics [66][67]. The adaptation is performed according to detailed CSI and, eventually, NSI, since it should follow channel variations and fit as much as possible to the channel characteristics. According to the side information from the channel and the source, the controller provides commands to the channel encoder (in terms of channel encoder type, code-rate and word length...), to the modulator (in terms of constellation size, bit-loading parameters or spreading factor in the case spread spectrum techniques are adopted) and to the interleaver, defining interleaver type and interleaving depth. The mode to be used in the case of multiantenna transmission will also be determined by the physical layer controller. At the receiver side, the controller has a dual task: according to the available information and possibly in accordance with the transmission side controller the receiver blocks parameters are determined. Moreover, some specific parameters are chosen by the Phy JSCC Controller at the receiver, e.g. the iteration number in iterative decoding scheme and the possible decoding iterations among the the demodulator, the channel decoder and the source decoder, through a proper exchanging of soft information.

4.1 Channel code rates optimization

According to the model developed in [29], different channel code rates are chosen by the Phy JSCC Controller for the different data partitions in order to minimize the overall distortion, jointly determined by the lossy source coding and the errors occurred during the transmission through the channel. The source encoder produces a bitstream that can be separated in N layers or partitions P_i of different significance. Each partition is protected with a different channel code of rate $R_{C,i}$ according to its sensitivity to channel errors.

Each partition P_i has a source rate:

$$R_{S,i} = \phi_i R_S = \frac{B_i}{B} R_S \tag{4.1}$$

where R_S is the average source rate in [bits/pixel], and $\phi_i = B_i/B$ is the ratio between the number of bits per frame belonging to the i^{th} partition, B_i , and the total number of bits/frame, B. Clearly it holds:

$$\sum_{i=1}^{N} \phi_i = \frac{1}{B} \sum_{i=1}^{N} B_i = 1. \tag{4.2}$$

Denoting with M and Q the frame width and height respectively, we have $B = MQR_S$.

Each partition is then protected with an error correcting code with rate $R_{C,i}$, such that the total source and channel code rate R_{S+C} is given by:

$$R_{S+C} = \sum_{i=1}^{N} \frac{R_{S,i}}{R_{C,i}} = \frac{R_S}{B} \sum_{i=1}^{N} \frac{B_i}{R_C, i} = \frac{B_C}{MQ}, \tag{4.3}$$

where $B_C \triangleq \sum_{i=1}^{N} \frac{B_i}{R_{C,i}}$ represents the number of channel coded bits per frame. Theoretically, the optimal joint source and channel coding approach would require to find both the sets $\mathbf{R}_S = (R_{S,1}, \ldots, R_{S,N})$ and $\mathbf{R}_C = (R_{C,1}, \ldots, R_{C,N})$ minimizing the overall end-to-end distortion D_{S+C}, which includes the contributes due to the lossy source coding and to the errors along the transmission channel, under a constraint on the maximum number of bits/pixel resulting from the whole encoding process. Defining the mean square error between the reconstructed frame, $\mathbf{F}^*$, and the corresponding original frame, $\mathbf{F}$, as

$$MSE(\mathbf{F}, \mathbf{F}^*) = \sum_{i=1}^{M} \sum_{j=1}^{Q} \frac{\left(F^*(i,j) - F(i,j)\right)^2}{M\dot{Q}}, \tag{4.4}$$

where $F^*(i,j)$ and $F(i,j)$ represent the luminance value of pixel (i,j) in the received and original frame respectively, we can write $D_{S+C} = D_{S+C}(\mathbf{R}_S, \mathbf{R}_C) = E\{MSE(\mathbf{F}, \mathbf{F}^*)\}$, where the expected value is computed with respect to the conditional probability density function (p.d.f.) $p\left(\mathbf{F}, \mathbf{F}^* \middle| \, \mathbf{R}_S, \mathbf{R}_C\right)$.

Following the JSCC approach described in Chapters 1 and 2, we are interested in a slightly simplified version of the stated problem: we are looking for the optimal source-dependent choice of channel coding rates for a given source coded bitstream with fixed parameters. Thus, given a fixed set of source rates $\mathbf{R}_S = (R_{S,1}, \ldots, R_{S,N})$ determined by the App JSCC Controller, we need to find the solution of the constrained minimization problem

$$\begin{cases} \mathbf{R}_C = \arg\min_{\mathbf{R'}_C} D_{S+C}(\mathbf{R}_S, \mathbf{R'}_C) \\ \\ \qquad s.t. \quad R_{S+C} \leq \frac{R_S}{R_C} \end{cases} \tag{4.5}$$

where the ratio R_S is determined by the source coding parameters and R_C is the constraint specified by the App JSCC Controller. The inequality in (4.5) can also be re-written as

$$R_{S+C} = \frac{R_S}{B} \sum_{i=1}^{N} \frac{B_i}{R_C, i} \leq \frac{R_S}{R_C} \quad \Rightarrow \quad \sum_{i=1}^{N} \frac{\phi_i}{R_C, i} \leq \frac{1}{R_C}. \tag{4.6}$$

Equivalently, the optimization problem (4.5) can be stated as a constrained maximization of the average PSNR.

Assuming that the source and channel distortions are additive (hypothesis which is valid for MPEG-4 video encoding under common operating conditions), and indicating with $\mathbf{P}_e$ the set $(P_{e,1}, \cdots, P_{e,N})$, where $P_{e,i}$ is the residual (after channel co-decoding) bit error rate of the i^{th} partition, the total distortion can be written as[1]:

$$D_{S+C}(\mathbf{R}_S; \mathbf{P}_e) = D_S(\mathbf{R}_S) + D_C(\mathbf{P}_e). \tag{4.7}$$

$D_S(\mathbf{R}_S)$ and $D_C(\mathbf{P}_e)$ are the distortion due to source compression when transmission over a perfect channel is assumed and the channel distortion when an uncompressed source is considered respectively. Supported by experimental results, we can further simplify the problem and assume that distortion additivity holds also with respect to the different data partitions. Thus we can write:

$$\begin{aligned} D_{S+C}(\mathbf{R}_S; P_{e,1}, P_{e,2}, ..., P_{e,N}) \approx & D_{S+C}(\mathbf{R}_S; P_{e,1}, 0, ..., 0) + D_{S+C}(\mathbf{R}_S; 0, P_{e,2}, 0, ...0) + ... \\ & + D_{S+C}(\mathbf{R}_S; 0, ..., 0, P_{e,N}) - (N-1) D_S(\mathbf{R}_S). \end{aligned} \tag{4.8}$$

Thus, in order to solve the minimization problem, we need to know the class conditional distortion curves

$$D_{S+C}(\mathbf{R}_S; P_{e,1}, 0, \ldots, 0), \;\; D_{S+C}(\mathbf{R}_S; 0, P_{e,2}, 0, \ldots, 0), \;\; \ldots, \;\; D_{S+C}(\mathbf{R}_S; 0, 0, \ldots, P_{e,N}) \tag{4.9}$$

[1] Here we use a slight abuse of notation, indicating $D_{S+C}(\mathbf{R}_S; \mathbf{P}_e)$ with the same form of $D_{S+C}(\mathbf{R}_S; \mathbf{R}_C)$, but we are justified by the observation that, given a transmission channel state, the residual probability of error is determined by the applied code rate, i.e. $P_{e,i} = P_e(R_{C,i})$ for $i = 1, \ldots, N$.

for every possible set of source rates $\mathbf{R}_S$.

In the following we denote with $\mathcal{R}$ the set of all the different channel code rates available for transmission. We indicate with R_C^k the k^{th} code rate available in $\mathcal{R}$ and we consider the set ordered according to the rule $R_C^k > R_C^{k+1}$ for $k = 1, \ldots, |\mathcal{R}| - 1$. Clearly, the optimal solution of (4.5) is constituted by a sequence of code rates $R_{C,i} \in \mathcal{R}$ for $i = 1, \ldots, N$. In principle, distinct channel codes can be used, characterized by different code rates and correction capabilities, but to reduce the transmitter complexity it is advisable to use a single channel code, also called *mother* code, followed by a proper puncturing unit. In our joint coding approach we have considered two different solutions, based on RCPC codes and on punctured irregular repeat-accumulate (IRA) LDPC codes.

Finding the optimal $\mathbf{R}_C$ involves a search in a set of $|\mathcal{R}|^N$ elements. To reduce the computational complexity of the algorithm, it is possible to apply some techniques mutuated from the lattice decoding field. In particular, we can build an N-level tree, with $|\mathcal{R}|$ branches departing from every node. Thus we have exactly $|\mathcal{R}|^N$ terminal nodes, corresponding to all the possible code rate combinations for the N partitions. An exhaustive search for the optimal rates would mean visiting all the terminal nodes, computing for each of them the distortion D_{S+C} and verifying the constraint on the average code rate. A better approach permits to reduce the number of visited nodes by pruning entire sub-trees which do not satisfy the constraint. Let's start considering the $N = 1$ case: clearly, we need to compute for each $R_C^k \in \mathcal{R}$ the distortion D_{S+C} and to verify the constraint $\frac{1}{R_C^k} \leq \frac{1}{R_C}$. Actually, thanks to the order of the elements of $\mathcal{R}$, it is possible to stop the computation after the first R_C^k such that $\frac{1}{R_C^k} > \frac{1}{R_C}$. In fact we have $\frac{1}{R_C^r} > \frac{1}{R_C^k}$ for every $r > k$. Analogously, for the N partition case, we start determining all the code rates allowed for the first data layer considered as if the only one present, calculating $\frac{\phi_1}{R_C^k} \leq \frac{1}{R_C}$. This way we can discard all the non-admissible nodes at the first level and the entire sub-trees departing from them. The algorithm proceeds then with the second level of the tree: for all the nodes remaining after the first *pruning* operation, we verify that the constraint on the average code rate is satisfied for the first two data layers and discard the non-admissible nodes with the respective sub-trees. The algorithm proceeds in the same way until the N^{th} level: in the end, just a sub-set of the possible terminal nodes remain and for the corresponding sets of code rates, $\mathbf{R}_C$, we need to evaluate the distortion D_{S+C} and to find the optimum.

An example of tree with $N = 3$ and $|\mathcal{R}| = 2$ has been reported in Fig. 4.1, where we have indicated with dashed lines the pruned branches and sub-trees and with a cross the corresponding non-admissible nodes. In this example, we need to compute the conditional

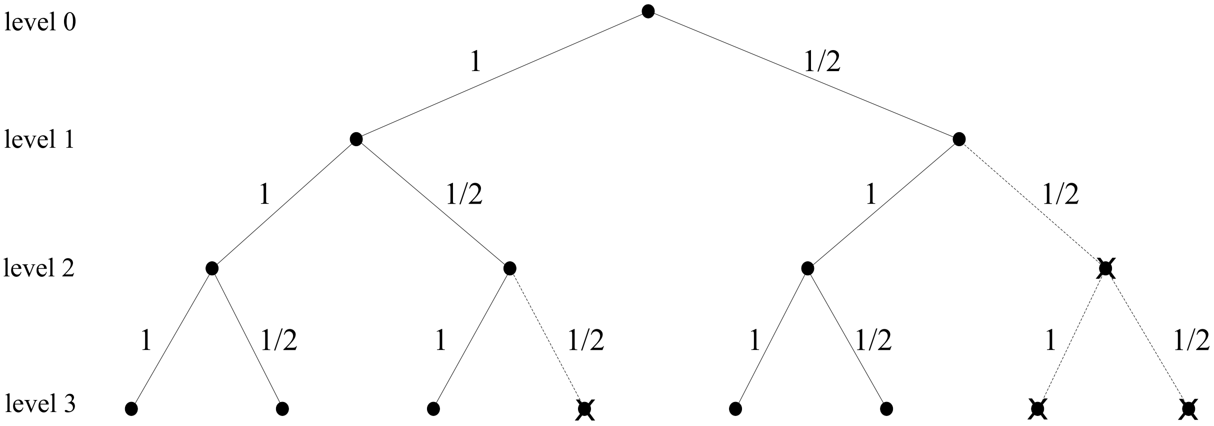


Figure 4.1: Example of code rate sets allowed by the constraint: $N = 3$, $(\phi_1, \phi_2, \phi_3) = (1/3, 1/2, 1/6)$, $\mathcal{R}$={1,1/2}.

distortion curves for just 5 different code rates combinations out of the $2^3 = 8$ possible in all. Clearly, this example is purely illustrative: the advantages in terms of complexity become more evident when considering transmission schemes with up to 20 available coding rates, like what we considered in some communication scenarios. The described optimization technique is given in Algorithm 1, where we have indicated with **break** the statement interrupting the execution of the corresponding **FOR** loop. Lines 3 and 4 of Algorithm 1 verify the existence of at least one code rate (R_C^1 is the maximum available code rate) satisfying the constraint: if such a condition is not true, the algorithm declares an error and exit.

4.2 Distortion models for MPEG-4 coded sequences

In this paragraph we specify the general algorithm presented in the previous section to real source and channel coding situations. To this purpose, the main problem we have to solve is how to determine the conditional distortion curves. So far we have assumed to know how to compute the terms $D_{S+C}(\mathbf{R}_S; 0, \ldots, P_{e,i}, \ldots, 0)$ without specifying what mathematical tools make it possible. We start observing that, evidently, the Phy JSCC Controller can only try to estimate the actual distortion, based on several elements:

- the video source statistical characteristics: amount of motion present in the sequence, spatial correlation within each frame, etc.

- the source coding parameters: quantization scale, intra-refreshment rate, etc.

Algorithm 1 Determine the optimal $\mathbf{R}_C$

```
Require: $\mathbf{R_S}$; $\phi_i$ for $i = 1, \ldots, N$; $R_C$; $P_e = P_e(R_C^k)$ for $k = 1, \ldots, |\mathcal{R}|$
Ensure: optimal $\mathbf{R}_C = (R_{C,1}, \ldots, R_{C,N})$
$\mathbf{R}_C \Leftarrow (1, \ldots, 1)$
$D_{min} \Leftarrow +\infty$
if $R_C^1 < R_C$ then
    declare an error and exit
else
    for $n_1 = 1$ to $|\mathcal{R}|$ do
        $s_1 \Leftarrow \frac{\phi_1}{R_C^{n_1}}$
        if $s_1 > \frac{1}{R_C}$ then
            break
        else
            for $n_2 = 1$ to $|\mathcal{R}|$ do
                $s_2 \Leftarrow s_1 + \frac{\phi_2}{R_C^{n_2}}$
                if $s_2 > \frac{1}{R_C}$ then
                    break
                else
                    ...
                    for $n_N = 1$ to $|\mathcal{R}|$ do
                        $s_N \Leftarrow s_{N-1} + \frac{\phi_N}{R_C^{n_N}}$
                        if $s_N > \frac{1}{R_C}$ then
                            break
                        else
                            $D_{test} \Leftarrow D_{S+C}(\mathbf{R}_S; P_e(R_C^{n_1}), \ldots, P_e(R_C^{n_N}))$
                            if $D_{test} < D_{min}$ then
                                $\mathbf{R}_C \Leftarrow (R_C^{n_1}, \ldots, R_C^{n_N})$
                                $D_{min} \Leftarrow D_{test}$
                            end if
                        end if
                    end for
                    ...
                end if
            end for
        end if
    end for
end if
```

- the transmission channel state: statistical channel characteristics, SNR, etc.
- the channel coding parameters: codeword length, code rates, ets.

In particular, the first two points jointly determine the source rates $\mathbf{R}_S$ for all the different data layers, while the latter two points characterize the functions $P_{e,i} = P_e(R_{C,i})$. Building an analytical model taking into account all these elements is not an easy task and several approaches have been adopted in the literature. We have implemented the distortion estimation algorithm reported in [29] and synthetically described in the following. This technique has been originally developed for the MPEG-4 standard but we are currently extending it to more recent standards, like H.264.

The frame distortion is differently estimated according to the type of frame considered. In the following we describe the technique in case of intra-coded (I) and inter-coded (P) frames.

4.2.1 I frames

The distortion term due to quantization is deterministic and, in our additive model, it cannot be affected by the code rate optimization technique previously described. Thus, the only term we need to minimize is the distortion determined by transmission errors and it can be modelled following the technique in [29]. Considering the occurrence of errors only in the first data partition, made up of the DC coefficients from the DCT, we have

$$D_{C,I}(P_{e,1}, 0) = \alpha(\phi_1 P_{e,1})^{\beta} N_p \tag{4.10}$$

where α, β are numerical parameters which can be determined through fitting. However, it has been noticed that, for a fixed packet length, the parameters α and β can be considered invariant for different sequences. In particular, in our work we have considered quite short packets, about a thousand bits long, and values of $\alpha = 9.47 \cdot 10^3$ and $\beta = 1.03$. For the second packet partition, constituted by the remaining DCT coefficients, the following model has been adopted:

$$D_{C,I}(0, P_{e,2}) = \gamma(P_{e,2})^{\delta} N_p^2. \tag{4.11}$$

Also in this case, for a fixed packet length, the same parameters $\gamma = 2.1$ and $\delta = 0.75$ have been used for all the investigated sequences.

4.2.2 P Frames

The adopted optimization technique is frame-based: in other words it tries to minimize the distortion introduced in the i^{th} frame, considering all the previous frames and their distortion as given. A better approach may consider entire groups of frames and aim to minimize the average or the maximum distortion within each group. In fact, the quality of predicted frames (P or B) is strongly dependant on the distortion of the reference frames. However, we have opted for the single frame approach, basically in order to reduce the complexity of the code rate selection tool and to minimize the introduced delay. In fact a group-of-frame approach would require the knowledge of the statistical parameters (e.g. the ϕ_i) of several frames before being applicable.

During the transmission of the i^{th} frame the distortion relevant to previously transmitted frames is fixed and the quality of the current frame depends on it. While for I frames the distortion estimation can be performed quite accurately, the problem with predicted frames is more delicate, because the mean square error (MSE) estimation depends on many previous estimation and the accuracy tends to decrease within a GOP with increasing frame numbers. Then we made some simplifying assumptions: first of all we considered only predicted frames of type P. In the second place, our goal is to minimize the increment of distortion introduced in the i^{th} frame with respect to the previous one, considered as perfectly received. Here we denote this term with $D_{C,P}(P_{e,1}, P_{e,2})$, where, as usual, $P_{e,1}$ and $P_{e,2}$ are the probability of bit error for data partition 1 and 2 respectively. The first data partition of a P-frame packet contains motion information and motion marker. The distortion curves $D_{C,P}(P_{e,1}, 0)$ may be modelled similarly to the case of I frames:

$$D_{C,P}(P_{e,1}, 0) = \varepsilon(P_1 P_{e,1})^{\eta} N_p \tag{4.12}$$

The parameters ε and η may be considered again independent of the sequence, but dependent on the bit rate (expressed here in number of packets per frame, N_p, assuming a fixed packet length). The second partition contains texture information, i.e. the DC and AC DCT coefficients. The model considered is again similar to what has been suggested for I frames, but with different parameters:

$$D_{C,P}(0, P_{e,2}) = \lambda(P_{e,2})^{\theta} N_{packets}^2. \tag{4.13}$$

In Fig. 4.2 we report the curves obtained with and without the adoption of the optimized code rate selection algorithm. The considered video sequence is the *Foreman* test sequence in CIF format, 15fps and GOP=15 frames. The quantization scale is the same

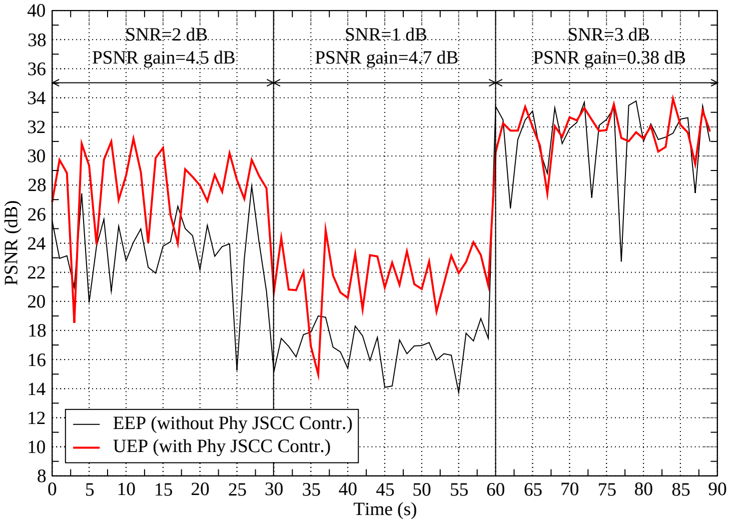
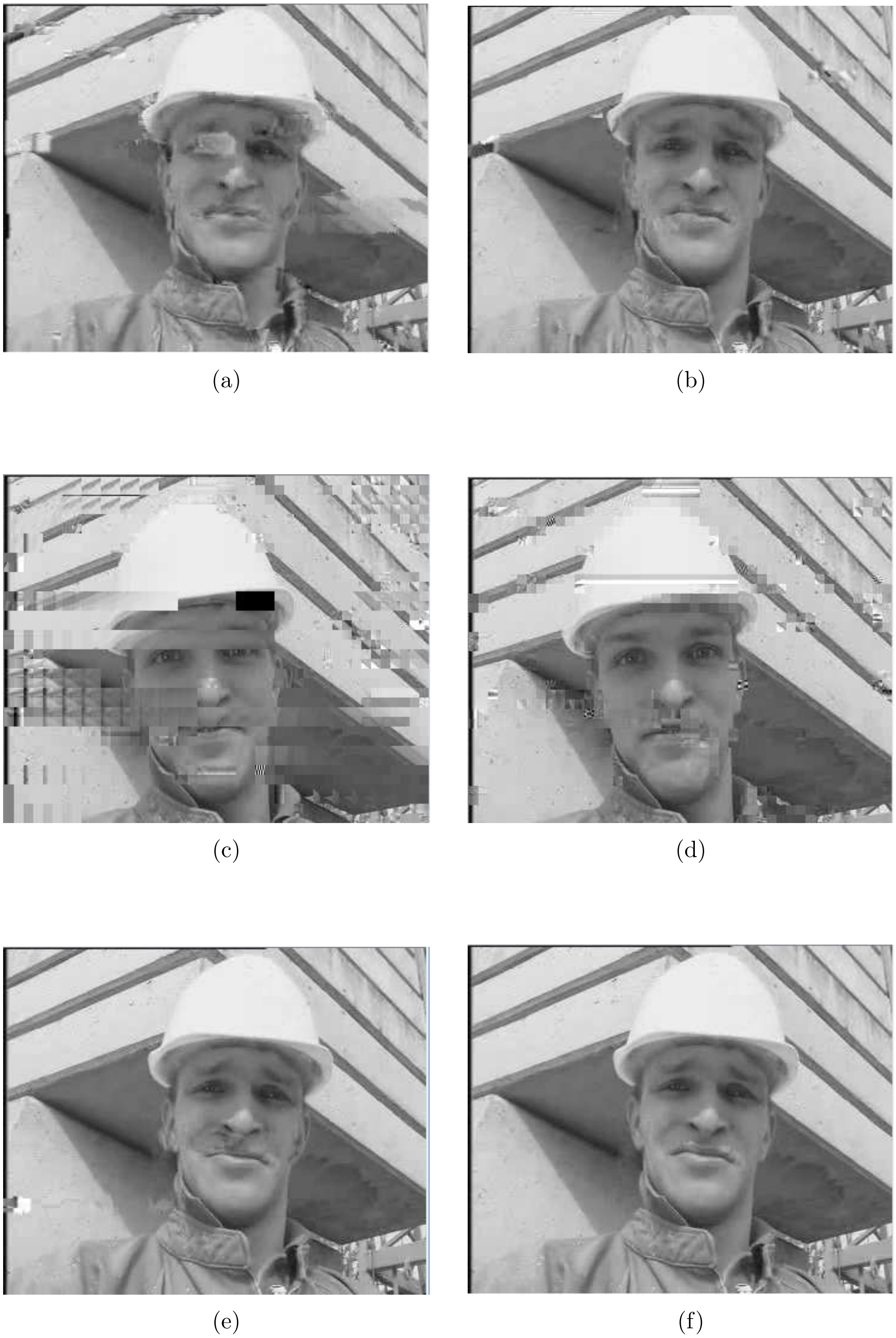


Figure 4.2: PSNR gain due to the optimization algorithm in an AWGN channel with varying SNR. The code is a $(k = 1, n = 3, k = 5)$ RCPCC and the constraint on the code rate is $R_C = 2/3$.

for both I and P frames, $qI = qP = 8$. The channel code is a rate compatible punctured convolutional code (RCPCC), with a *mother* code with $k = 1$, $n = 3$ and constraint length $K = 5$ [70]. We considered a puncturing period $P = 8$ and 19 different puncturing matrices. The average code rate is $R_C = 2/3$, with both UEP and equal error protection (EEP) policy. Fig. 4.2 refers to binary phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel with a varying SNR of 1, 2 and 3 dB. As explicitly reported above the curves, the gain in terms of PSNR varies from 0.38 to 4.7 dB, showing the effectiveness of the proposed technique. Some examples of frames obtained with the traditional transmission approach and with the optimized UEP have been reported in Fig. 4.3 for different channel conditions. Also from a visual comparison of the frames, the improvements in terms of perceived quality are remarkable.

4.2.3 Parameters required by the algorithm

It is worth here making some observations about the parameters required by the optimal code rate selection algorithm. We start considering the estimation of the probability of error for the different data partitions, $P_{e,i}$. We notice that these values are fundamental in order to exploit the conditional distortion curves. So far, we have assumed to know exactly the dependency $P_{e,i} = P_e(R_{C,i})$ once the channel state is given. In fact, we suppose



(a) (b) (c) (d) (e) (f)

Figure 4.3: Examples of frames obtained with ([b], [d], [f]) and without ([a], [c], [e]) the optimization algorithm: [a]-[b] SNR=2 dB; [c]-[d] SNR=1 dB; [e]-[f] SNR=3 dB.

that the information about the channel state are provided as feed-back from the radio receiver to the transmitter, with a rate sufficiently high to track the channel variations. Unfortunately, even in case of channel state information ideally known by the transmitter, the assumption of knowing $P_{e,i}$ is too optimistic in many realistic communication scenarios. We approached the problem considering, as reported in the example above, optimal RCPC codes [70][71][34], known for having performance almost as good as the best known convolutional codes with the same protection rate [35]. Then, in order to determine $P_{e,i}$, we tried to exploit the corresponding upper bound

$$P_e \leq \frac{1}{P} \sum_{d=d_f}^{\infty} a_d \cdot P_d \tag{4.14}$$

where d_f is the free distance of the code, a_d is the number of existing paths at distance d and $P_d = \frac{1}{2}\text{erfc}\sqrt{\frac{dE_S}{N_0}}$ is the probability that the wrong path at distance d is selected (here we consider an AWGN channel with SNR $= E_S/N_0$). The bound (4.14) is quite tight in the high SNR region, while it is loose for bad channel conditions. Unfortunately, the low SNR region is the most important when considering UEP techniques, because it is exactly when the EEP policy starts failing that unbalancing the protection rates becomes a preferable solution. As a consequence, the distortion estimation may be performed with inaccurate values of $P_{e,i}$ and it may easily result in even worse quality performance than a traditional EEP scheme.

For these reasons, we opted for a solution based on look-up tables. In particular, we obtained several curves for the $P_{e,i}$, under different channel conditions and SNRs. During the transmission, the optimization algorithm computes the estimated bit error rates through a proper interpolation of the values available in the look-up tables. This solution provides better results than the technique based on the upper bound (4.14) and so we decided to adopt it also within the *Phoenix* project.

Other important parameters are the percentages ϕ_i for $i = 1, \ldots, N$, representing the ratio between the number of bits per frame belonging to the i^{th} partition and the total number of bits/frame. These side information are required in order to keep satisfied the constraint on the average code rate while optimizing the different protection levels. There are several ways for the radio transmitter to determine them. We have considered two different possibilities: first of all, these information can be provided on a frame-by-frame basis by the source encoder, as SRI. In particular, they may be included in an IPv6 network header extension with the SSI and exploited by the Phy JSCC Controller at the physical layer. These information are quantized and coded in few bits, determining a small overhead and allowing their periodical repetition, with a rate depending on the IPv6 network state. In fact the problem of network packet loss may be easily overcome with sufficiently frequent SRI re-transmissions. Another solution may consists in the estimation of the ϕ_i's based on the specific SSI received in the last N_W packets. In this case, the Phy JSCC Controller computes the partition lengths by averaging on a sliding window of N_W packets and verifies the constraint on R_C with the corresponding terms. Clearly, the shorter is the averaging window, the faster is the Phy JSCC Controller in reacting to changed statistical video characteristics, but also the less accurate is the estimated distortion calculated from the model. Within the *Phoenix* project we decided to apply the optimal code rate selection algorithm and the ϕ_i is specified by the source encoder when using RCPCCs as channel codes, while a sub-optimal (but simpler) solution has been adopted with LDPC codes.

4.3 Unequal error protection with LDPC codes

Discovered by R. Gallager in 1961 [36], low density parity check codes (LDPCCs) have been object of intensive studies and deep investigations in the recent years. In fact, their remarkable performance in terms of error correction capability and their extreme flexibility in every designing aspects make them good candidates for a wide range of application scenarios. LDPCCs are linear block codes characterised by a sparse parity check matrix $\mathbf{H}$, i.e. a matrix with a low density of ones. If k and n are the lengths of the block of information bits and of the code word respectively, $\mathbf{H}$ has dimension $(n-k)\times n$ and assumes the typical aspect:

$$\mathbf{H} = \begin{bmatrix}
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0
\end{bmatrix}$$

The generation of this matrix can be done deterministically, according to algebraic and geometric rules, or randomly, according to the desired weight distribution of rows

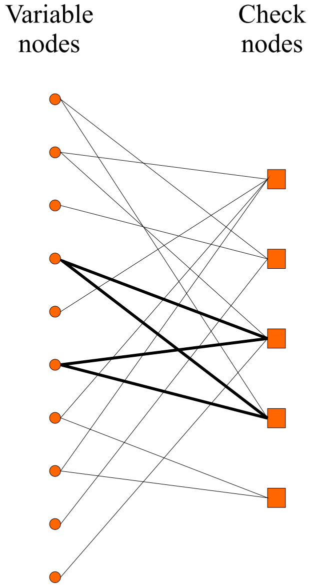


Figure 4.4: Example of Tanner Graph.

and columns. Another way of representing the parity check matrix is the Tanner Graph of the code, where on the left are reported the *variable nodes* (representing the bits of the code word) and on the right the *check nodes* (representing the parity check equations).

With this graphical representation it is immediately clear which bits of the code word are involved in the calculation of each parity check equation and, vice versa, which set of parity check equations is affected by a given bit.

A *cycle* is a path constituted by adjacent graph edges that, starting from a certain node, ends up in the same node. The *degree* of a variable or check node is defined as the number of edges connected to that node. According to this parameter, we can distinguish two family of LDPCCs: the *regular* and the *irregular* codes. An LDPCC is regular if the check node degree and the variable node degree are constant. Otherwise, the code is said irregular. The performance behavior of these two families of codes, in terms of BER versus SNR, are typically different and suggests that different codes would be preferable with respect to the channel conditions. In fact, Richardson et al. [41] have proposed weight distributions for irregular codes that permit to achieve better performance at low SNR values if compared to regular codes. However, this kind of irregular codes typically shows an error floor whereas the regular ones have floors at lower BER values (or maybe they don't have a floor at all) [40].

4.3.1 Decoding Procedure

The decoding process is based on a message passing algorithm, in which the variable and the check nodes iteratively exchange messages along the graph edges. Many kinds of algorithms have been proposed in the literature, either soft or hard, and characterized by different complexity and performance. The main techniques are:

- *Belief Propagation* (BP) - Based on soft value messages. Proposed by Gallager, it is the most complex message-passing algorithm and, on cycle-free graph, it converges to the A Posteriori Probability Ratio for each variable node. So, for cycle-free graph it can be seen as the optimum. Because of the non-linear function computations that it requires, the BP algorithm is often implemented by means of proper look-up tables.

- *Min Sum* (MS) - Based on soft value messages. It can be seen as a simplified version of the BP algorithm and, on cycle-free graph, it constitutes only a sub-optimal solution. The MS algorithm shows a reduced complexity if compared with the BP, but it achieves worse performance.

- *Bit Flipping* (BF) - Based on hard value messages. The algorithm simply inverts the value of the bits which, at each step, are considered erroneous with highest probability. It presents even less complexity than the MS algorithm but it has also the worse performance.

It is a remarkable characteristic of these codes that all the message passing algorithm are completely independent of the particular code used. Therefore it is easily imaginable a scenario where the same encoded video stream is transmitted over the radio channel, while different receivers implement different decoding techniques, according to their computational capabilities and target video-qualities.

As previously stated, the decoding algorithm is iterative. There are no constraints on the number of iterations but, obviously, the more they are the better are the performance. In video transmissions the number of iterations will depend on the maximum delay acceptable according to the real-time constraints, besides to the computational capabilities of the receiver terminal.

In addiction, at the end of the decoding process, it's possible to know if the resulting sequence is an allowed codeword or not. In other words, with LDPCCs error detection capability comes for free.

4.3.2 Encoding Procedure

Until a few years ago, the encoding task has been the main drawback of LDPCCs because of its apparent complexity. On the contrary, recent studies and developments have shown that relatively simple encoders can be designed for cyclic or quasi-cyclic LDPCCs based on finite geometries or for IRA-like codes (Irregular Repeat and Accumulate codes) [42][39][37].

4.3.3 LDPC-based UEP

One of the forward error correction (FEC) schemes employed in the Phoenix chain is based on a [36] LDPCC codec. A rate-compatible scheme has been designed, which permits the choice of the final code rate within 5 options: 1/3 (denoted as *scheme 0*), 1/2 (*scheme 1*), 2/3 (*scheme 2*), 3/4 (*scheme 3*) and 5/6 (*scheme 4*).

The code family keeps the codeword length constant (i.e., n is constant, while the code rate adaptation is obtained by a proper choice of the information block length, k). In a first implementation, the code has been designed choosing *n=4200*. Therefore, the code family is constituted by the following codes:

- Scheme 0: n=4200, k=1400
- Scheme 1: n=4200, k=2100
- Scheme 2: n=4200, k=2800
- Scheme 3: n=4200, k=3150
- Scheme 4: n=4200, k=3500

The encoding procedure exploits puncturing and shortening of a (10500,3500) *mother code* to achieve the desired code rates and block lengths. The belief propagation (BP) algorithm is based on the bipartite graph of the mother code, with a proper initialization of the channel inputs for the punctured shortened variable nodes: punctured variable nodes are initialized with an a priori probability log-likelihood ratio (APP-LLR) equal to 0, while shortened variable nodes are initialized with a large positive APP-LLR (which approximates an infinite-reliable knowledge of the codeword bit associated to the shortened variable node).

The iterative decoding of a punctured LDPC code can succeed if the code is properly designed, and if the choice of the puncturing patterns does not lead to *stopping sets*. The

Scheme	Code rate	Periodic puncturing pattern (parity part)	Shortening range (systematic part)
0	1/3	10101010101010101010	1400-3499
1	1/2	11011011011101101101	2100-3499
2	2/3	01111011110111101111	2800-3499
3	3/4	11111011111011111101	3150-3499
4	5/6	01111111110111111111	None

Table 4.1: LDPCC puncturing and shortening to obtain the different coding schemes.

mother code is a systematic irregular repeat-accumulate [37][39] (IRA) LDPC code. IRA codes present many advantages with respect to randomly-designed LDPC codes:

- an IRA code can be efficiently encoded through a simple concatenation of a low-density generator matrix code and an accumulator;
- they present a large number of degree-2 variable nodes, allowing the achievement of near-Shannon-limit decoding thresholds;
- the parity bits are placed in correspondence of degree-2 variable nodes. This association allows the choice of puncturing patterns that do not involve stopping sets (iterative decoding convergence is preserved).

The LDPC *mother* code we have used in our work is a (10500,3500) IRA code characterized by the shortening/puncturing parameters listed in table 4.1 for the different coding schemes available.

4.4 Code rate selection for LDPC-based UEP

There is an important difference between a scheme performing UEP through RCPCCs and one based on LDPC codes: with a solution based on convolutional codes, the transmitted codewords can be variably long and their structure can reflect the data partitioning of the video packets. In other terms, after coding an entire packet with the *mother* encoder, different puncturing matrices can be used along the same codeword, determining different protection levels for the corresponding data. Clearly the decoder requires the knowledge of the de-puncturing matrices to apply and the positions where to change them along the received sequence. On the contrary, with the proposed UEP scheme based on LDPCCs, the protection inside each packet is constant and determined by the selected coding scheme.

Moreover, the packets to encode should have a predetermined length, according to the values of k in table 4.1. The latter problem can be easily overcome through shortening, while, in order to manage packets containing data belonging to different priority layers, it becomes necessary to rearrange the whole packet structure. In particular we need to build packets with data belonging to a single partition and to apply the corresponding channel coding scheme.

The solution we adopted in the Phoenix project relies on a buffer with a single input queue and multiple output queues, corresponding to the different priority layers considered. In this paragraph we shortly describe the packet reorganization and code rate selection technique adopted with LDPC coding schemes.

When a packet arrives at the radio link block, it is decomposed and the data belonging to the different priority layer are sent to N distinct output queues. When dealing with MPEG-4 coded bitstreams, three data partitions have been considered:

- network headers (real time protocol (RTP), UDP/UDP lite, IP) and MPEG-4 VP header;
- partition 1 data of frame-I (DC DCT coefficient and resync markers) and frame-P (motion vectors and resync markers);
- partition 2 data of frame-I (AC DCT coefficients) and frame-P (DCT coefficients).

In this paragraph we denote with k_i the information block length selected for the i^{th} data partition. When a queue contains k_i bits, the data are extracted, channel encoded, modulated and transmitted through the radio channel. At the receiver side a dual process is performed by a buffer with multiple input queues and a single output queue. Thus, selecting the values k_i for the transmitter output queues means deciding the code rates for the different data partitions, while the transmitted packets have all the same size (4200 bits). In our work we made the assumption that all the information required by the receiver to correctly perform the packet re-composition are ideally available. In other words we are supposing that a proper header is added to the transmitted packets and that it will be sufficiently protected against channel errors. Algorithm 2 and Fig. 4.5 illustrate the packet re-organization procedure with LDPC codes.

A final observation on line 6 of Algorithm 2: the code rate selection verifies that the constraint on the average code rate R_C is satisfied and it may be based on the optimal algorithm of section 4.1. With LDPC codes, however, we implemented a simpler strategy, basically motivated by two reasons:

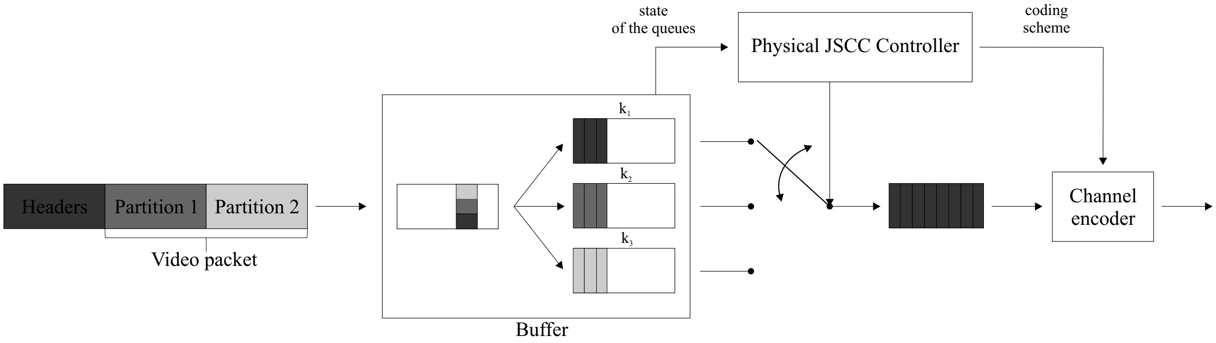


Figure 4.5: Bitstream re-organization in fixed-length packets at the transmitter side.

Algorithm 2 Transmission procedure with LDPC codes

```
for each VP do
    decompose packet partitions and send them to the queues
    for i=1 to N do
        update $\phi_i$ averaging on a sliding window
    end for
    select $(k_1, \dots, k_N)$
    for i=1 to $N$ do
        while $k_i$ bits are present in queue $i$ do
            take $k_i$ bits and send to channel encoder
        end while
    end for
end for
```

- the number of code rates available for LDPCC is limited with respect to the RCPC case, so that just a few combinations are possible with a given ϕ_i;

- the intrinsic error detection capabilities of LDPC decoding allow avoiding the usage of the functions $P_{e,i} = P_e(R_{C,i})$, which, as previously noted, are difficult to model.

The transmission is initialized in EEP mode. Through the feedback channel, the decoder inform the transmitter side when it can not correctly decode the received packets, because of the channel conditions. Then the Phy JSCC Controller decides if switching to a UEP strategy or not, based on the constraint on the average code rate and on the CSI. The UEP strategy here mentioned consists in unbalancing the protection among different data partitions, decreasing the code rate for the most important data (headers and partition 1) and increasing the least as possible the code rate for partition 2. Moreover, the transmitter keep track of the channel conditions which prevented the decoder from converging to an

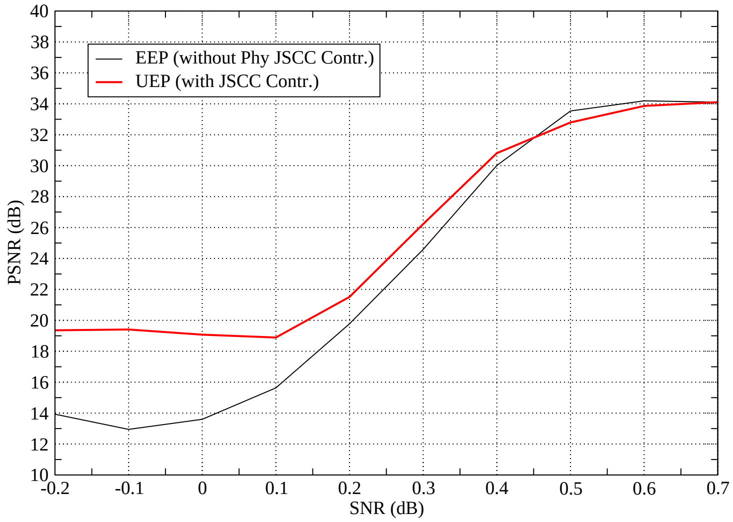


Figure 4.6: PSNR gain due to the optimization algorithm in AWGN channels with different SNR. The *mother* code is an IRA LDPCC $(3500, 10500)$ and the constraint on the code rate is $R_C = 2/3$.

allowed codeword and switch back to the EEP strategy as soon as the SNR increase.

In Fig. 4.6 we have reported some curves showing the advantages of the described LDPC-based UEP strategy versus a traditional approach. The curves refer to an AWGN channel with different SNR and they have been obtained averaging through 3 simulation runs of several seconds each.

4.5 UEP and OFDM modulation

In the following sections we will show a novel UEP scheme based on multi-carrier modulation and a comparison with other UEP schemes [82].

Fig. 4.7 illustrates the considered transmission system. The transmitter section is basically made up of a channel coder followed by a bit loading unit that distributes the data bit, according to the algorithm implemented, among the subchannels (more details about its functions will be given further) and a conventional OFDM modulator. The OFDM scheme allows for the transmission of N parallel complex symbols A_n $(n = 1, 2, ..N)$, that belong to a M_n points constellation set $\{\pm 1, \pm 3, ..., \pm(\sqrt{M_n} - 1)\}$ for both real and imaginary dimensions, into N_C parallel subchannels (or subcarriers). The symbol (or frame) duration is denoted with T_s. Generally, only a limited, and constant over the subcarriers, set of values for $M_n = M$ is adopted in practical modems (e.g., $M = 2, 4, 16, 64$) [21].

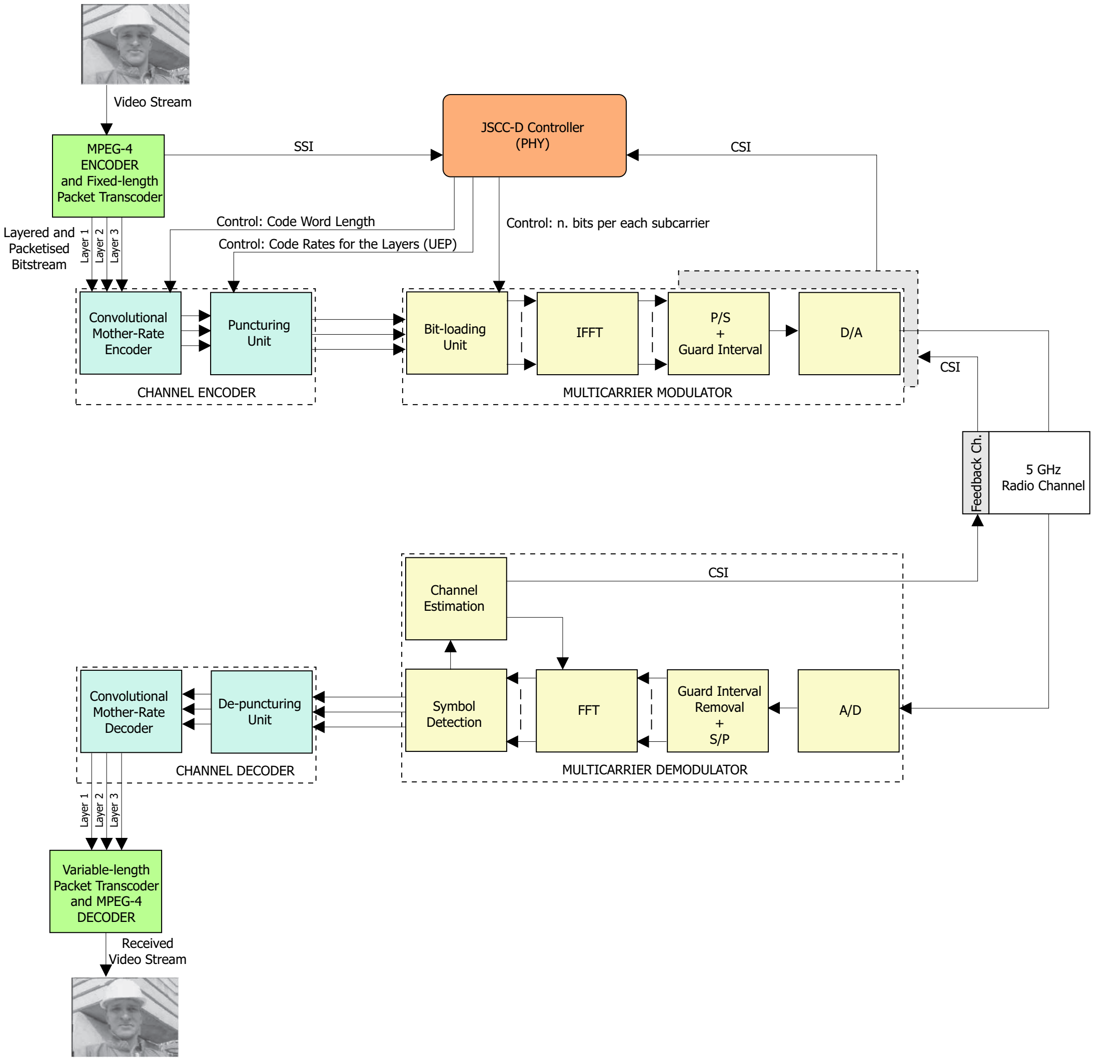


Figure 4.7: Transmission system block diagram.

In order to grant the orthogonality between subchannels in ideal channel conditions, the subchannel subdivision is obtained by means of an inverse fast Fourier transform (IFFT) of order N_{FFT} ($N_C < N_{FFT}$ to accommodate virtual subcarriers). Samples at the output of the IFFT block are converted from parallel to serial and transmitted every T_c seconds (chip time). In practice, due to propagation effects, subchannels still do not remain orthogonal, so a cyclic prefix (guard interval) is added to the OFDM symbol (the IFFT output) in order to remove the inter-symbol interference (ISI) among sub-channels [32]. Its duration is a multiple D of the chip time T_c, i.e., $T_g = D \cdot T_c$. At the receiver side, the reverse process is performed. The cyclic prefix represents a redundancy, in fact, only the time $T_u = N_{FFT} \cdot T_c$ is dedicated to the transmission of useful data, whereas the total OFDM symbol time is $T_s = T_u + T_g = T_c \cdot (N_{FFT} + D)$. The power efficiency (less than 1) due to the guard interval is

$$\eta_D = \frac{T_u}{T_s} = \frac{N_{FFT}}{(N_{FFT} + D)} \, . \tag{4.15}$$

If the maximum multipath delay T_d is less than the guard interval T_g, no ISI is present and the complex received signal at the n-th output of the fast Fourier transform (FFT) block can be written, in a normalized form, as [33]

$$Z_n = H_n \cdot w_n \cdot A_n + x_n \, , \tag{4.16}$$

where H_n is the channel transfer function gain related to the n-th subchannel, and w_n is a weight coefficient which allows non uniform power level allocation on the transmitter side as required by common bit loading algorithms. In not adapted schemes $w_n = 1 \; \forall n$. Following the same normalization done in [33], the random variable x_n represents the zero mean complex Gaussian thermal noise component at the n-th FFT output with power

$$\sigma_x^2 = E[|x_n|^2] = 2N_0/T_u \, , \tag{4.17}$$

where N_0 is the single side power noise density. Recalling that symbol A_n belongs to a $M_n - QAM$ constellation, the average power, P_n, dedicated to the n-th subchannel is

$$P_n = E[|A_n|^2] \cdot w_n^2 = \frac{2(M_n - 1)\, w_n^2}{3} \, , \tag{4.18}$$

leading to a total average transmitted power P_T

$$P_T = \sum_{n=1}^{N_u} P_n, \tag{4.19}$$

where N_u is the actual number of subchannels used by the bit-loading algorithm. We have neglected here the presence of pilot sub-carriers allocated for channel estimation purposes.

When a M_n-quadrature amplitude modulation (QAM) signaling is adopted, assuming ideal phase offset compensation, perfect carrier recovery and synchronization, the bit error probability related to the n-th subchannel can be approximated as follows [34]

$$P_{b_n} \cong \frac{2(\sqrt{M_n} - 1)}{\sqrt{M_n} \cdot \log_2 M_n} \cdot erfc\sqrt{\frac{w_n^2 \cdot |H_n|^2}{\sigma_x^2}} \, . \tag{4.20}$$

Considering (4.17) (4.18) and (4.19) we obtain

$$P_{b_n} \cong \frac{2(\sqrt{M_n}-1)}{\sqrt{M_n}\cdot\log_2 M_n}\cdot erfc\sqrt{\frac{P_n\cdot 3|H_n|^2}{2(M_n-1)\sigma_x^2}} = \tag{4.21}$$

$$= \frac{2(\sqrt{M_n}-1)}{\sqrt{M_n}\cdot\log_2 M_n}\cdot erfc\sqrt{\frac{E_s}{N_0}\frac{3\,\varepsilon_n\cdot|H_n|^2\cdot\eta_D}{2(M_n-1)}}, \tag{4.22}$$

where

$$\frac{E_s}{N_0} = \frac{P_T\cdot T_s}{2N_0} \tag{4.23}$$

denotes the transmitted (OFDM) radio frequency symbol-energy-to-noise ratio, and $\varepsilon_n = P_n/P_T$ indicates the fraction of the power dedicated to the n-th subchannel. Obviously, it is $\sum_{n=1}^{N_u}\varepsilon_n = 1$. Once the code rate, R_c, and the actual number of bit transmitted per frame, b_T, are fixed, $\frac{E_s}{N_0}$ can be expressed as a function of the received average bit-energy-to-noise ratio $\frac{E_b}{N_0}$

$$\frac{E_s}{N_0} = \frac{E_b}{N_0}\,R_c\,b_T\,. \tag{4.24}$$

The performance at each subchannel depends on $|H_n|$, so severely attenuated subchannels could compromise the performance. In general, a suitable channel coding is necessary to improve the overall performance (Coded OFDM) [33].

4.5.1 Ordered subcarrier selection algorithm

Current wireless local area network (WLAN) standards [21][22], consider a fixed bit loading scheme where, once the decision on the constellation size M based on overall propagation conditions has been made, all subchannels ($N_u = N_C$) utilize the same size M ($M_n = M$) and the same power fraction ($\varepsilon_n = 1/N$), independently by the single subchannel condition. In the following, this case is referred to as the *reference scheme* (conventional OFDM scheme with no adaptation). The total number of bits transmitted by every OFDM symbol time, T_s, is $b_T = N_C\ log_2(M)$.

The basic principle of adaptive modulation techniques is the opportunity of dynamically modifying the modulation parameters (M_n, ε_n and N_u) according to the time-variant channel conditions [23]. This can be accomplished efficiently if the transmitter knows the channel state. A feedback channel should thus be available, as shown in Fig. 4.7, in order to pass the channel state information to the transmitter. The rate of CSI depends on the channel variability, in particular on the channel coherence time. Common adaptive schemes require that each subchannel be loaded using a particular constellation size M_n and fractional power level ε_n, different from that allocated in the other subchannels [25]. The optimal set for ε_n and M_n, that maximizes the power margin, is given by the Campello's conditions [26]. In those cases, all source bits are assumed to have the same importance (EEP).

These algorithms lead to a high level of modem complexity and the necessity to provide a large signaling overhead in time-varying wireless channels, especially in high-speed systems. To partially overcome this problem, some techniques appeared in the Literature; Grunheid et al. [44] propose a simplified scheme where the optimization is performed with a blockwise allocation of modulation levels. In [45] it is shown that a constant power allocation scheme has a negligible performance loss compared to the optimal solution.

In order to obtain low complexity modems, we propose a modified scheme transmitting the same amount of bits b_T as in the reference scheme, but where only a subset $N_u \leq N_C$ of the available N_C subchannels is effectively used. Now, the actual constellation size has to be suitably increased in order to allocate all the b_T bits, i.e.,

$$M_n = M = 2^{b_T/N_u} , \quad n = 1, 2 ... N_u . \tag{4.25}$$

Obviously, only a limited number of values for N_u is allowed if we want the constellation size M to result in a practical integer value. The total transmitted power is uniformly distributed among the N_u used subchannels as a consequence ($\varepsilon_n = 1/N_u$).

The basic idea of the *ordered subcarrier selection algorithm* is to select only the strongest N_u subchannels (i.e., the subchannels characterized by a higher value for $|H_n|^2$) and to use higher constellation sizes by keeping the total bit rate and transmitted power unchanged. In our approach, both the power level and the constellation size are kept constant over the selected set of subchannels. The receiver's task is to estimate the channel gain, H_n, select the N_u strongest (most reliable) subchannels and, through the feedback channel, inform the transmitter which to use in the next packet transmission. It is to be pointed out that the feedback throughput required is very limited compared to that required by common bit loading algorithms [24].

To find which choice for N_u gives good results, we considered the analysis of the bit error probability in [47], obtained by the proposed algorithm, in the case where all subchannels are affected by independent Rayleigh fading. The average bit error probability is minimized for $N_u = N/2$ if $M = 4$ (i.e., when only half subchannels and quadruplicated constellation size are used) and $N_u = 2N/3$ if $M = 16$. This result shows that the optimum choice of the number of active subchannels, N_u, does not depend on the actual instantaneous SNR but only on the long-term overall channel statistics (in this case, Rayleigh fading).

4.5.2 Extension to the Multi-Layer case

We now extend the considered bit loading algorithm to the multi-layer case, where several data streams must be transmitted simultaneously with different performance requirements (UEP) as typical in multimedia applications. In this case, the total number of subchannels is divided into N sets (the number of layers), each one, denoted with $C(l)$, is associated to a different layer. The bitstream, with bit-rate B_{r_l}, associated to each layer is required to have a specific target bit error probability P_{b_l}. The problem is to find the optimal set of parameters $\{C(l), M_n, \varepsilon_n\}$ ($l = 1, 2, .., N$ and $n = 1, 2.., N_C$), given P_{b_l}, B_{r_l} and the channel state H_n, that minimizes the total transmitted power P_T. The optimization problem is NP-hard [46] and some sub-optimal algorithms are present in the Literature [27][31][46]. They require the knowledge of the relationship between the video quality, in terms of PSNR (see below) or subjective measures, and the correspondent bit error probability P_{b_l} required for each layer. However, as we have seen in the previous section, this relationship is not easy to find as it requires extensive simulation or, alternatively, a model valid in general conditions.

Thus, we have investigated a more simple sub-optimal scheme capable of realizing unequal error protection at modulation level. It is based on the above mentioned adaptive ordered subcarrier selection algorithm, where UEP is simply achieved by assigning the bits, belonging to each layer, to subchannels starting from the most reliable down to the least reliable. It must be highlighted that the ordered subcarrier selection algorithm minimizes the overall average bit error probability. However, the layered bit assignment described above leads to an unbalanced average bit error probability between different layers data streams, since bits belonging to more important layers are more protected due to the ordering process.

4.5.3 Fixed-packet-length-transcoding assisted UEP

Proportional unequal error protection
[58] is a useful technique to avoid transmitting side information when in presence of a source bitstream with variable packets and partitions length. It is evident that, with the application of this proportional unequal error protection (P-UEP) scheme, one of the

most critical aspects is the detection of start codes (SCs), which identify the beginning of video packets. In fact, if we fail the identification of one of these synchronization points, we obtain an incorrect value for the length of the received packet and, as a direct consequence, the size of the differently protected partitions within the packet are also wrongly calculated.

In [60], a technique to make sufficiently robust the start codes detection has been proposed. It has been suggested to substitute the original MPEG-4 start codes with other sequences more reliably detectable in a particularly noisy environment, as a wireless channel.

Here an alternative technique, mainly based on the re-organization of the bitstream in packets of a predetermined fixed length is proposed. This technique may be useful to increase the bitstream error resilience and to enable a direct application of unequal error protection in any case of source bitstreams composed of packets and partitions of different lengths. The description of the technique is given in the following with reference to the MPEG-4 video coding standard.

4.5.4 FPT-UEP

The technique proposed allows to provide fixed length packets to the lower protocol layers, and to univocally reconstruct from the corrupted version of these packets the corresponding MPEG-4 compliant packets to be fed to the MPEG-4 decoder.

The proposed fixed-length packet transcoding unequal error protection (FPT-UEP) technique, as shown in fig.4.8, is structured in four steps:

1. SC substitution;
2. append of stuffing bits;
3. partitioning of the packet and insertion in different queues;
4. assembling of the fixed-length packet.

In the following, these points will be described in more details.

The original MPEG-4 start codes are substituted with other codewords more robust to channel errors. In particular, the new start codes are chosen in order to avoid their emulation in the bitstream even if some bits have been corrupted during the transmission.

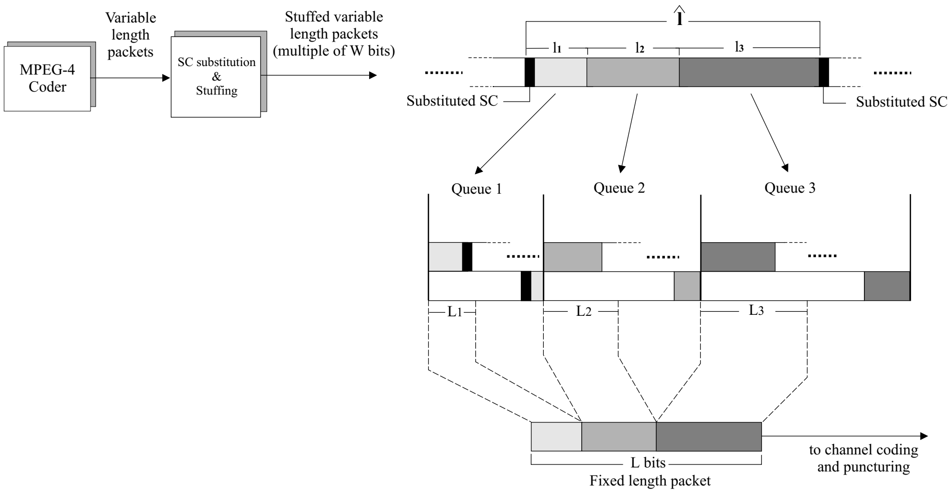


Figure 4.8: Fixed-length-packet transcoding (N=3).

The resultant length of the packets is indicated with l.

A sequence of s stuffing bits with value '1' is appended at the end of the packet, so that

$$\hat{l} = l + s = mW \tag{4.26}$$

where $\hat{l}$ is the length of the "stuffed" packet, m is an integer and W is an appropriate number of bits, whose meaning will become clear in the following. In other words, the stuffing bits are aimed at making the length of the packet multiple of W bits.
The packet is decomposed in N partitions in a proportional way according to the coefficients

$$p_i = \frac{w_i}{W} \tag{4.27}$$

with $w_i \in \{1, 2, ..., W\}$, and under the constraint

$$\sum_{i=1}^{N} p_i = 1 \tag{4.28}$$

or, equivalently,

$$\sum_{i=1}^{N} w_i = W. \tag{4.29}$$

The bits belonging to the N different partitions are then inserted in N distinct queues, scheduled according to FIFO rules.

From the SC queues, the bits are taken in correct proportions in order to build a packet of the fixed length desired. The overall size L of the fixed-length packet may be any multiple of W, i.e.

$$L = q \cdot W \tag{4.30}$$

where q is a positive integer. The packet results composed of N distinct parts, each containing the bits from the corresponding queue. The i^{th} partition has length $L_i = p_i \cdot L$, so that

$$L = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} p_i \cdot L. \tag{4.31}$$

The fixed-length packet is then coded and punctured if needed. As a result, the bitstream transmitted is a regular sequence of fields of different importance, and their lengths are now fixed. In this way, the decoding process may be correctly performed with the exact knowledge of partitions lengths; if punctured codes are used, the de-puncturing process will be always correctly performed: in fact, the knowledge of L and of the fixed coefficients p_i is sufficient to determine exactly the portions of the bitstream where to apply the different de-puncturing matrices.
At the receiver side, an inverse algorithm allows the reconstruction of the original packets. N memory buffers are required. For $i = 1, ..., N$, L_i bits are taken from the generic packet decoded and inserted in the i^{th} buffer. Then the start codes detection is performed. Thanks to the technique described above, all the start codes belong to the first partition. As a consequence, their search can be limited to the first buffer: in other words, the processing window is formed by $p_1 \cdot L$ bits every L bits received. The search is performed through hard correlation between copies of each SC and the decoded bits contained in the first buffer. The output of the correlator is then compared with a threshold, initially set equal to the SC length, so that the process is equivalent to searching the SCs by identity. It follows that, if one or more bits have been corrupted during the transmission, the SC is not immediately detected. In this case, the search can consider a lower threshold. It is now evident the convenience of substituting the original SCs with others, less easily emulated by the MPEG bitstream. The re-synchronization tool included in MPEG-4 permits to define the mean value for the length of the VPs. Thanks to this property, it is always possible to determine a maximum dimension for the variable-length packets (i.e. a size they never exceed), which is never extremely different from the mean value specified during the source coding process. Calling $\hat{l}_{max}$ the maximum length after the SCs substitution and the stuffing process, we know that a SC has been surely skipped when we have processed $\hat{l}_{max} \cdot p_1$ bits in the first buffer without detecting any SC. In this

case, the threshold is lowered of a prefixed quantity and the search is done through the last $\hat{l}_{max} \cdot p_1$ bits. This process goes on until a SC has been detected and, considering hard correlations, this is equivalent to admit a number of errors in the searched sequence increasing of 1 at each step.

From the knowledge of the position of the SCs, we can determine the size of the first partition of the VPs. We may thus compute the overall length of the packet, according to

$$\hat{l} = \frac{l_1}{p_1} \tag{4.32}$$

and the lengths of the other partitions, contained in the different buffers:

$$l_i = p_i \cdot \hat{l} = p_i \cdot \frac{l_1}{p_1} \tag{4.33}$$

.

With the knowledge of these lengths, an equivalent MPEG-4 packet may be reconstructed: after the substitution of the original start codes and de-stuffing, the bitstream obtained is MPEG-4 compatible.

The stuffing bits added at the transmitter side are then removed from the packet. We observe that the stuffing and the correspondent de-stuffing (at the receiver side) are allowed by the characteristics of the compression standard considered. All the added bits, in fact, have to be recognizable at the receiver, in order to be eliminated. According to the standard, MPEG-4 variable-length packets are multiple of 8 bits: if not the particular stuffing sequences listed in tab. (4.5.4) are appended, according to the number of bits required to reach that multiplicity. As evident from the table, a direct consequence is that the last byte of the packet contains always a 0, so it is easily recognizable during the inverse process of elimination of the stuffing 1's. In practice, the de-stuffing consists in truncating the packet to the last byte containing a 0.

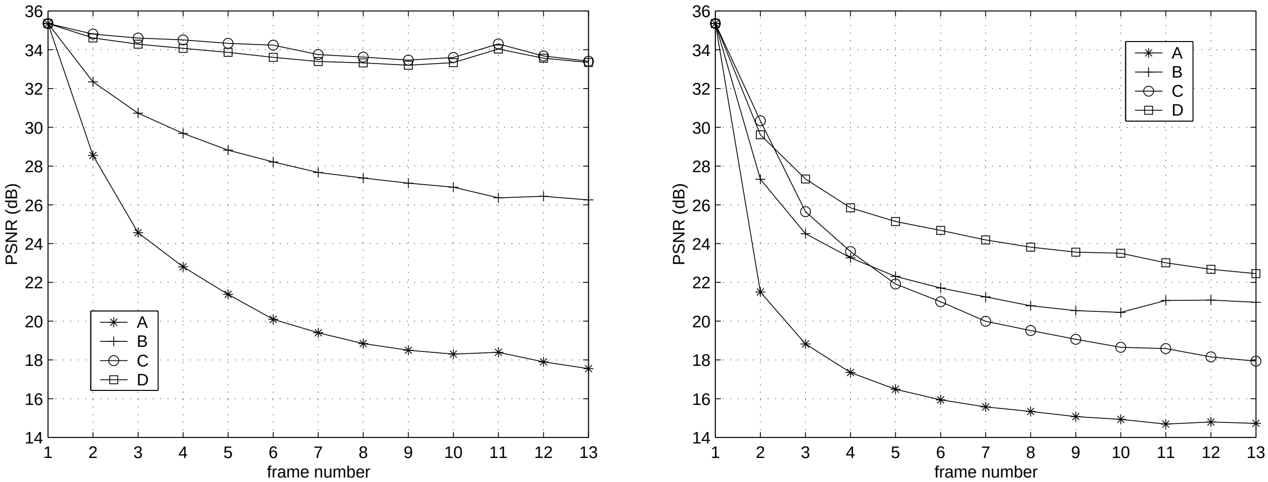
4.5.5 Source and transmission parameters

Here we will show the results obtained from the described scheme. We we coded according to the MPEG-4 standard the first 13 frames of a video sequence (the "foreman" test sequence in CIF format) at a bit rate of 644 kbit/sec. The FPT-UEP technique described in has been considered for the reorganization of the bitstream in packets with fixed length and made of fixed length partitions (Fig. 4.8).

In this work, we have organized the bitstream in packets made of 432 bits, with 27 bits for the first portion of the packet, containing start codes and headers, 108 bits for

Table 4.2: MPEG-4 stuffing bits.

Number of required bits	**MPEG-4 stuffing sequence**
1	0
2	01
3	011
4	0111
5	01111
6	011111
7	0111111
8	01111111



(a)

(b)

Figure 4.9: Performance comparison between schemes A-B-C-D in terms of PSNR(dB) as a function of the frame number for $E_b/N_0 = 11$ dB (a), $E_b/N_0 = 7$ dB (b). EEP channel coding.

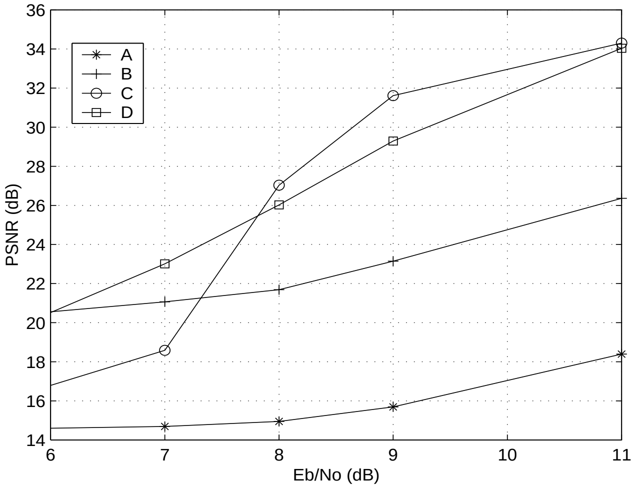


Figure 4.10: PSNR (dB) of frames I v.s. signal-to-noise ratio (E_b/N_0) for schemes A,B,C and D. EEP channel coding.

the second portion of the packet, containing data relevant to the first data partitions, and 297 bits for the last portion, containing data relevant to the second data partitions. Consequently, the unequal protection schemes considered in the paper, both through modulation and through channel codes, refer to a fixed packet structure.

When the UEP is realized at channel coding level, the following coding rates are used for each layer: $R_{c_1} = 1/3$, $R_{c_2} = 8/21$, $R_{c_3} = 8/13$, for an average code rate of $R_c \simeq 1/2$. For a fair comparison, when EEP is adopted or UEP is implemented at modulation level, the coding rate is kept constant for all layers to $R_c = 1/2$ as well. rate compatible punctured recursive systematic convolutional (RCPRSC) codes with rational systematic generator matrix $Gs(D) = (1, R1(D) = (1 + D + D^2 + D^4)/(1 + D^3 + D^4)$, $R2(D) = (1 + D^2 + D^3 + D^4)/(1 + D^3 + D^4))$ and puncturing period P=8 are used [77].

Moreover, we assume that the first frame is received error free, in order to conceal the subsequent frames; we may in fact retransmit the frame if any errors occur, since a small delay may be tolerated at the beginning of the bitstream.

Without loss in generality, system parameters are taken from the IEEE802.11a physical layer specifications (or HIPERLAN2) [21, 22]: $T_s = 4\mu s$, $N_{FFT} = 64$, $N_C = 48$, $T_g = 800$ ns, subcarrier spacing $\Delta f = 312.5$ KHz. In this case $\eta_D = 0.8$.

The total system capacity is kept constant at $b_T/T_s = 24$ Mb/s. Since the average code rate is $R_c = 1/2$, the final useful bit rate becomes $B_r = 12$ Mb/s. The transmission of one packet requires 10 OFDM frames. As the fixed total bit-rate is here 12 Mbit/s,

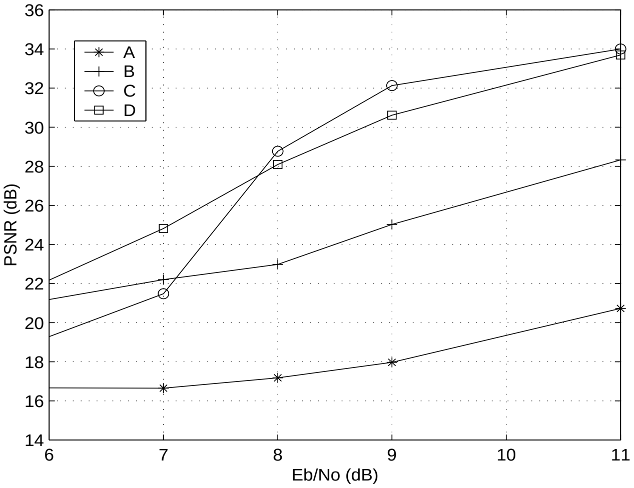


Figure 4.11: PSNR (dB) of frames P v.s. signal-to-noise ratio (E_b/N_0) for schemes A,B,C and D. EEP channel coding.

we supposed to send a packet every $678\mu s$, considering others multimedia streams to be transmitted in the remaining time.

As far as the channel model is concerned, we refer to the 5Ghz ”E” European Telecommunication Standards Institute (ETSI) channel model [61] (outdoor in non line-of-sight condition) characterized by 18 Rayleigh fading paths. The channel is assumed invariant during the transmission of each packet.

The optimization (bit-loading) is performed, according to the temporal evolution of the channel, every T_{csi} seconds, supposing that the channel state information is sent with the same rate. It is advisable that $T_{csi} < T_{ch}$, where T_{ch} is the channel correlation time. In [47] it has been shown that no significant performance degradation is present if $T_{csi} < 7 - 10$ ms, in the case the user moves with a maximum speed of 3Km/h.

4.5.6 Numerical results and comparison of different techniques

The PSNR is averaged over the nine P frames of the first GOV and the first four frames of the second GOV. Results of thirty simulations, performed with different noise seeds, have been averaged in order to obtain more reliable results. The average PSNR is thus

$$PSNR_{avg} = \frac{1}{N_s N_f} \sum_{s=1}^{N_s} \sum_{f=1}^{N_f} PSNR(s,f)\,, \tag{4.34}$$

where N_s is the number of simulations performed, N_f the number of frames considered in the average, s and f the simulation index and the frame index.

In the following we consider the bit-loading schemes below:

- Scheme A) reference scheme (no adaptation)
- Scheme B) adaptive ordered subcarrier selection algorithm with $N_u = N$ (UEP through carrier re-ordering)
- Scheme C) adaptive Campello's algorithm [26]
- Scheme D) adaptive ordered subcarrier selection algorithm with $N_u = N/2$ (UEP through carrier re-ordering).

It is worth noticing that scheme C (Campello) is the optimal bit loading solution but it does not offer the possibility to perform UEP, since the bit error rate is the same for all subchannels. On the contrary, proposed schemes B and D are sub-optimal but they allow the possibility to perform UEP at modulation level due to the subchannel ordering process. Scheme A (no adaptation) is considered for comparison. Also in this case UEP cannot be performed at modulation level. The different schemes considered are reported for better clarity in Table 4.3.

In Fig. 4.9, the evolution of the PSNR as a function of the frame number is shown for schemes A,B,C and D. The case $E_b/N_0 = 11$ dB is reported in Fig. 4.9(a) where, comparing curves A and C, it is possible to have an idea of the large gain obtainable by the introduction of adaptive loading algorithms. The same gain is achieved with the simpler scheme D, proposed herein, that employs the UEP at modulation level. The benefit of UEP becomes more evident at lower SNR, as shown in Fig. 4.9(b) for $E_b/N_0 = 7$ dB, where both schemes B and D, that implement UEP at modulation level, overcome scheme C. In any case, the gain with respect to the reference scheme (scheme A) remains still remarkable (about 7 dB).

The PSNR against E_b/N_0, related to I frames only, is shown in Fig. 4.10. It is interesting to highlight the crossing point between curves C and D referring to Campello's (EEP) and the ordered subcarrier selection algorithms, respectively; at low SNR values the UEP makes scheme D more robust than scheme C, even though scheme D is simpler and it is sub-optimal in single layer systems. In fact, the gain due to UEP of scheme D at low SNR values is able to compensate for the loss due the sub-optimality of bit loading. For better channel conditions (high SNR values), the UEP benefits decreases and cannot

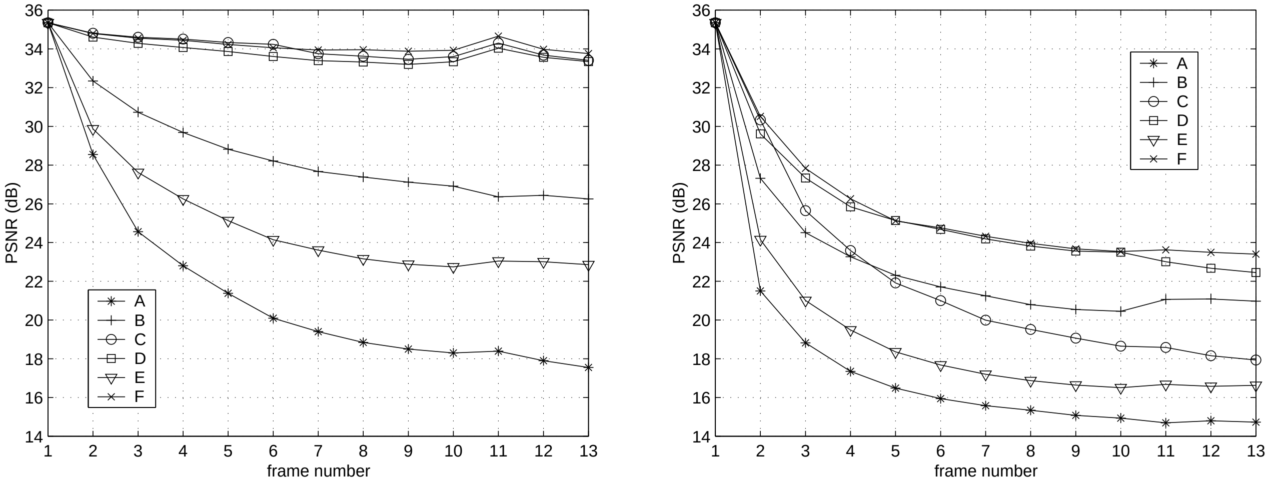
A	Reference scheme: no adaptation. No UEP.
B	Adaptive ordered subcarrier selection algorithm, $\mathbf{N_u} = \mathbf{N}$ (UEP through subcarrier re-ordering).
C	Adaptive Campello's algorithm. No UEP.
D	Adaptive ordered subcarrier selection algorithm, $\mathbf{N_u} = \mathbf{N/2}$ (UEP through subcarrier re-ordering).
E	Reference scheme: no adaptation. UEP at channel coding level.
F	Adaptive Campello's algorithm. UEP at channel coding level.

Table 4.3: Schemes considered.

compensate for the sub-optimality loss of the bit loading algorithm. A similar behavior can be seen in Fig.4.11 regarding P frames.

The impact of UEP realized through channel coding is illustrated in Figures 4.12(a) and 4.12(b) for $E_b/N_0 = 11$ dB and $E_b/N_0 = 7$ dB, respectively. Schemes E and F are the same as A and C but with UEP realized at channel coding level with the specified code rates. Schemes A-D are reported for comparison. At high SNR levels (Fig. 4.12)(a) the impact of UEP is significant only when applied to the reference scheme (curve E), but does not give any significant improvement to scheme C (curve F). On the contrary, at lower values for SNR (Fig. 4.12)(b), the performance obtained with scheme F becomes comparable to the performance of scheme D, where the UEP is realized at modulation level. It should be remarked that scheme D is much less complex that scheme C.

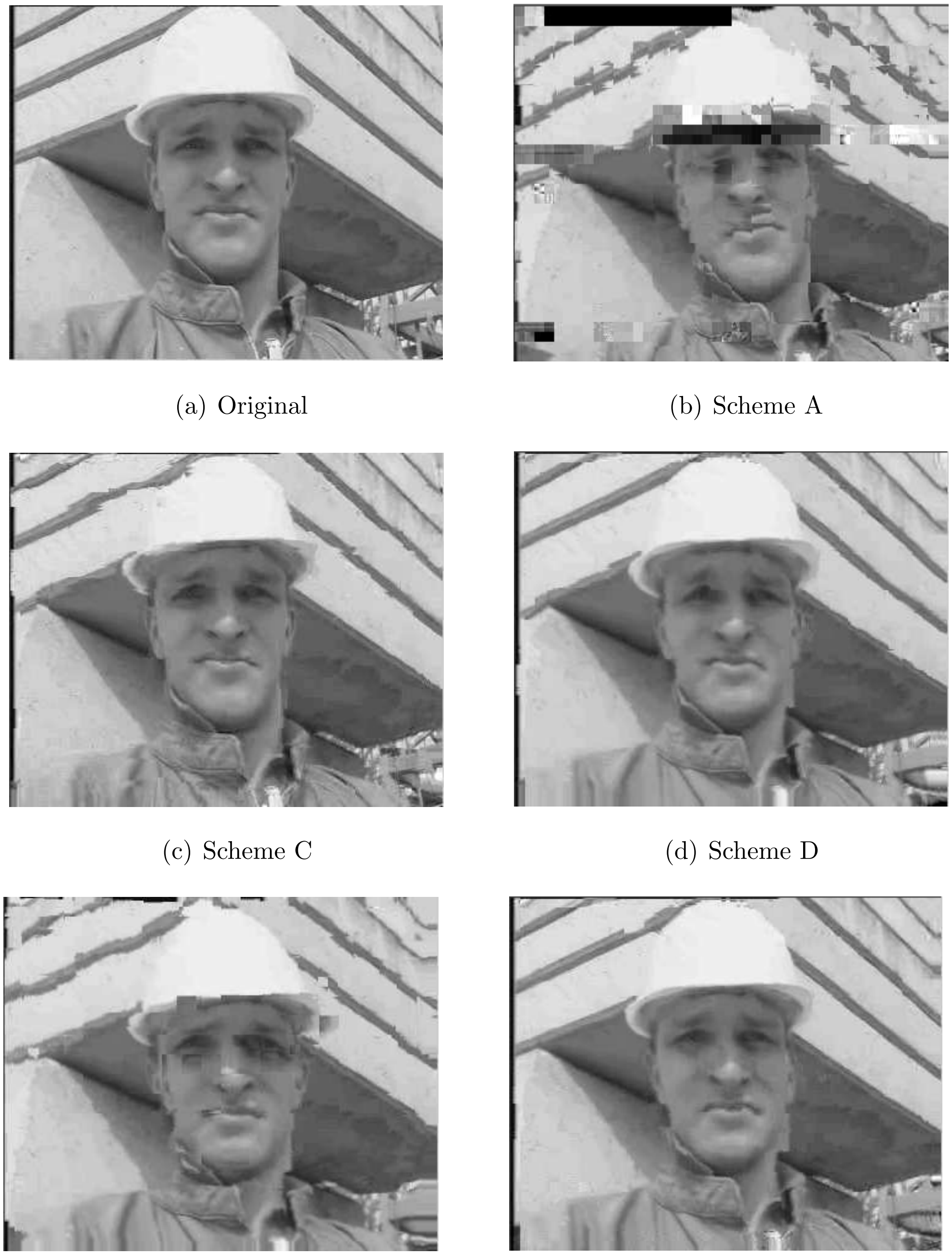
The visual quality improves too with the adaptive loading technique considered, as shown in Fig. 4.13, where the received frame number 8 of the foreman sequence is reported for schemes A,C,D,E and F in the case of $E_b/N_0 = 7$ dB. The quality improvement is evident, above all, for schemes D and F.



(a)

(b)

Figure 4.12: Performance comparison between schemes A-F in terms of PSNR(dB) as a function of the frame number for $E_b/N_0 = 11$ dB (a), $E_b/N_0 = 7$ dB (b). Schemes E and F are the same as A and C but with UEP realized through channel coding.



(a) Original

(b) Scheme A

(c) Scheme C

(d) Scheme D

(e) Scheme E

(f) Scheme F

Figure 4.13: Frame (P) no. 9 of the foreman sequence. $E_b/N_0 = 7$ dB.

Chapter 5

Application scenarios and results

In this chapter we present the simulation chain developed during the *Phoenix* project, allowing to test many of the techniques described in the previous chapters. Results relevant to various application scenarios are then reported, highlighting the improvements that a joint coding/modulation approach determine.

5.1 Phoenix simulator

During the *Phoenix* project development, two different simulation chain have been realized by the consortium: the first one was based on distinct chain blocks exchanging video data and side information through files; the second one was realized as an integrated executable, whose parts were provided by the *Phoenix* partners. The development of the simulators required intense work and strict collaboration among partners across all Europe. In Fig. 5.1 and 5.2 we have reported the block scheme describing the data and information flow across the communication chain at the transmitter and the receiver side respectively [65]. All the connections among the blocks have been defined in detail through proper interfaces definitions. In particular the type of information or data exchanged, the input and the output connections and the number of bits codifying each parameter have been defined through tables reported in [65]. Examples of the interface definition work have been reported in Table 5.1 and 5.2.

The main features implemented in the simulation chains are:

- application layer controller, also including an optional content level UEP block (using RCPC codes);

- source encoder/decoder (three possible codecs: MPEG-4, H.264/AVC and Scalable

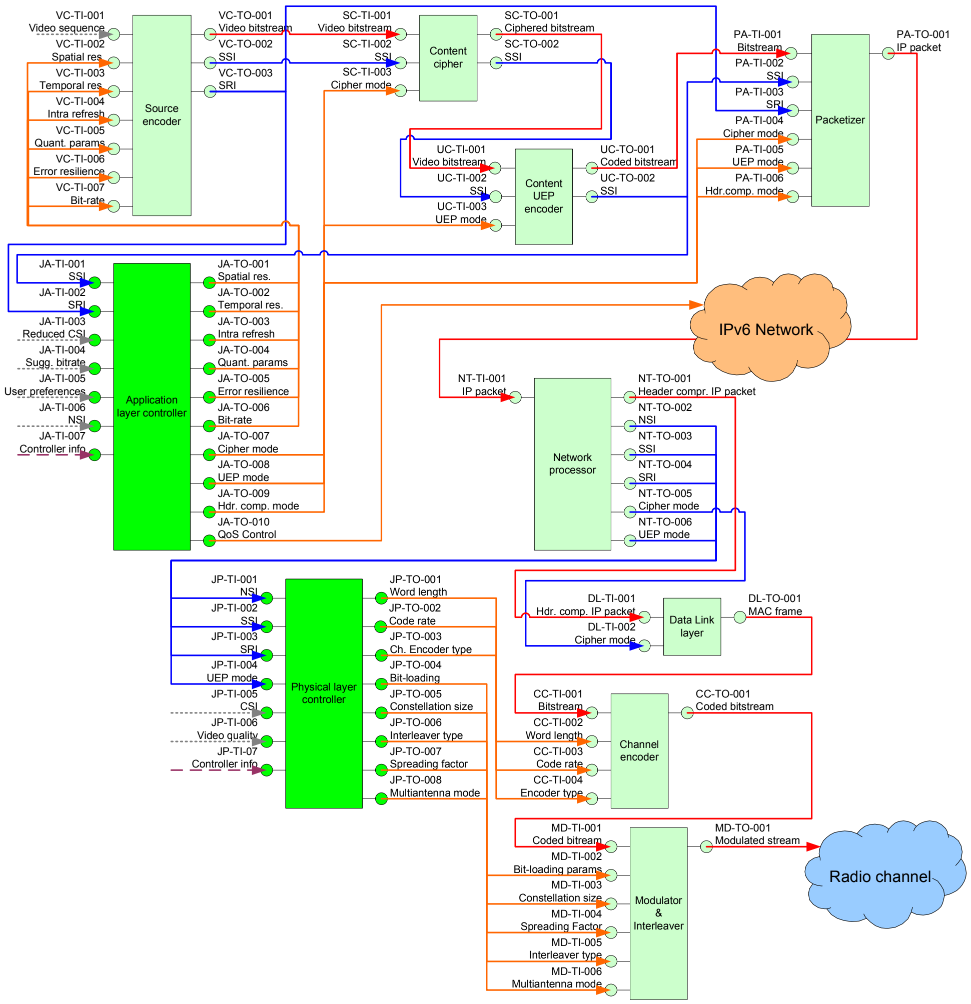


Figure 5.1: Block scheme of the *Phoenix* simulator at the transmitter side.

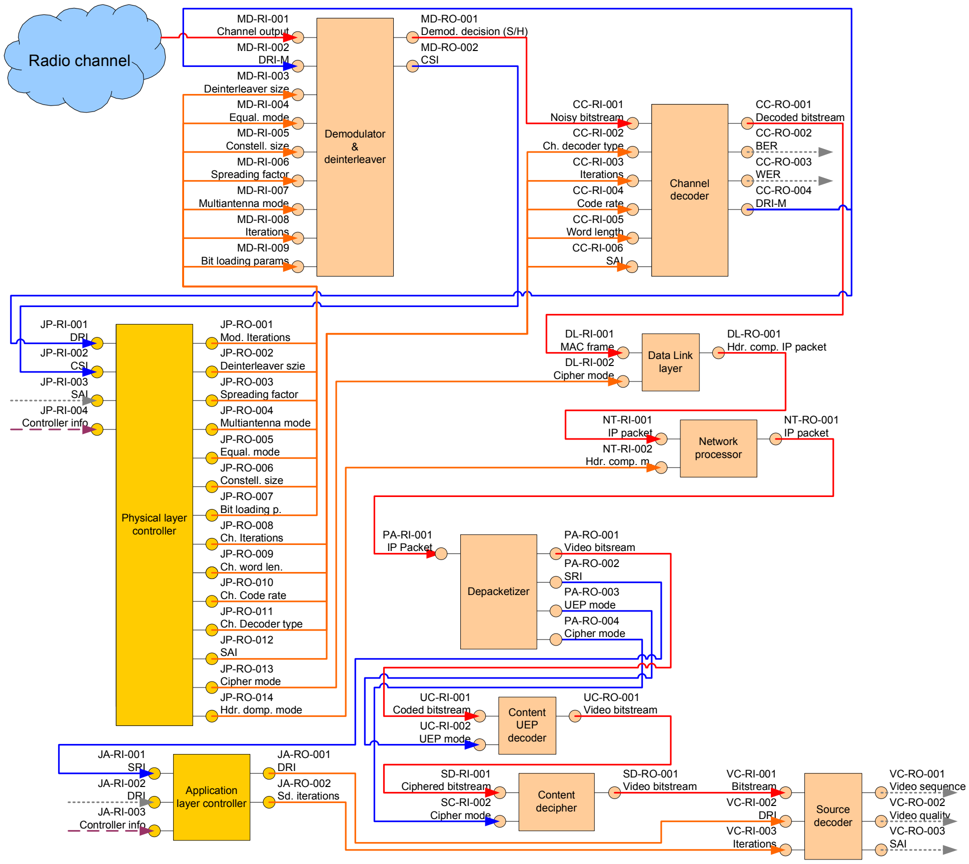


Figure 5.2: Block scheme of the *Phoenix* simulator at the receiver side.

Type of channel encoder to be used	
ID	CC-TI-004
Input/output	Input - from the physical layer controller
Size	{8 bits} 5 bits
Values	0: no channel encoders [bits 1 to 9 are reserved to RCPC codes] 1: RCPC with k=1, n=3, K=5 (memory 4), puncturing period 8 2: RCPC with k=1, n=5, K=6 (memory 5), puncturing period 8 3: RCPC with k=1, n=2, K=9 (memory 8), puncturing period 8 4: RCPC with k=1, n=3, K=9 (memory 8), puncturing period 8 5: RCPC with k=1, n=2, K=7 (memory 6), puncturing period 8 ... [bits 10 to 19 are reserved to LDPC codes] 10: LDPC IRA code, mother code: n=2100, possible Rc: 1/3, 1/2,2/3, 3/4, 5/6 11: LDPC IRA code, mother code: n= 4200, possible Rc: 1/3, 1/2,2/3, 3/4, 5/6 ... [bits 20 to 29 are reserved to Turbo codes] 20: Turbo code ...
Description	We may have the possibility to choose between soft or hard decoding, MAP or MLSE decoding, *etc*
Applicability	When different channel encoders are available at the transmitter

Table 5.1: Example of interface definition at the transmitter side.

Number of iterations of the channel decoder	
ID	CC-RI-003
Input/output	Input
Size	4 bits
Values	>1
Description	Max. number of iterations in channel decoding Used with LDPC or Turbo codes. Nota: according to the trade-off between complexity and performance evaluated by the controller, the required number of iteration may be provided to the channel decoder by the Phy JSCC controller
Applicability	LDPC, Turbo codes.

Table 5.2: Example of interface definition at the receiver side.

Video Coding in H.264/AVC Annex F), where soft-input source decoding is also allowed for H.264/AVC;

- cipher/decipher unit;
- RTP header insertion/removal;
- transport protocol header (*e.g.* UDP-Lite, user datagram protocol (UDP), or DCCP) insertion/removal;
- IPv6 header insertion/removal;
- IPv6 mobility modelling;
- IPv6 network simulation;
- Robust Header Compression (RoHC);
- DLL header insertion/removal;
- radio link, including
  - physical layer controller,
  - channel encoder/decoder (convolutional, RCPC, LDPC codes with soft and iterative decoding allowed),
  - interleaver,
  - modulator (OFDM, TCM, TTCM, STTC), with soft and iterative demodulation allowed,
  - several simulated channels (e.g. AWGN, Rayleigh fading, shadowing, frequency selective channels).

Our work focused mainly on the development of the App JSCC Controller for MPEG-4 sequences, the Phy JSCC Controller, MPEG-4 co-decoding and the radio link block. In particular, with regard to the radio link, we have worked on the channel coding aspects, the OFDM modulation and on the radio channels. Iterative soft channel decoding/demodulation have been tested with the mo-dedulators provided by the University of Southampton.

5.2 JSCC results in a WLAN scenario

Fig. 5.3 shows a performance comparison among examples of video transmissions adopting different strategies [89], from the “classical” scheme to the proposed joint-adaptive techniques at application and physical layer, as described in chapters 3-4. The comparison is made in terms of subsequent video quality values, each obtained through the average over 1 s of 5 simulations run with different noise seeds. The values obtained have been normalized (by setting the maximum value to 1), in order to allow a direct comparison between different quality indexes: the PSNR and the SSIM [52]. The scenario considered is a WLAN supporting, at the radio link level, a coded bitrate of 12 Mbit/s. The video stream is coded according to the MPEG-4 standard and is supposed to be multiplexed with other real-time transmissions so that it occupies only an average portion of the available bandwidth, corresponding to a coded bitrate of 1 Mbit/s. The Foreman sequence in CIF resolution has been selected according to the considered scenario. The App controller states, representing the possible quantization parameters, frame rate, and GOP rate, are those represented in Fig. 3.2. The channel code is the IRA-LDPCC described in chapter 4, with a “mother” code rate of (3500,10500), properly punctured and shortened in order to obtain different code rates. Thus, the resulting codewords are 4200 bits long. The code rate is fixed for the non adapted system (meaning that an EEP policy is adopted) and its value is determined according to the constraint on the maximum allowed coded bitrate (i.e. 1 Mbit/s). On the contrary, in the adapted case the code rate can change according to the SSI in order to perform UEP. In the reference transmission scheme, the modulation is OFDM with 48 carriers for data transmission and a frame duration of 4 μs, whereas the Phy JSCC also applies margin adaptive bit-loading. The simulated channel is obtained according to the ETSI standard channel A [22], and it takes into account also a log-normal flat fading component with channel coherence time of 5s, to consider the fading effects due to large obstacles. The figure is related to $E_b/N_0 = 11.2dB$, where E_b is the energy per coded bit.

From the curves in Fig. 5.3 the gain achieved with JSCC/D controllers and with RoHC is evident. The RoHC tool, based on the IETF RoHC recommendations and realized by *Thales*, France, allows for an efficient compression of the RTP/UDP-Lite/IPv6 headers for the transmission through the radio channel. The header size reduction determines a large increase in terms of robustness to errors, as shown by the results, since more protection can be provided to headers and data if the header size is reduced. In the reference system, no RoHC is adopted, nor adapting algorithms. When RoHC is enabled, it allows to reduce

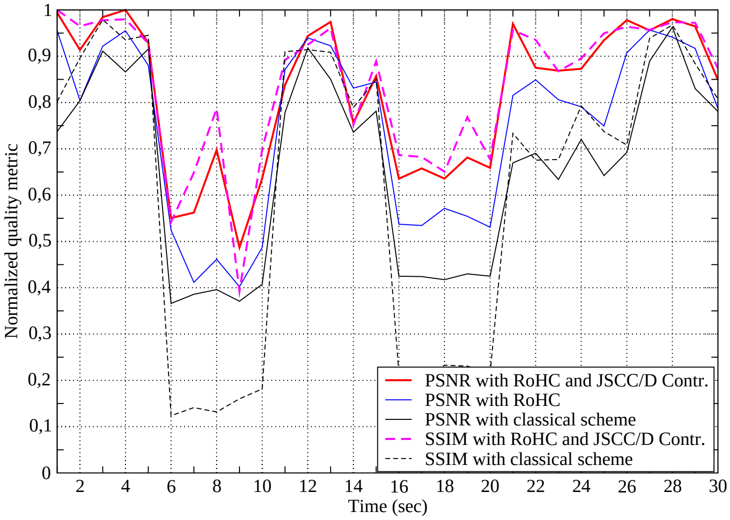


Figure 5.3: Received video quality versus time with the JSCC adapted and non adapted system. Foreman sequence in CIF format. MPEG-4 source encoding.

the overhead due to the header transmission and a channel code with a lower code rate can be applied, determining an average PSNR improvement of 2.73dB. If both RoHC and JSCC/D controllers are enabled the improvement is particularly evident, with any considered metric. In particular, we observed an average gain of 5dB in terms of PSNR in the conditions under analysis.

Visual results, in accordance with average visual impact, are reported in Fig. 5.16, where as an example the received frame no. 435 for the CIF Foreman sequence is reported, both for the adapted and the non-adapted case. The original frame is also reported for

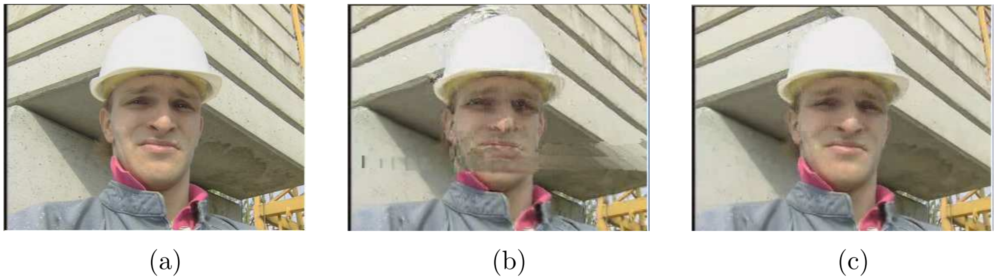


Figure 5.4: Visual results. [a] Frame no. 435 of the Foreman sequence - original; [b] Frame no. 435 of the Foreman sequence - MPEG-4 - no RoHC, no JSCC; [c] Frame no. 435 of the Foreman sequence - MPEG-4 - RoHC, JSCC.

comparison.

5.3 Scenarios envisaged in the Phoenix project

Within the *Phoenix* project, several different application scenarios have been identified, in order to emphasize the interest of the end-to-end optimization strategies developed by the consortium for the transmission of multimedia data over an IP wireless link. In particular, seven scenarios, as listed in the following, were envisaged and for each scenario the *Phoenix* consortium identified realistic sets of parameters for the different blocks of the transmission chain. The seven scenarios that have been identified aim at providing a wide range of applications in which the end-to-end optimization of the video transmission over an IP wireless link can help improve the transmission in a significant manner.

All the partner of the *Phoenix* consortium determined the parameters for the blocks of the chain and the algorithms that they developed. Thus, we worked mainly on the radio link block, defining both the traditional and the adapted transmission for the MPEG-4 source co-decoder, the channel co-decoder, the multicarrier mo-demodulator, the App and the Phy JSCC Controllers. Moreover we adjusted the controlling algorithms to the video sequences transmitted in the different scenarios. The list of all the configuration parameters has been reported in [68].

The scenarios are:

- Scenario 1 is called ”video conference on the move” and corresponds to conversational mode, UMTS channel (or 4G), multicast, mobile phone, high cost is possible, confidentiality, multi-user scenario.

- Scenario 2 is called ”video conference from a caf” and corresponds to conversational mode, WLAN channel (or 4G), multicast, mobile phone, no mobility, high cost is possible, confidentiality, multi-user scenario.

- Scenario 3 is called ”video on demand” and corresponds to streaming mode, WLAN channel, unicast, mobile phone, confidentiality.

- Scenario 4 is called ”learning application” and corresponds to live encoded video data hence streaming mode with low delay, WLAN channel, laptop, no mobility, multicast, confidentiality.

- Scenario 5 is called ”video call on the move” and corresponds to UMTS channel, unicast, UMTS channel, mobile phone, confidentiality (optional).

	BPSK	**STTC(2x2)**	**WLAN OFDM**	**WLAN OFDM 4G**

Scenario 1	SNR=8 dB
	G=1.74 dB
		SNR=6 dB
		G=0.42 dB
			N/A	N/A

Scenario 2	SNR=9 dB
	G=2.51 dB
		SNR=7 dB
		G=4.61 dB
			SNR=34 dB
			G=2.01 dB
				SNR = 30 dB
				G=2.69 dB

Scenario 3	SNR=8 dB
	G=3.11 dB
		SNR=5 dB
		G=2.77 dB
			SNR=32 dB
			G=2.79 dB
				SNR=30 dB
				G=6.06 dB

Scenario 4	SNR=8 dB
	G=1.34 dB
		SNR=5 dB
		G=4.41 dB
			SNR=32 dB
			G=2.43 dB
				SNR=30 dB
				G=5.17 dB

Scenario 5	SNR=7 dB
	G=2.48 dB
		SNR=5 dB
		G=3.04 dB
			N/A	N/A

Scenario 6	SNR=8 dB
	G=3.26 dB
		SNR=5 dB
		G=4.23 dB
			SNR=28 dB
			G=2.34 dB
				SNR=26 dB
				G=4.54 dB

Scenario 7	SNR=9 dB
	G=3.33 dB
		N/A	N/A	N/A

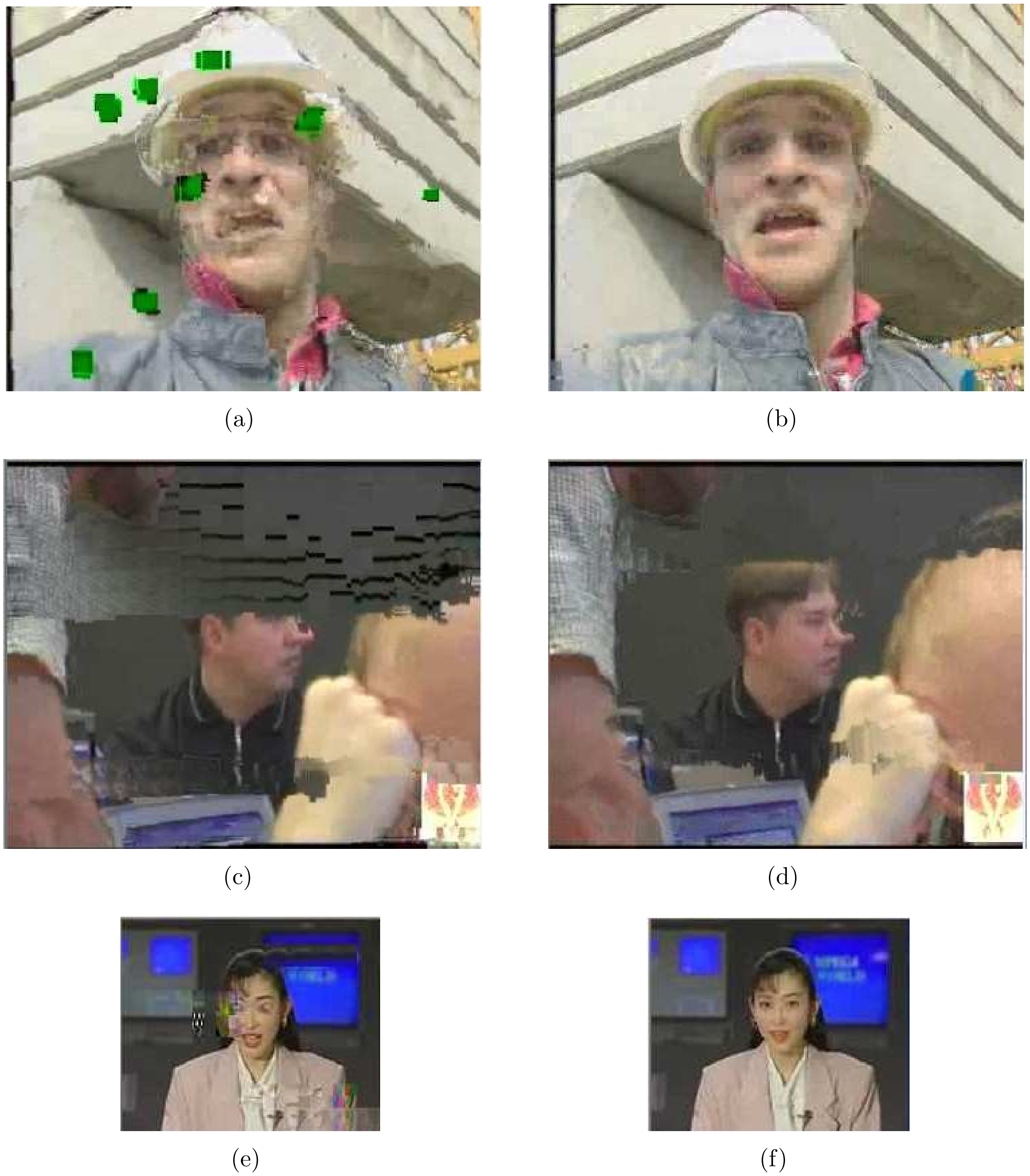
Table 5.3: Gains obtained with the different scenarios.

- Scenario 6 is called ”video call from a caf” and corresponds to unicast, WLAN, mobile phone, no mobility, lower cost issues, confidentiality (optional).
- Scenario 7 is called ”live news (pushed video information)” and corresponds to Multicast, UMTS channel (possibly MBMS), mobile phone, low delay, streaming mode.

Table 5.3 provides a summary of the gains achieved in case of MPEG-4 coding with the JSCC adaptation when compared to not adapted (or traditional) case, for different scenarios. In practice, the average improvement in terms of PSNR considering all the scenarios is above 3dB. In the following we report, as an example, some of the obtained curves and some examples of corresponding frames. Here each curve refers to a single simulation run, but tests have shown that, when average values are considered, smoother behaviors are observed.

5.4 Feedback information SSIM

In order to show that similar gains can be achieved with the proposed system also by considering different video quality metrics as feedback, we show in the following an example where the SSIM video quality metric is considered as feedback information for the App JSCC Controller. The considered video sequence is *Akiyo* in QCIF format and the transmission configuration parameters refer to Scenario 6 (video call from a caf), with STTC 2x2 and SNR=4 dB. The results are illustrated in Fig. 5.12 in terms of average



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.5: Examples of frames obtained in: Scenario 2 - classical (a) and adapted approach (b); Scenario 4 - classical (c) and adapted approach (d); Scenario 5 - classical (e) and adapted approach (f).

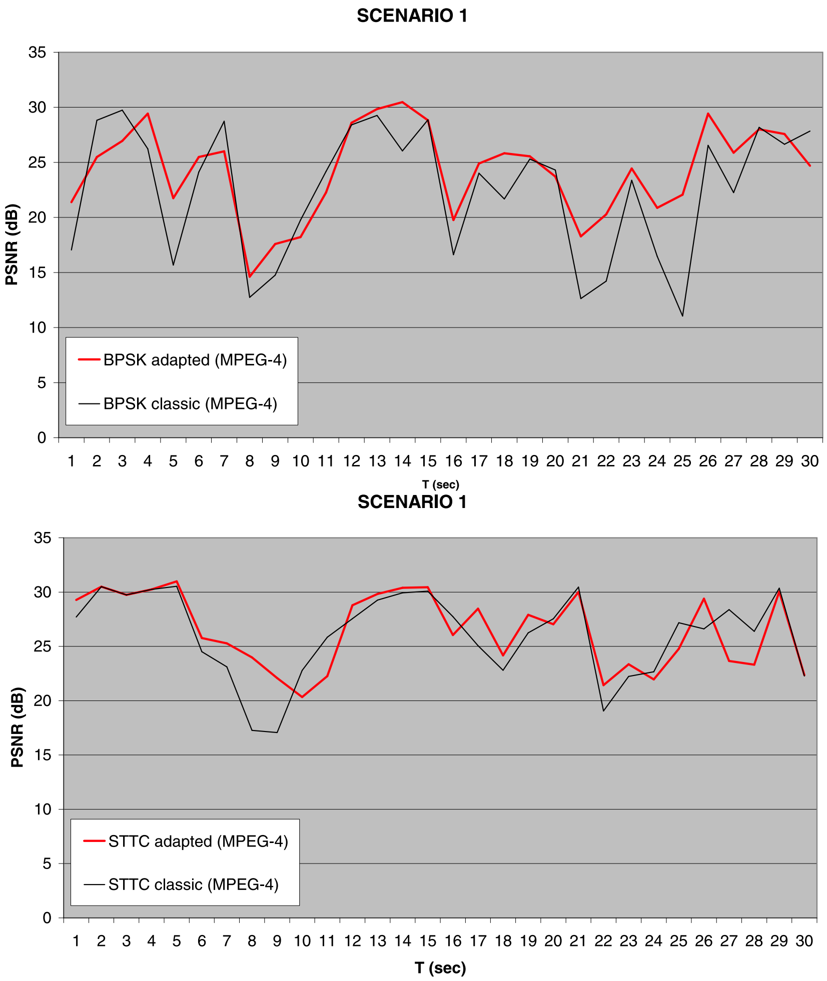


Figure 5.6: Performance comparison (scenario 1).

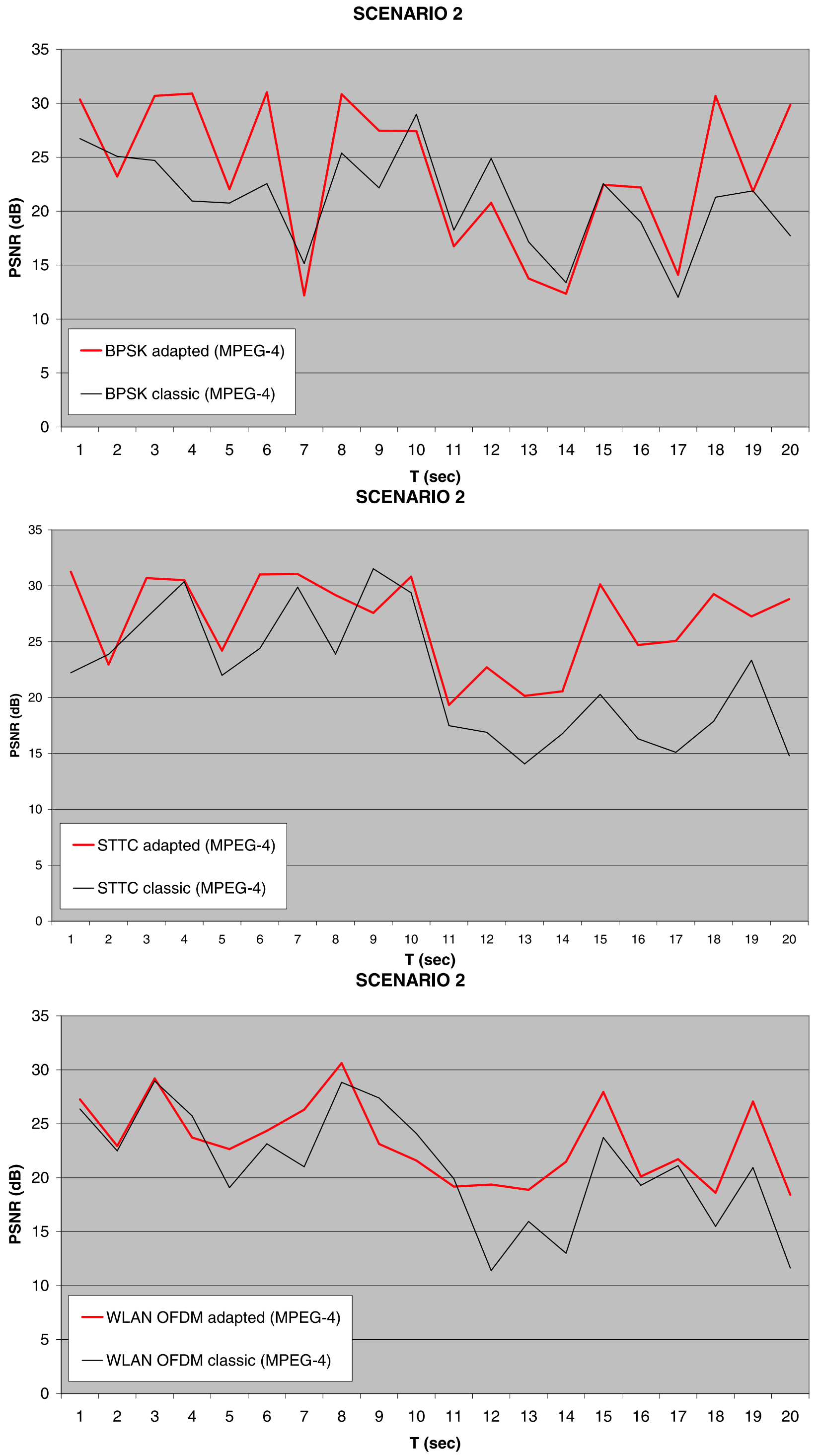


Figure 5.7: Performance comparison (scenario 2).

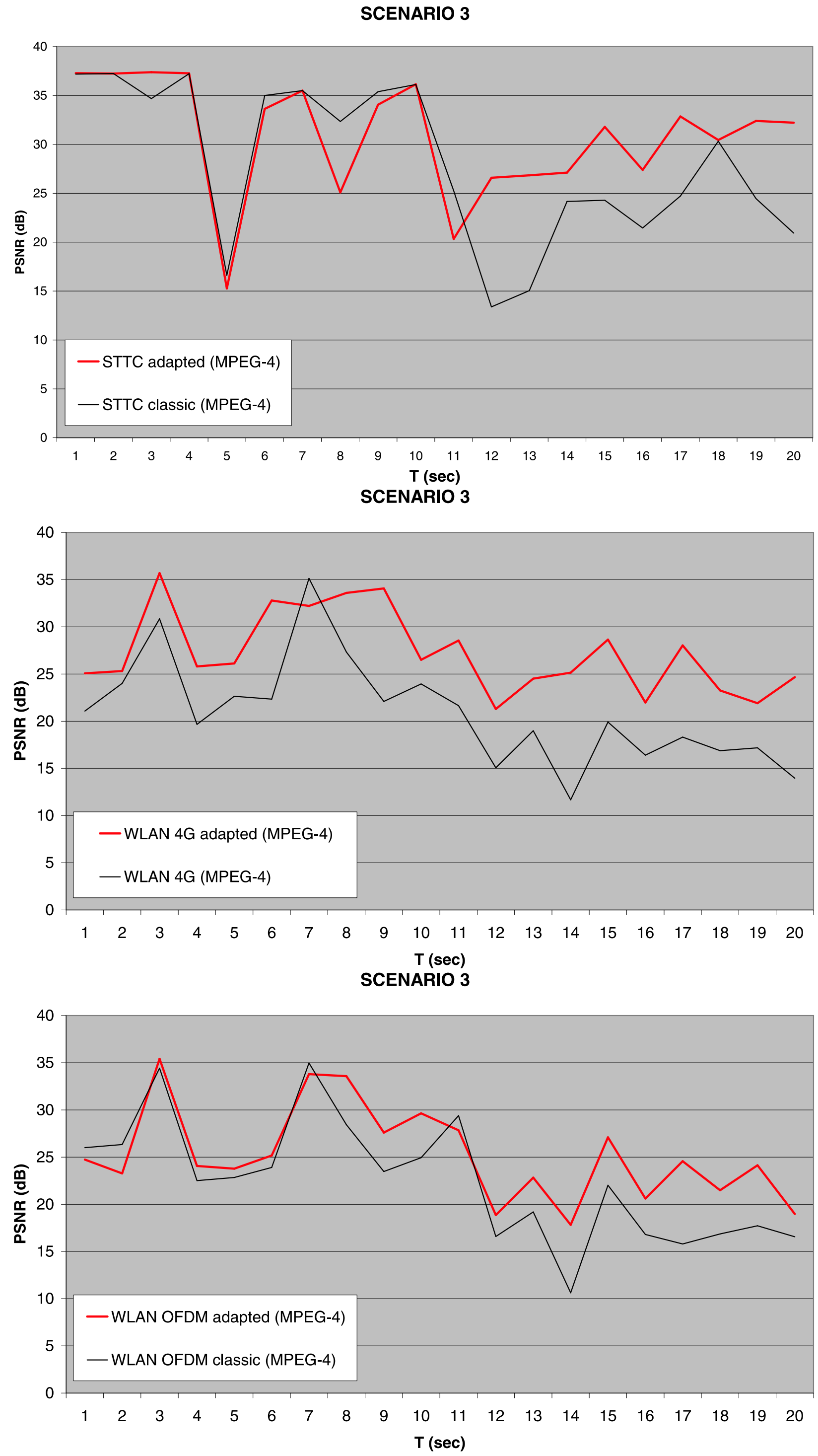


Figure 5.8: Performance comparison (scenario 3).

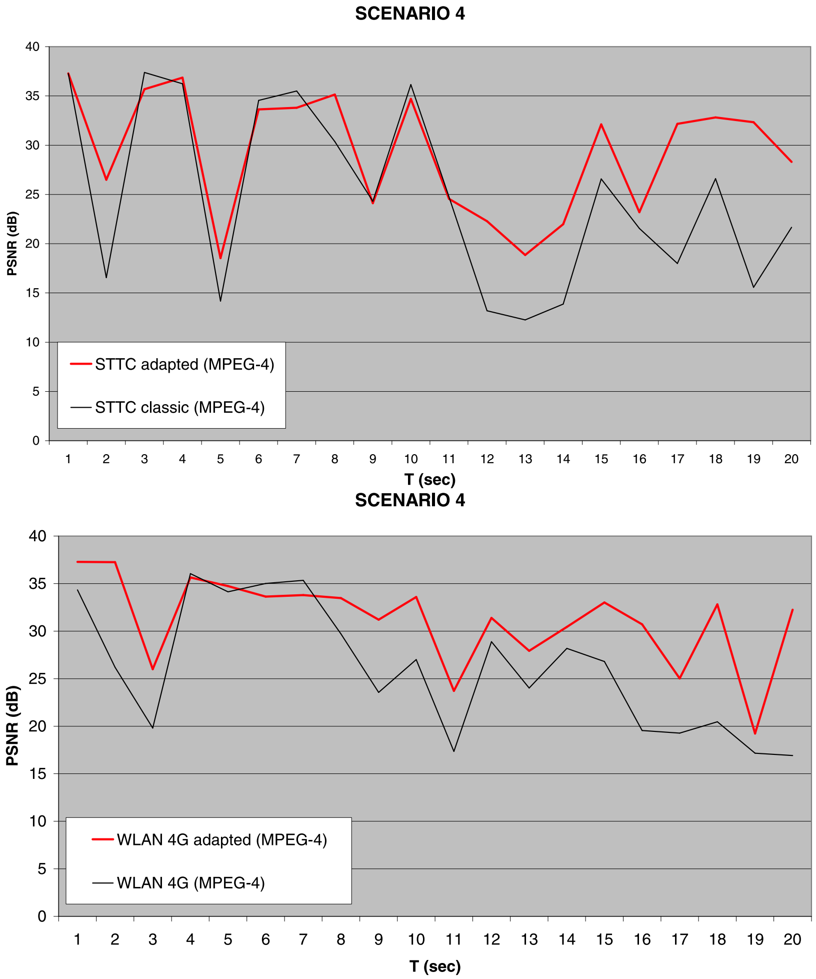


Figure 5.9: Performance comparison (scenario 4).

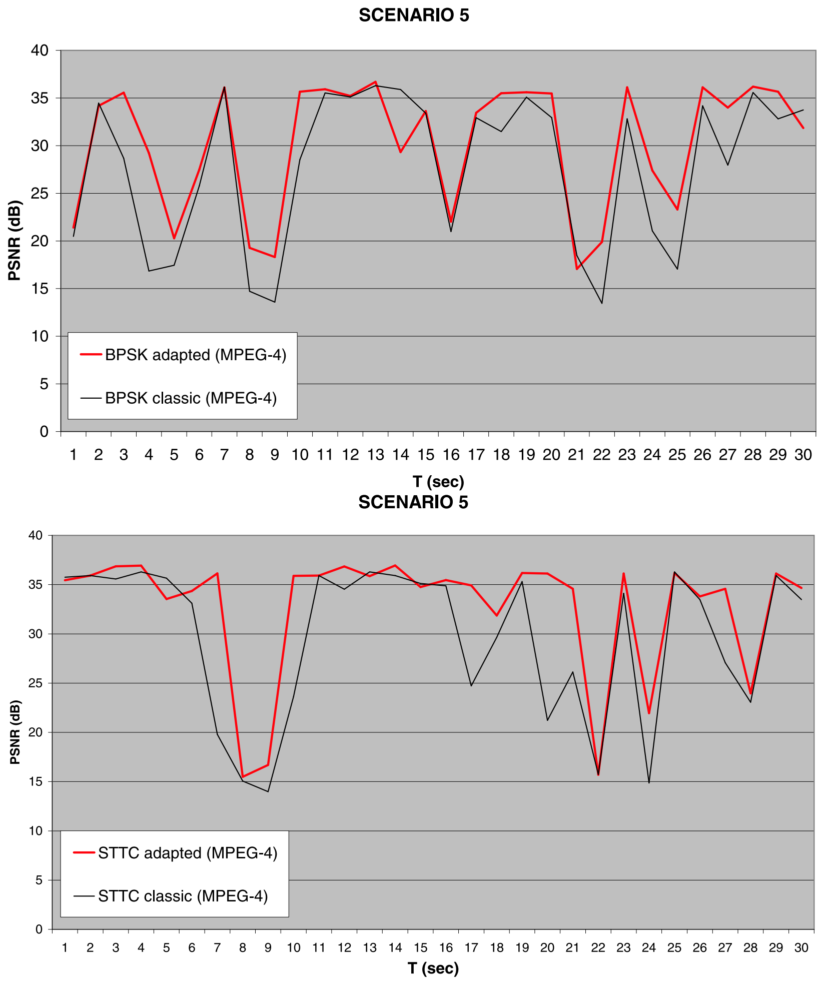


Figure 5.10: Performance comparison (scenario 5).

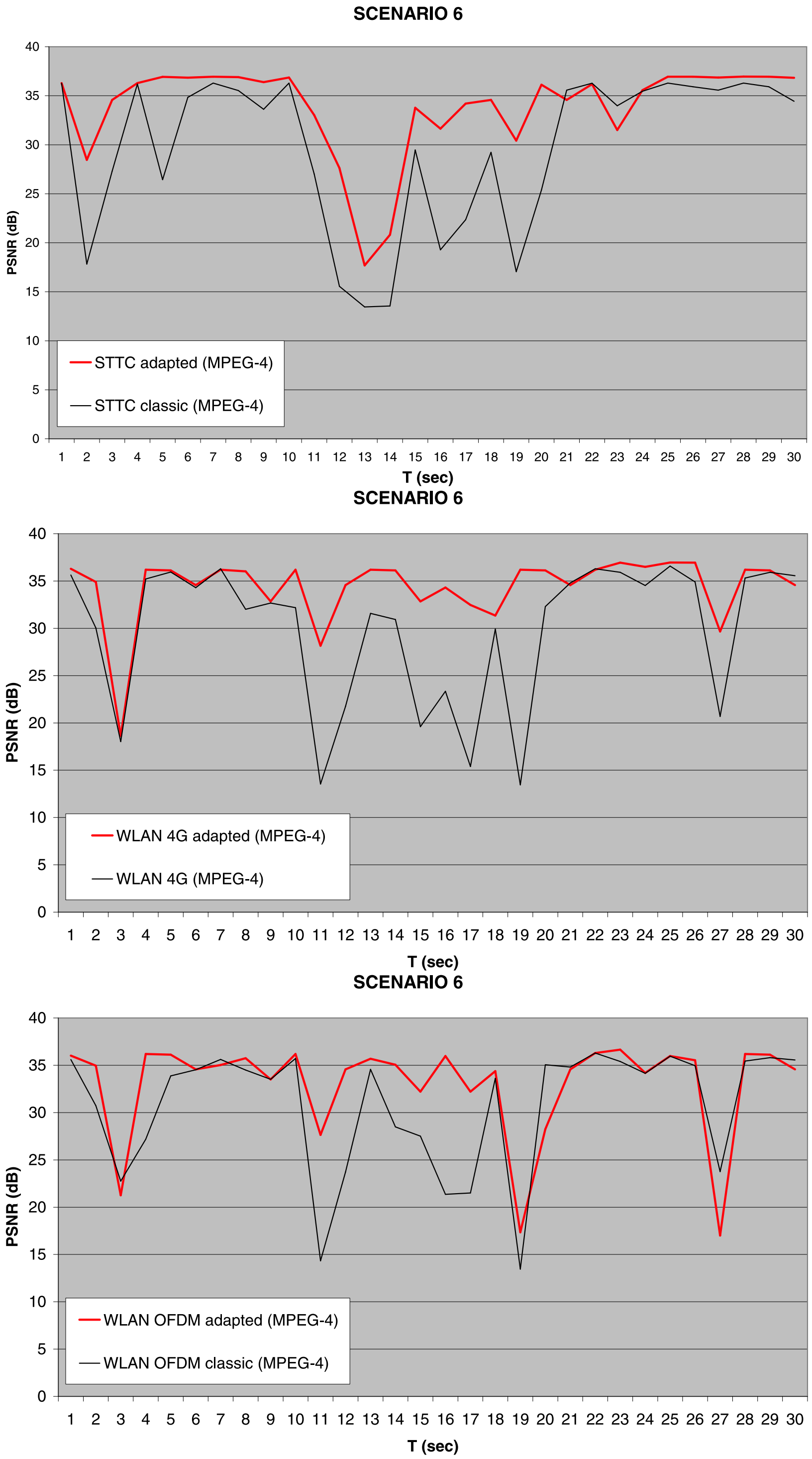
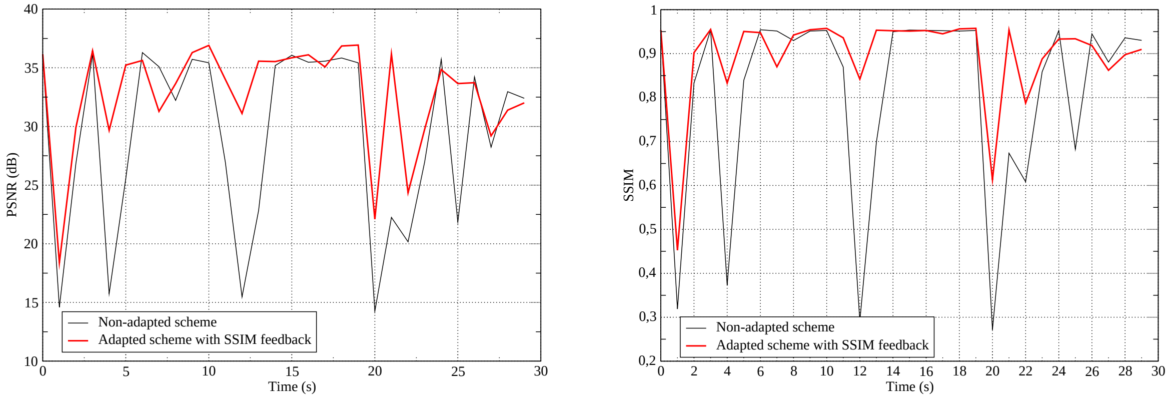
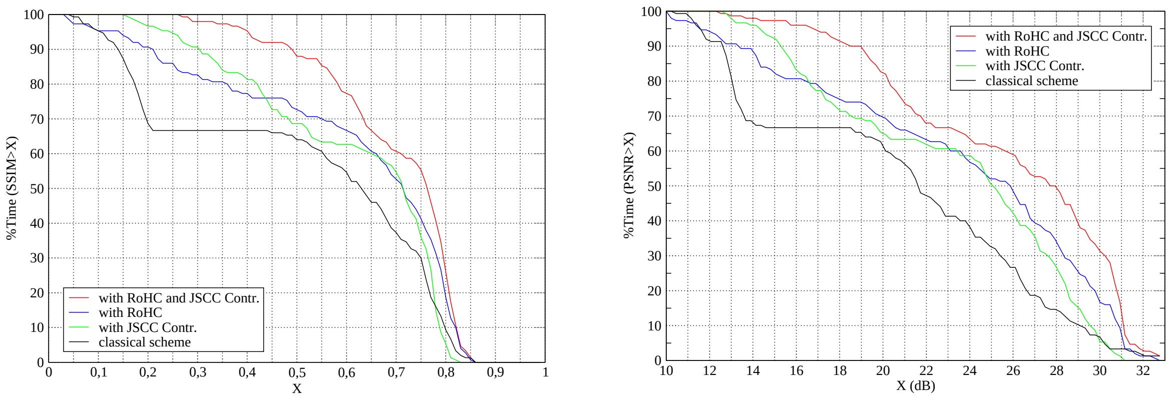


Figure 5.11: Performance comparison (scenario 6).



(a) (b)

Figure 5.12: Final quality in terms of PSNR [a] and SSIM [b] in case of SSIM index used



(a) (b)

Figure 5.13: Resulting CDFs for of PSNR [a] and SSIM [b] in case of SSIM index used as quality feedback.

quality metric versus time and in Fig. 5.13. The reported results have been obtained averaging over 3 simulation run.

5.5 Bio-medical application

Current and emerging developments in wireless communications integrated with developments in pervasive and wearable technologies will have a radical impact on future healthcare delivery systems. Nowadays, wireless telemedicine is a reality, requiring also the transmissione of medical video sequences over often unreliable links. M-Health can be defined as mobile computing, medical sensor, and communications technologies for healthcare [72]. Video streaming is the highest demanding application in this case. Clearly, video compression techniques for medical applications have to satisfy requirements of high fidelity, in order to avoid the loss of information that could help diagnosis and to prevent the transmission of fake artifacts that could deceive the medical operator. For keeping diagnostic accuracy, lossless compression techniques are often considered when medical video sequences are involved. Anyway, when transmission is over band limited, error prone channels, we have to face a trade-off between compression fidelity and protection and resilience to channel errors and packet loss. It has been observed [73] that when lossy compression is limited to ratios from 1:5 to 1:29, compression can be achieved with no loss in diagnostic accuracy.

For these reasons, lossy compression techniques have been considered for medical images and video compression.

In this paragraph we show the results obtained applying the described JSCC/D paradigm to biomedical video application. In particular we consider a teleultrasonography application. The situations where the video transmission may become a critical issue in the biomedical field are various: in emergency situations it may be possible for a medical operator to ask for a first approach diagnosis to a remote specialist, or even for the remote operator to control the medical equipment during the visit.

In the following example results, we consider a WLAN connection. We assume the WLAN supports, at the radio link level, a coded bit-rate of 12 Mbit/s. The ultrasonography video stream is coded according to the MPEG-4 standard and supposed to be multiplexed with other real-time transmissions, so that it occupies only an average portion of the available bandwidth corresponding to a coded bitrate of 650 kbit/s. CIF resolution has been selected. RoHC is applied, in order to compress the transport and network headers by transmitting only non-redundant information.

The App and Phy JSCC Controllers are considered, where the application layer controller performs source bit-rate adaptation and the physical layer one provides UEP, according to the average bit rate suggested by the App JSCC Controller, and drives adaptive bit-loading for multicarrier modulation.

Channel codes are IRA-LDPC codes with the “mother” code rate of (3500,10500), properly punctured and shortened in order to obtain different code rates. The resulting codewords are always 4200 bits long. The code rate is 2/3 for the non adapted system (EEP); in the adapted case the code rate can change according to SSI in order to perform UEP. The average coded bit-rate is the same in both the considered cases.

State	**(qI, qP)**	**Frame rate**	**Gop size**
1	(14, 16)	7.5	8
2	(11, 14)	7.5	8
3	(11, 12)	7.5	8
4	(11, 13)	15	15
5	(10, 12)	15	15

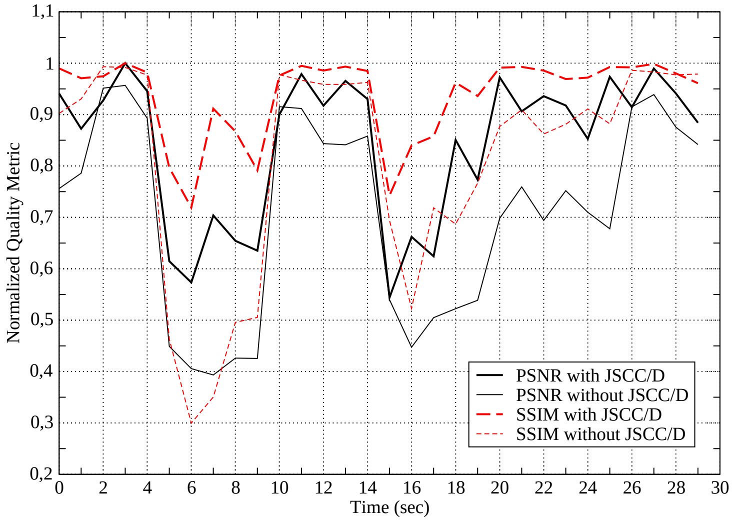


Figure 5.14: Normalized PSNR and SSIM versus time with the JSCC adapted and non-adapted systems.

In the first case, the modulation is a “classical” orthogonal frequency division multiplexing (OFDM) with 48 carriers for data transmission and a frame duration of 4 μs; margin adaptive bit-loading techniques managed by the Phy JSCC are considered in the adapted system.

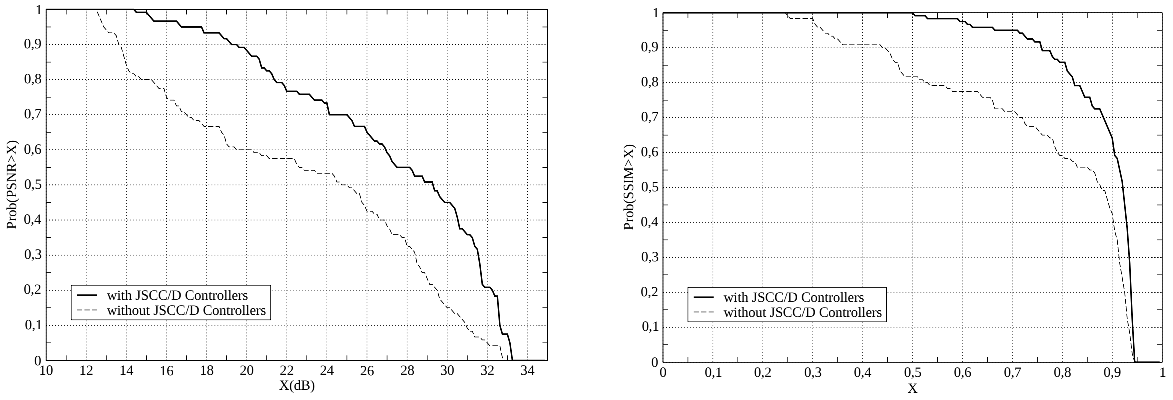
The channel is obtained according to the ETSI channel A model, representing the conditions of a typical office environment. It takes into account also a log-normal flat fading component with channel coherence time of 5s, to consider the fading effects due to large obstacles. A median signal-to-noise ratio of E_b/N_0=13.2 dB have been considered.

The states defined for the FSM modelling the App JSCC Controller have been reported in Table 5.4. The source bit-rate after MPEG-4 compression depends on the App JSCC Controller state and ranges from 207 kbit/s (state 1) to 384 kbit/s (state 5), taking into account also the overhead due to the various network headers.

In Fig. 5.15 we have reported the **CDF!** (CDF) obtained through simulation for the

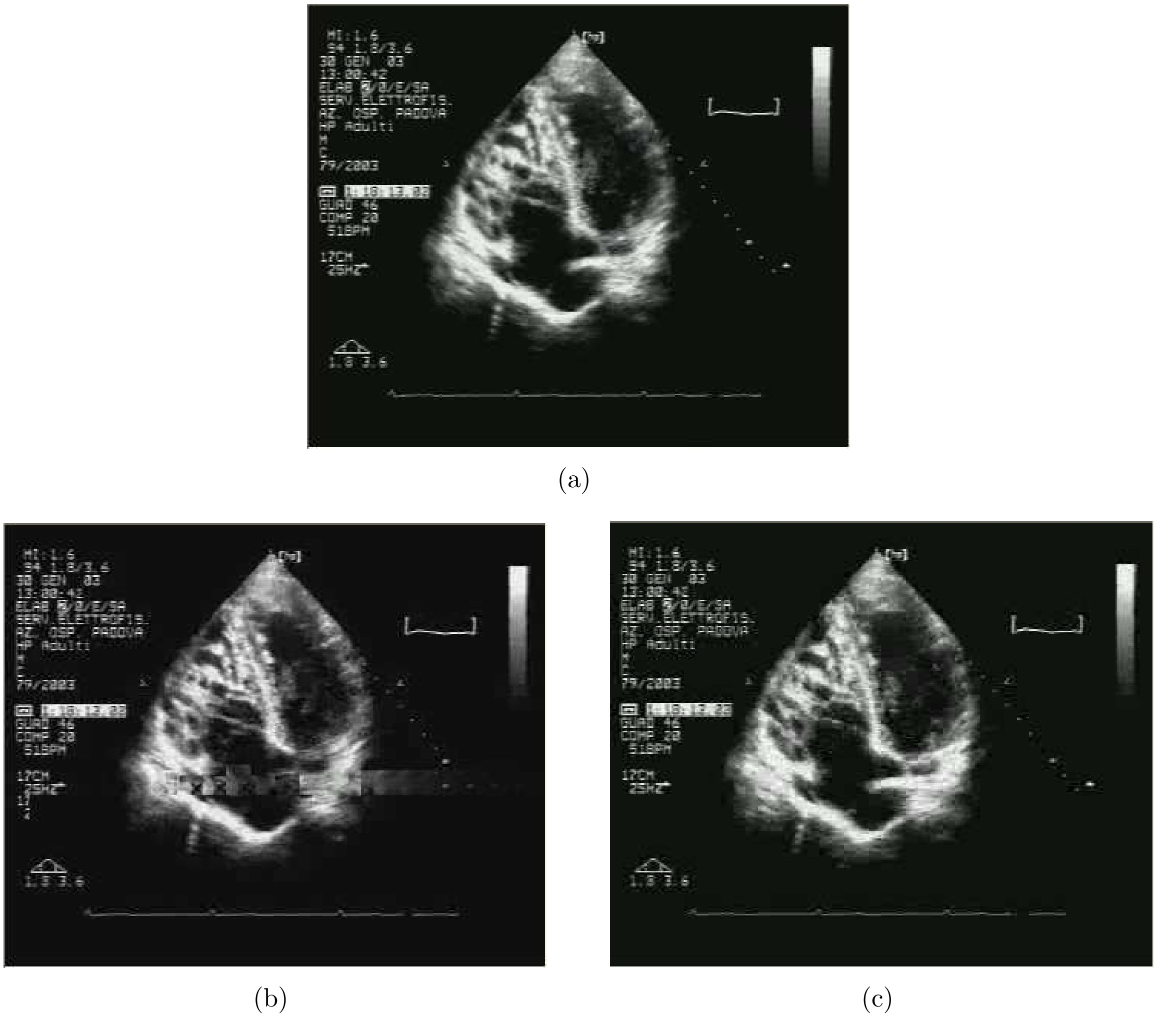
PSNR and the SSIM. The large gain due to the application of the joint controllers is evident. Figure 5.14 shows the results in terms of PSNR and SSIM [52] versus time. The quality curves reported in the graph have been obtained as the average of 4 distinct simulations, run with different noise seeds. Moreover, the quality values have been normalized with respect to the maximum value achieved in order to allow the comparison of different metrics in the same figure. An average gain of 4.4dB is provided by the adapted system, allowing the performance of the diagnosis with much higher accuracy than in the non-adapted case, as visual results confirm.

Figure 5.16 reports visual results for the echo-cardiography sequence considered. The original frame no. 123 is reported in Fig. 5.16a. The corresponding received video frame with the non-adapted system is reported in Fig. 5.16b; evident artifacts are visible, in terms of light stripes, affecting the correctness of the diagnosis. 5.16c reports the corresponding received video frame with the adapted system presenting a much higher visual quality, also reflected in diagnosis accuracy.



(a) (b)

Figure 5.15: Comparison between the obtained CDF for the PSNR and SSIM with and without the JSCC Controllers.



(a)

(b)

(c)

Figure 5.16: Visual results. [a] Frame no. 123 - original; [b] Frame no. 123 - MPEG-4 - without JSCC/D; [c] Frame no. 123 - MPEG-4 - with JSCC/D.

Chapter 6

Channel coding with SI at the encoder

Channel coding with side information on the channel state and, in particular, the DPC and the DTC problems are topics strictly connected to different fields of the information theory. Fist of all because the development of efficient coding technique capable of exploiting the side information (SI) often requires concepts borrowed from both the source and the channel coding theory, and, in the second place because of the wide spectrum of applications going from the coding for multiuser scenarios (e.g. the broadcast channel (BC)) to watermarking and information embedding techniques.
Thus, from our perspective, studying and working on the DPC and DTC problems was interesting for a twofold reason:

- in a BC scenario a DPC approach allows for an efficient exploitation of the radio resources, maximizing the throughput of the system and thus supporting the word of the JSCC Controllers;

- DPC and DTC are key strategies in the development of good watermarking technique; as observed at the end of Chapter 3 this may constitute a good solution to realize a no-reference distortion metric to be used as quality feedback for the App JSCC Controller.

In this Chapter we introduce the problem of coding with SI available at the transmitter and the DPC/DTC scenarios.

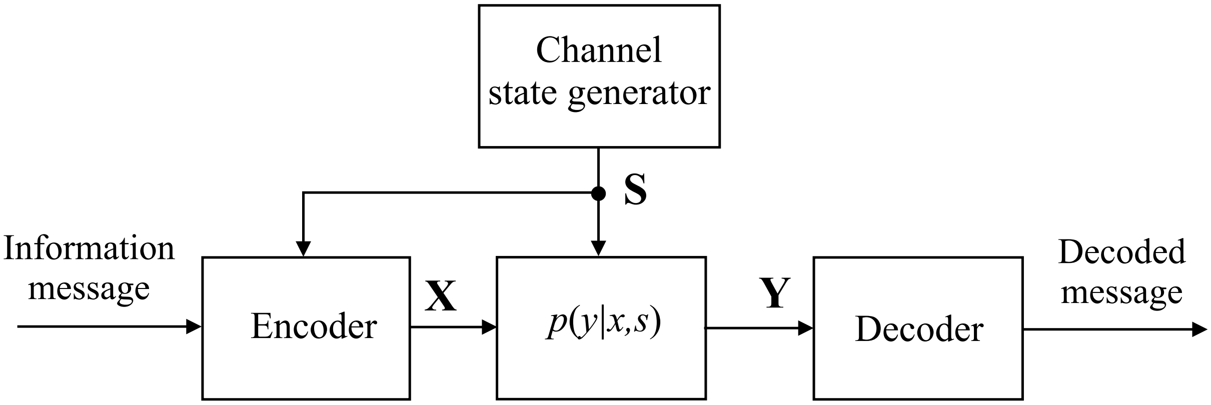


Figure 6.1: General scheme of channel coding with non-causal SI available at the encoder.

6.1 Channels with SI non-causally known by the encoder

The general problem of channel coding with non-causal SI at the encoder side has been depicted in Fig. 6.1. In the case of finite alphabet random variables (r.v.s), the capacity of memoryless channels with SI non causally known by the encoder is synthetized be the famous result by Gel'fand and Pinsker [12]

$$C_{SI-T}^{non-caus} = \max_{p(x,u|s)} I(U;Y) - I(U;S) \tag{6.1}$$

where U is a proper auxiliary r.v. and the maximum is taken over the distributions with the form $p(y,x,u|s) = p(y|x,s)p(x,u|s)$. The proof of this formula is based on well-known random coding and binning arguments and its validity can be readily extended to continuous-alphabet r.v.s by substituting the maximum operation with the supremum.[1]

Sketching the proof of the achievability of (6.1) is extremely useful to understand the general philosophy underlying the different DPC and DTC techniques. U is an auxiliary r.v. belonging to the finite alphabet $\mathcal{U}$, with distribution $p(u)$, whose significance will become clear in the following. $A_\epsilon^{(n)}(U)$ represents the set of typical sequences of length n with respect to $p(U)$ [2], defined as

$$A_\epsilon^{(n)}(U) = \left\{ \mathbf{u} \in \mathcal{U}^n : \left| \frac{1}{n} \log_2 \frac{1}{p(\mathbf{u})} - H(U) \right| < \epsilon \right\} \tag{6.2}$$

[1] Note that with distribution $p(x)$ of a RV X we will denote either a probability mass function (PMF) or a probability density function (PDF) accordingly to the nature of the alphabet of X. The proper meaning will result clear from the context.

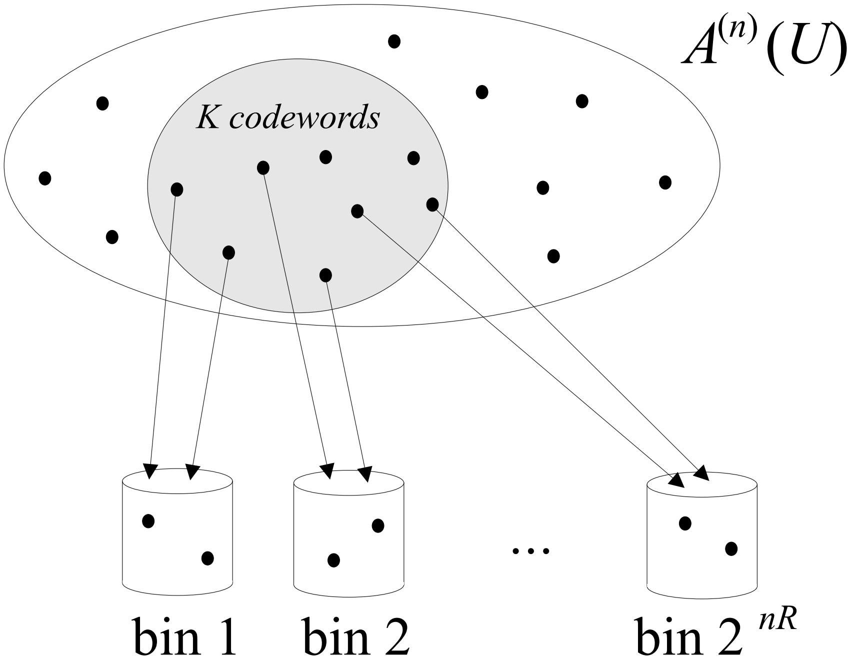


Figure 6.2: Random codebook generation for encoding with SI.

where $H(\cdot)$ denotes the entropy of the r.v. and ϵ is a small positive constant. Thus, if $\mathbf{u} \in A_\epsilon^{(n)}(U)$,

$$2^{-n(H(U)+\epsilon)} \leq p(\mathbf{u}) \leq 2^{-n(H(U)-\epsilon)}. \tag{6.3}$$

For n sufficiently large, the number of elements of $A_\epsilon^{(n)}(U)$ is bounded by:

$$2^{n(H(U)-2\epsilon)} \leq |A_\epsilon^{(n)}(U)| \leq 2^{n(H(U)+2\epsilon)}. \tag{6.4}$$

Thus, using the notation $a_n \doteq 2^{n(b\pm\epsilon)}$ to mean

$$\left|\frac{1}{n}\log_2 a_n - b\right| < \epsilon \quad \text{for } n \text{ sufficiently large} \tag{6.5}$$

we can also write $p(\mathbf{u}) \doteq 2^{-n(H(U)\pm\epsilon)}$ and $|A_\epsilon^{(n)}| \doteq 2^{n(H(U)\pm2\epsilon)}$.

The achievability of (6.1) is based on the classical technique of random coding: we generate a $(2^{nR}, n)$ code at random and we show that the probability of error averaged over all codes decreases to zero as the code length $n \to \infty$. Thus, because of the average, there must exist at least one code with such a vanishing error probability.

- *Random codebook generation.* We randomly distribute K typical sequences $\mathbf{u} \in A_\epsilon^{(n)}(U)$ among 2^{nR} different bins. Each bin is associated to a distinct information message $m \in \{1, ..., 2^{nR}\}$. Then the code is revealed to both the sender and the receiver. This random generation process has been illustrated in Fig. 6.2.

- *Encoding process.* By the symmetry of the random code construction, the conditional probability of error does not depend on which message is sent. Let's suppose that the message m_i has been selected for the transmission and let's denote with $\tilde{\mathbf{s}}$ the side information sequence of length n, which is non-causally known by the encoder. We look in the i^{th} bin for a sequence $\tilde{\mathbf{u}}$ jointly typical with $\tilde{\mathbf{s}}$, i.e.

$$(\tilde{\mathbf{u}}, \tilde{\mathbf{s}}) \in A_\epsilon^{(n)}(U, S) \tag{6.6}$$

where the set of jointly typical sequences $A_\epsilon^{(n)}(U, S)$ is defined as

$$\begin{aligned} A_\epsilon^{(n)}(U,S) = \Big\{ & (\mathbf{u}, \mathbf{s}) \in \mathcal{U}^n \times \mathcal{S}^n : \\ & \mathbf{u} \in A_\epsilon^{(n)}(U), \mathbf{s} \in A_\epsilon^{(n)}(S) \text{ and} \\ & \left| \frac{1}{n} \log_2 \frac{1}{p(\mathbf{u}, \mathbf{s})} - H(U,S) \right| < \epsilon \Big\}. \end{aligned} \tag{6.7}$$

Thanks to the asymptotic equipartition property (AEP), it is possible to show ([2]) that the probability that a couple of independently selected sequences $\mathbf{u}$ and $\mathbf{s}$ belongs to $A_\epsilon^{(n)}(U, S)$ is

$$\Pr\{(\mathbf{u}, \mathbf{s}) \in A_\epsilon^{(n)}(U,S)\} \leq 2^{-n(I(U;S)-3\epsilon)}. \tag{6.8}$$

Thus the probability of finding in the selected bin a sequence $\tilde{\mathbf{u}}$ jointly typical with $\tilde{\mathbf{s}}$ tends to 1 with $n \to \infty$, if at least $2^{n(I(U;S)+\delta)}$ typical sequences are present in the bin, where δ is a small positive constant. An encoding error is declared if no such a sequence is found, while, if no errors occur, the encoder proceeds selecting a sequence $\tilde{\mathbf{x}}$ which is jointly typical with the couple $(\tilde{\mathbf{u}}, \tilde{\mathbf{s}})$ and transmits it. In other words, the transmitted sequence $\tilde{\mathbf{x}}$ must satisfy

$$(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{\mathbf{s}}) \in A_\epsilon^{(n)}(X, U, S), \tag{6.9}$$

where the set of jointly typical sequences $A_\epsilon^{(n)}(X, U, S)$ is defined as usual:

$$\begin{aligned} A_\epsilon^{(n)}(X,U,S) = \Big\{ & (\mathbf{x}, \mathbf{u}, \mathbf{s}) \in \mathcal{X}^n \times \mathcal{U}^n \times \mathcal{S}^n : \\ & \mathbf{x} \in A_\epsilon^{(n)}(X),\ \mathbf{u} \in A_\epsilon^{(n)}(U), \mathbf{s} \in A_\epsilon^{(n)}(S), \\ & (\mathbf{x}, \mathbf{u}) \in A_\epsilon^{(n)}(X,U),\ (\mathbf{x}, \mathbf{s}) \in A_\epsilon^{(n)}(X,S), \\ & (\mathbf{s}, \mathbf{u}) \in A_\epsilon^{(n)}(U,S) \quad \text{and} \\ & \left| \frac{1}{n} \log_2 \frac{1}{p(\mathbf{x}, \mathbf{u}, \mathbf{s})} - H(X,U,S) \right| < \epsilon \Big\}. \end{aligned} \tag{6.10}$$

An assumption usually made w.l.o.g. is that the transmitted sequence $\tilde{\mathbf{x}}$ is a deterministic function of $(\tilde{\mathbf{u}}, \tilde{\mathbf{s}})$, i.e. $\tilde{\mathbf{x}} = f(\tilde{\mathbf{u}}, \tilde{\mathbf{s}})$. Because of the AEP ([2]), the probability of an error event at the transmitter tends to 0 with $n \to \infty$ when the number of sequences per bin has been properly chosen.

- *Decoding process.* The received sequence $\tilde{\mathbf{y}}$ is taken according to the conditional p.d.f. $p(\mathbf{y}|\mathbf{x} = \tilde{\mathbf{x}}, \mathbf{s} = \tilde{\mathbf{s}})$. After receiving the sequence $\tilde{\mathbf{y}}$, the decoder looks for the bin containing a sequence $\hat{\mathbf{u}}$ jointly typical with it and decides for the transmitted message $\hat{m}$ associated with that bin. This is possible because, as previously stated, the distribution of the codewords among the bins is a-priori known also by the decoder. A decoding error occur if no such a sequence $\hat{\mathbf{u}}$ is found in the i^{th} bin or if a sequence jointly typical with $\tilde{\mathbf{y}}$ is found in bin $k \neq i$.

Let's now analyze the probability of error. Denoting with $\mathbf{u}_j(k)$ the j^{th} sequence in the k^{th} bin, we define the events:

$$E_i = \{(\mathbf{u}_j(i), \tilde{\mathbf{s}}) \notin A_\epsilon^{(n)}(U,S) \ \ \forall j\} \tag{6.11}$$

$$E_{kj} = \{(\mathbf{u}_j(k), \tilde{\mathbf{y}}) \in A_\epsilon^{(n)}(U,Y)\}. \tag{6.12}$$

The overall error event can now be represented as a union of (not disjoint) events:

$$E(i) = E_i \cup \left[\bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{ij} \right]^C \cup \bigcup_{k=1, k\neq i}^{2^{nR}} \ \bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{kj} \tag{6.13}$$

where with the notation $[\cdot]^C$ we have indicated the complementary event. The first term at the right hand side (RHS) of (6.13) represents the encoding error event, the second term corresponds to the situation in which there are no sequences jointly typical with $\tilde{\mathbf{y}}$ in the i^{th} bin, while the third term is the event that a sequence $\hat{\mathbf{u}}$ jointly typical with $\tilde{\mathbf{y}}$ is found in a wrong bin. The average probability of error for the considered random code

$$P_e = \frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} \Pr\{E(i)\} \tag{6.14}$$

can be bounded by the sum of the probabilities of the disjoint events

$$\Pr\{E(i)\} \leq \Pr\{E_i\} + \Pr\left\{ \left[\bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{ij} \right]^C \right\} + \Pr\left\{ \bigcup_{k=1, k\neq i}^{2^{nR}} \ \bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{kj} \right\}. \tag{6.15}$$

From the AEP [12][2], for n sufficiently large, the first and the second term at the RHS of (6.15) can be upper-bounded by

$$\Pr\{E_i\} \leq \epsilon \tag{6.16}$$

$$\Pr\left\{\left[\bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{ij}\right]^C\right\} \leq \epsilon, \tag{6.17}$$

while for the third term the following inequality holds:

$$\Pr\left\{\bigcup_{k=1,k\neq i}^{2^{nR}} \bigcup_{j=1}^{2^{n(I(U;S)+\delta)}} E_{kj}\right\} \leq 2^{nR}\, 2^{n(I(U;S)+\delta)}\, 2^{-n(I(U;Y)-3\epsilon)}. \tag{6.18}$$

Thus, combining (6.15) with (6.16), (6.17) and (6.18) we obtain

$$\Pr\{E(i)\} \leq \epsilon + \epsilon + 2^{nR}\, 2^{n(I(U;S)+\delta)}\, 2^{-n(I(U;Y)-3\epsilon)}, \tag{6.19}$$

from which it results that, if $R < I(U;Y) - I(U;S) - \delta - 3\epsilon$ the probability of error $\Pr\{E(i)\}$ (and so also P_e) can be made arbitrarily small for increasing n. Since ϵ and δ can be made arbitrarily small, the average probability of error, averaged over all choices of codebooks in the random code generation, is arbitrarily small. This shows the existence of at least one code capable of achieving rate $R < I(U;Y) - I(U;S)$ with a vanishing probability of error.

Now we can also quantify the number K of typical sequences $\mathbf{u}$, generated according the p.d.f. $p(U)$, and randomly distributed among the bins during the codebook construction. In fact, we have

$$K = 2^{nR}\, 2^{n(I(U;S)+\delta)} \leq 2^{n(I(U;Y)-3\epsilon)}. \tag{6.20}$$

6.2 Writing on dirty paper

Now we consider a particular, although important, case of channel coding with SI on the channel state known by the encoder. The SI S represents additive interference met by the transmitted signal in the memoryless channel

$$Y = X + S + Z, \tag{6.21}$$

where, as usual, X and Y are the transmitted and received symbols respectively, and Z is the additive noise. Let's consider the case of real symbols, with $S \sim \mathcal{N}(0, P_S)$ and

$Z \sim \mathcal{N}(0, P_N)$. In Fig. 6.3 we have graphically reported this transmission scenario. In [13], M. Costa demonstrated that the capacity of this channel, when the transmitted power is subject to the constraint

$$\mathbb{E}\left[X^2\right] \leq P_X \tag{6.22}$$

and when the whole interference sequence is known in advance by the encoder, is the same as without any interferers. Clearly, if the SI were known (also) by the decoder, the capacity of the channel would be

$$C = \frac{1}{2}\log_2\left(1 + \frac{P_X}{P_N}\right) \quad \left[\frac{\text{bits}}{\text{symb}}\right], \tag{6.23}$$

because the decoder can perfectly cancel the interference before decoding the transmitted codeword. Less obvious is the capacity of the dual situation of SI available at the encoder.

Let's suppose, as usual, that both the channel encoder and the decoder know the statistical description of the channel, i.e. the conditioned p.d.f. $p(y|x,s)$. In this situation, several strategies are actually possible ([13]):

- neglect the SI and apply a classical channel encoding technique. The maximum rate achievable with this solution is obviously

$$R_1 = \frac{1}{2}\log_2\left(1 + \frac{P_X}{P_S + P_N}\right) \quad \left[\frac{\text{bits}}{\text{symb}}\right]. \tag{6.24}$$

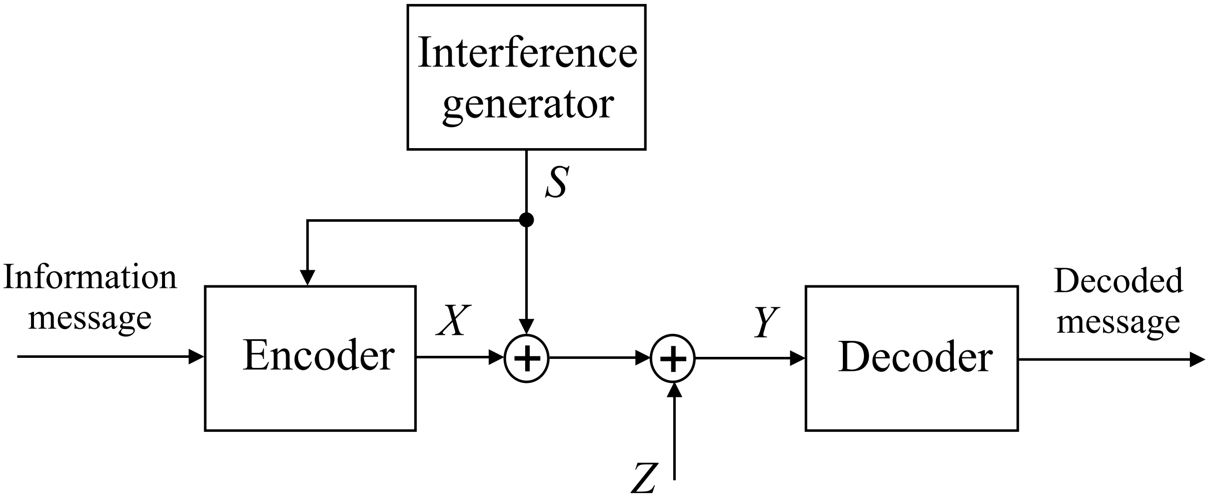


Figure 6.3: General scheme of DPC coding: the SI refer to additive interference present in the transmission channel.

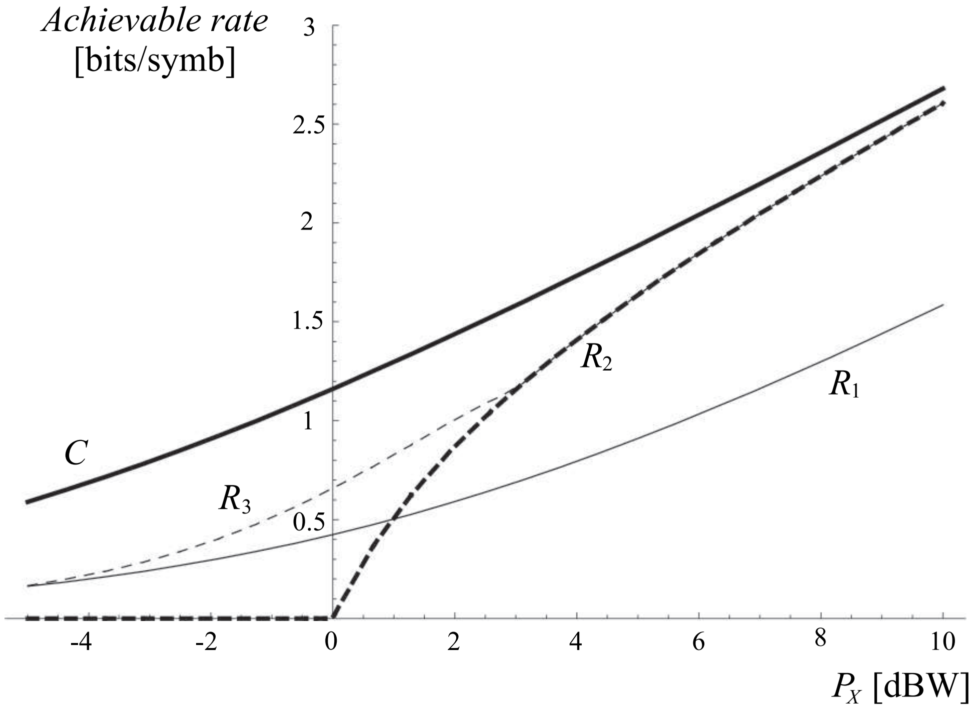


Figure 6.4: Achievable rates with the different strategies. The curves correspond to the case $P_S = 0$dBW, $P_N = -6$dBW and $\eta = 0.25$.

- pre-cancel the interference, if $P_X > P_S$. This leads to the maximum achievable rate

$$R_2 = \frac{1}{2}\log_2\left(1 + \frac{P_X - P_S}{P_N}\right) \quad \left[\frac{\text{bits}}{\text{symb}}\right]. \tag{6.25}$$

- pre-cancel a fraction η of the interference S, where $0 \leq \eta \leq \min\{1, P_S/P_X\}$. In this case, the maximum achievable rate is

$$R_3 = \frac{1}{2}\log_2\left(1 + \frac{(1-\eta)P_X}{P_N + (\sqrt{P_S} + \sqrt{\eta P_X})^2}\right) \quad \left[\frac{\text{bits}}{\text{symb}}\right]. \tag{6.26}$$

- actively exploit the knowledge of the SI by selecting codewords in the direction of the interference sequence. This solution constitutes the DPC approach and, as shown in the following, allows to reach the same capacity of the channel without any interference, $R = C$.

In Fig. 6.4 it is reported, as an example, a comparison of the rates achievable with the strategies mentioned above. In particular, the curves refer to the case $P_S = 0$ dBW, $P_N = -6$ dBW and $\eta = 0.25$.

Let's prove that the DPC constitutes the optimal approach, allowing to reach the rate C. First of all we observe that it must be $R \leq C$, because the interference in the channel cannot increase the amount of reliable information exchanged between the transmitter and the receiver. We start considering the extension of the Gel'fand and Pinsker's result to the case of non-finite alphabet r.v.s:

$$C_{SI-T}^{non-caus} = \sup_{p(x,u|s)} I(U;Y) - I(U;S), \tag{6.27}$$

where the supremum is taken over all the p.d.f. of the form $p(y,x,u|s) = p(y|x,s)p(x,u|s)$. The main problem in order to determine the channel capacity consists in finding the optimal distribution for the auxiliary r.v. U. In fact, we notice that the proof sketched in section 6.1 does not give any insight into the distribution which the optimal solution should have.

Now we show that the choice $U = X + \alpha S$, with $X \sim \mathcal{N}(0, P_X)$ and S statistically independent, constitutes the optimal strategy, leading to the equality in $R \leq C$. The parameter α, sometimes indicated as *shrinkage factor*, is real valued and selected within the range $[0,1]$: with $\alpha = 0$ the encoder completely neglect the anticipative knowledge of the interference, while $\alpha = 1$ corresponds to the maximum exploitation of the SI. First we find the expression of the achievable rate $I(U;Y) - I(U;S)$ correspondent to the specified U, which is dependant on the α, then we calculate the value $\alpha = \alpha_{opt}$ which maximize the mutual information difference.

Let's compute separately the terms in (6.27). Denoting with $H(\cdot)$ the differential entropy, we have

$$\begin{aligned} I(U;Y) &= H(U) - H(Y|U) \\ &= H(X+S+Z) - H(X+S+Z|X+\alpha S) \\ &= H(X+S+Z) + H(X+\alpha S) - H(X+S+Z;X+\alpha S), \end{aligned} \tag{6.28}$$

where $H(\cdot|\cdot)$ and $H(\cdot,\cdot)$ are the conditioned and the joint differential entropy respectively.

Given the statistical independence among the X, S and Z, we obtain

$$H(X+S+Z) = \frac{1}{2}\log_2\left(2\pi e(P_X+P_S+P_N)\right) \tag{6.29}$$

$$H(X+\alpha S) = \frac{1}{2}\log_2\left(2\pi e(P_X+\alpha^2 P_S)\right) \tag{6.30}$$

$$\begin{aligned} H(X+S+Z;X+\alpha S) &= \frac{1}{2}\log_2\left((2\pi e)^2 \begin{vmatrix} P_X+P_S+P_N & P_X+\alpha P_S \\ P_X+\alpha P_S & P_X+\alpha^2 P_S \end{vmatrix}\right) \\ &= \frac{1}{2}\log_2\left((2\pi e)^2\left((P_X+P_S+P_N)(P_X+\alpha^2 P_S)-(P_X+\alpha P_S)^2\right)\right). \end{aligned} \tag{6.31}$$

Hence, substituting (6.29), (6.30) and (6.31) into (6.28) we have:

$$I(U;Y) = \frac{1}{2}\log_2\frac{(P_X+P_S+P_N)(P_X+\alpha^2 P_S)}{(1-\alpha)^2 P_X P_S+(P_X+\alpha^2 P_S)P_N}. \tag{6.32}$$

Similarly, for $I(U;S)$ we obtain:

$$I(U;S) = H(U)+H(S)-H(U,S) \tag{6.33}$$

where

$$\begin{aligned} H(S) &= \frac{1}{2}\log_2 2\pi e P_S \\ H(U,S) &= \frac{1}{2}\log_2\left((2\pi e)^2 \begin{vmatrix} P_X+\alpha^2 P_S & \alpha P_S \\ \alpha P_S & P_S \end{vmatrix}\right) \\ &= \frac{1}{2}\log_2\left((2\pi e)^2 P_X P_S\right). \end{aligned} \tag{6.34}$$

Thus, after a few simplifications, $I(U;S)$ reduces to

$$I(U;S) = \frac{1}{2}\log_2\frac{P_X+\alpha^2 P_S}{P_X}. \tag{6.35}$$

Combining the two expressions (6.32) and (6.35), we have

$$R(\alpha) = I(U;Y)-I(U;S) = \frac{1}{2}\log_2\frac{P_X(P_X+P_S+P_N)}{(1-\alpha)^2 P_X P_S+(P_X+\alpha^2 P_S)P_N}, \tag{6.36}$$

which can be maximized taking the derivative with respect to α, leading to

$$\frac{d}{d\alpha}R(\alpha) = 0 \quad \Leftrightarrow \quad \alpha = \alpha_{opt} = \frac{P_X}{P_X+P_N}. \tag{6.37}$$

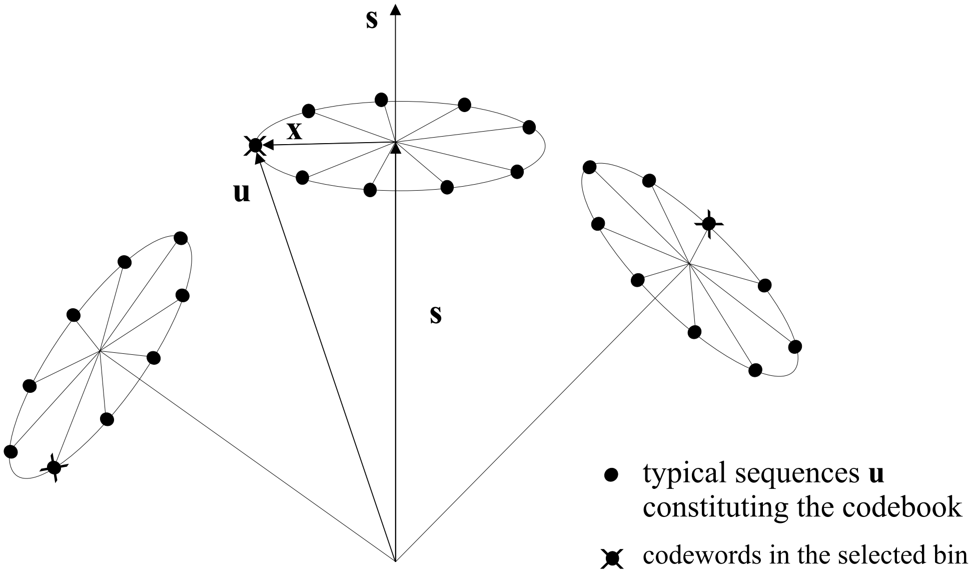

s
x
u
s
● typical sequences **u** constituting the codebook
codewords in the selected bin

Figure 6.5: General scheme of channel coding with causal SI available at the encoder.

The maximum is reached in correspondence of

$$\alpha_{opt} = \frac{P_X}{P_X + P_N} = \frac{\text{SNR}}{1 + \text{SNR}} \tag{6.38}$$

where $\text{SNR} = P_X/P_N$ is the signal to noise power ratio. Thus we have

$$\max_{\alpha} R(\alpha) = R(\alpha_{opt}) = \frac{1}{2}\log_2\left(1 + \frac{P_X}{P_N}\right) = C \tag{6.39}$$

and this proves that the initial choice of $U = X + \alpha S$ is optimal. We notice that the optimal choice for α does not depend on the interference statistics, but only on the SNR characterizing the transmission: when the channel is good (high SNR), $\alpha \to 1$ and the optimal encoder tends to take into large consideration its SI knowledge, while, when the channel is very noisy (low SNR), $\alpha \to 0$ and the encoder tends to neglect the SI. In fact, as we will show in Chapter 7, the main advantage due to the introduction of the shrinkage factor α is the possibility to minimize the overall noise seen by the decoder. In other words, with the optimal choice for α the decoder is capable to compute the minimum mean square error (MMSE) estimation of U needed to achieve the capacity (see [20] for the role of MMSE estimator in lattice-based capacity achieving schemes).

Fig. 6.5 illustrates the encoding process. Codewords and sequences of symbols of length n have been represented with points in $\mathbb{R}^n$ (in the figure $n = 3$). Basically, the encoder looks for the codeword **u** nearest to the shrunk version of the interference sequence, $\alpha\mathbf{s}$, within the bin selected by the information message. The transmitted sequence $\mathbf{x} = \mathbf{u} - \alpha\mathbf{s}$ does not belong to a finite set of possibilities, but it is a sequence in $\mathbb{R}^n$ itself, whose elements vary with continuity. The encoder exploits the interference as much as possible, adding the minimum power sequence capable of driving it to an allowed codeword $\mathbf{u}$: in a certain sense, it "adapts" itself to the interference. Thus, the decoder task is not to decide or, better, to estimate the transmitted sequence $\mathbf{x}$, but to recognize which codeword $\mathbf{u}$ has been selected for the transmission. Similarly, no estimation of the interference $\mathbf{s}$ has to be done to correctly decode the message. If the rate of the code is below the channel capacity, with a probability tending to 1 there is only one sequence $\hat{\mathbf{u}}$ jointly typical with the received sequence $\mathbf{y}$ and it corresponds to the codeword selected by the transmitter. So the decoder decides for the message associated to the bin containing $\hat{\mathbf{u}}$.

6.3 Dirty paper coding for binary channels

The remarkable result reported in the previous section was then extended to the case of arbitrarily varying interference and, in case of asymptotically high signal-to-noise ratios (SNR), to non-Gaussian noise. Anyway, some caution is needed when extending Costa's observations to non-Gaussian channel. In fact, the claim that the two dual scenarios of interference perfectly-known at the decoder and at the encoder are completely equivalent in terms of capacity is true in the Gaussian case but, unfortunately, it is not valid for many other important channels. In the binary case X, S, Z and Y belong to a binary alphabet and the transmission is subject to the constraint

$$E\{d_H(X, 0)\} \leq d, \tag{6.40}$$

where $d \in [0, 1/2]$ and $d_H(\cdot, \cdot)$ denotes the Hamming distance. Let's consider the interference $S \sim \mathrm{Ber}(1/2)$ and the noise $Z \sim \mathrm{Ber}(p)$, statistically independent, and a memoryless channel. The interference reduces to zero the mutual information between the input and the output of the channel if neither the encoder nor the decoder can exploit some SI. On the contrary, its presence is completely eliminated if the interference is known by the decoder, so that the capacity becomes:

$$C_{bin}(d) = h(d * p) - h(p) \tag{6.41}$$

where we have defined $p_1 * p_2 = p_1(1 - p_2) + (1 - p_1)p_2$ and $h(p_1) = -p_1 \log_2 p_1 - (1 - p_1) \log_2(1 - p_1)$.

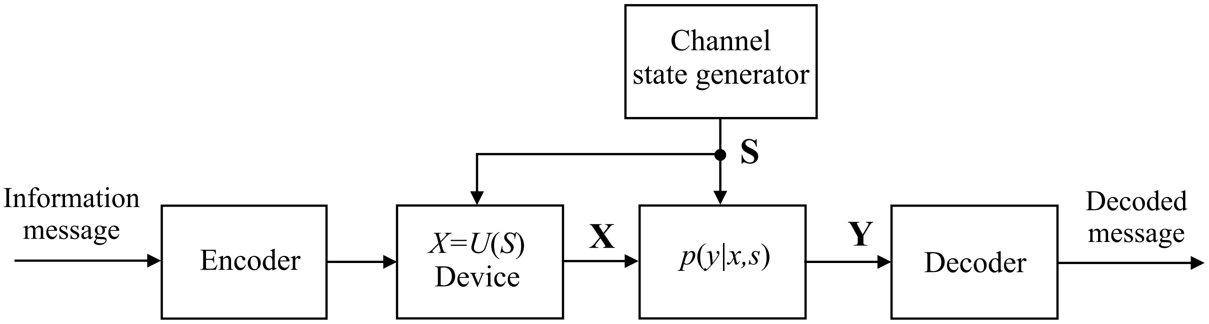


Figure 6.6: General scheme of channel coding with causal SI available at the encoder.

Barron et al. showed in [14] that in the binary dirty paper case the capacity becomes

$$C_{bin}^{dpc}(d) = u.c.e.\{g(d)\} \tag{6.42}$$

where $u.c.e.$ stands for "upper concave envelope" and $g(d)$ is defined as

$$g(d) = \begin{cases} g_1(d) = 0 & \text{if } 0 \leq d < p \\ g_2(d) = h(d) - h(p) & \text{if } p \leq d \leq 1/2. \end{cases} \tag{6.43}$$

6.4 Channels with SI causally known by the encoder

Historically, the first approach to the problem of channel coding with SI at the encoder considered a non-anticipative knowledge of the channel state. This situation has been graphically reported in Fig. 6.6 and it is particularly relevant for a twofold reason: first of all, in many communication scenarios presuming to know the future sequence of channel states is not realistic and so a DPC approach is not feasible. In the second place, even if the encoder could perfectly know the entire SI sequence, the complexity of the coding scheme capable of properly exploiting such a knowledge may become extremely high or even prohibitive for an actual implementation. Thus a simplified approach which takes into account only the past sequence of SI and the present channel state may become preferable.

The fundamental expression for the capacity with causal SI at the encoder was found by Shannon in [15], in the case of discrete memoryless channel with memoryless channel states. It can be written as

$$C_{SI-T}^{caus} = \max_{p(u)} I(U;Y) \tag{6.44}$$

where U is an auxiliary RV statistically independent of the channel state. Moreover Shannon found that the optimal input of the channel depends only on the current channel

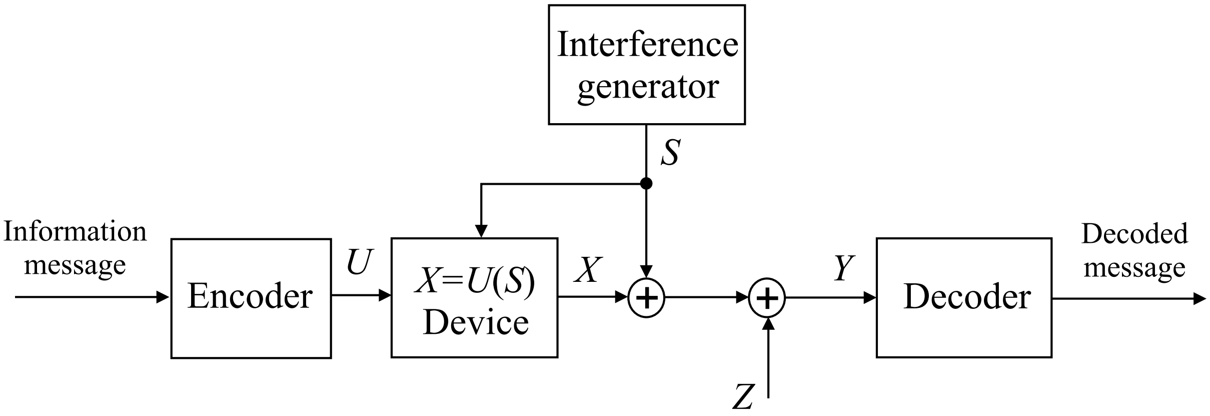


Figure 6.7: General scheme of DTC coding: the SI refer to additive interference present in the transmission channel.

state and not on the previous ones. Thus the generic U can be seen as a "strategy" independent of S, defining the value of the current channel input X based only on the current interference sample S.

6.5 Dirty tape coding

The classical DTC scheme has been reported in Fig. 6.7. As in the DPC case, the SI is relative to an additive interferer along the channel. Even if the capacity result by Shannon (6.44) dates back to 1953, there are more open issues concerning DTC strategies than DPC. For instance, the capacity of the Gaussian DTC is still unknown. Some general bounds have been found, based on the performance of an universal coding scheme independent of the interference statistics and the worst-case-interference capacity is known for asymptotically high SNR [11]. Basically, the difficulties in finding the answer to this and similar capacity problems arise from the observation that, with causal SI, the capacity does not seem to be independent of the interference statistics. Thus optimal DCT encoding strategies should take it into account, while with DPC the encoder can completely neglect the interference p.d.f..

On the contrary, the capacity of the binary dirty tape channel can be found based on the *coding over strategies* technique suggested by Shannon in [15]. Considering the model for the binary channel already descripted for DPC, the DTC capacity subject to the constraint (6.40) is (see [16])

$$C_{bin}^{dtc}(d) = 2d\,[1 - h(p)]. \tag{6.45}$$

Chapter 7

A low complexity DPC scheme

In the last few years a great interest toward the development of coding techniques for dirty paper scenarios has risen. The problem, originally treated by M. Costa in [13], is strictly related to key and still open issues in digital communications, like coding for MIMO broadcast channels (BC) , data-hiding and watermarking schemes. According to the general scheme in Fig. 7.1, if the transmitter knows in advance the whole interference which will affect its signal at the receiver side, it may apply an appropriate strategy in order to exploit it rather than simply ignoring or cancelling it. Costa proved that, with proper codes, it is possible to reach the same capacity as if the interference is completely absent. A scenario involving a perfect knowledge of the interference is, for instance, the multi-user MIMO downlink communications, where it is possible to encode in succession the data for the different users and so to interpret the previously coded signals as known interference for the others. Some papers have appeared in the last years proposing different co-decoding schemes based on linear and lattice codes [3] [4]. Very recent works showed that DPC for low SNR generally require highly complex co-decoders [5]. In this chapter we propose a low-complexity 1/2 rate co-decoding scheme, involving basically a LDPCC and a vector quantizer (VQ) based on a convolutional code (CC) properly combined. This solution is easily generalizable to obtain different code rates.

7.1 System Model

In the following we will indicate vectors in bold, random variables in capital case and their generic samples in lower case. Moreover we will consider indifferently symbols in $\mathbb{C}$ or in $\mathbb{R}^2$, thanks to the isomorphism between the two sets.

Basically, writing on dirty paper (WDP) deals with co-decoding schemes for channels

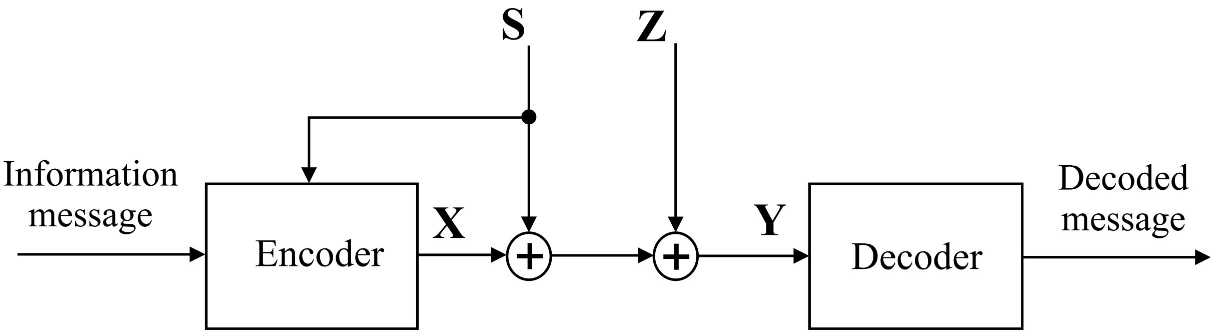


Figure 7.1: General transmission model representing the WDP problem.

with interference non-causally known by the transmitter, as depicted in Fig. 7.1, and with a constraint on the mean transmitted power P_X. In the following we consider discrete-time signals. The interference $\mathbf{S} = [S_1 \ ... \ S_k]$ is a k-element vector with zero-mean independent, identically distributed (i.i.d.) complex Gaussian entries with independent real and imaginary parts having equal variance $P_S/2$. Thus the mean interference power is $\mathbb{E}[|S_i|^2] = P_S$. The channel is additive with regard to both $\mathbf{S}$ and the noise sequence $\mathbf{Z} = [Z_1 \ ... \ Z_k]$, whose elements are zero-mean i.i.d. complex Gaussian, with independent real and imaginary parts with the same variance $P_N/2$; the mean noise power is $\mathbb{E}[|Z_i|^2] = P_N$. As demonstrated by Costa in [13], if the whole interference sequence is known in advance by the encoder, it is possible to reach the same capacity as without any interferer. In particular, by using the results by Gelfand and Pinsker [12] and statistically modelling the transmitted symbol as $X = U - \alpha S$, where U is a proper auxiliary random variable and α a scaling factor, the capacity of the system with known interference is:

$$C = \log_2 \left(1 + \frac{P_X}{P_N}\right) = \log_2 (1 + \mathrm{SNR}) \quad \text{[bits/symb]} \tag{7.1}$$

where $\mathrm{SNR} = P_X/P_N$. The random variable U is assumed to have a complex Gaussian distribution, so that also the transmitted symbol has a zero-mean complex Gaussian p.d.f.. The scaling factor α is in $(0, 1)$ and allows for a partial pre-cancellation of the interference $\mathbf{S}$. As discovered by Costa, a capacity achieving coding strategy entails an intermediate value of α, depending on the received SNR. Costa showed that $\alpha_{opt} = P_X/(P_X + P_N)$ and, in particular, it does not depend on P_S. In our coding architecture, as it will become more clear in section 7.3, the residual interference acts like an additional noise component [3][4]. Choosing $\alpha = \alpha_{opt}$ it is possible to minimize the overall noise power at the receiver side.

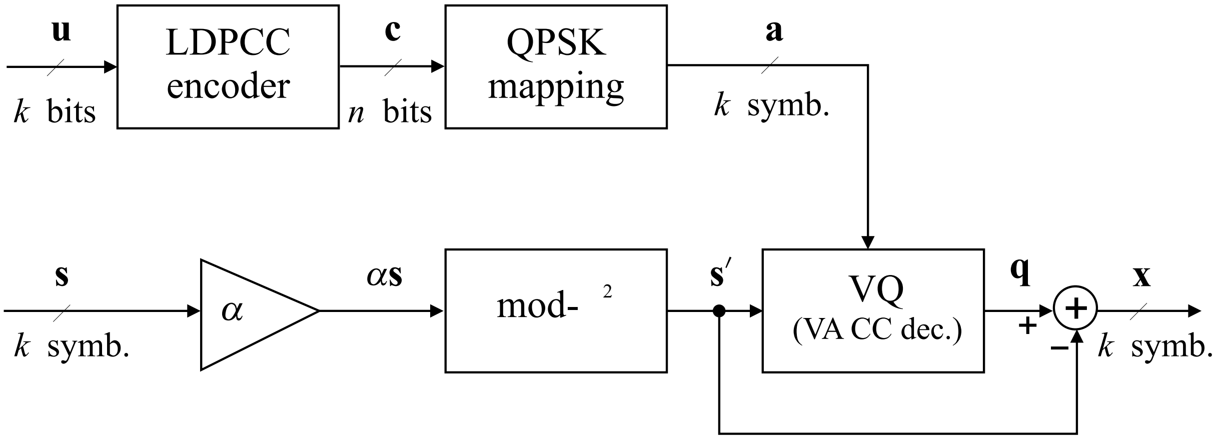


Figure 7.2: Scheme of the proposed WDP encoder (transmitter side).

7.2 Encoding Scheme

Our main idea behind the coding scheme is to exploit the non-causal knowledge of the interference vector, finding the modulated codeword nearest to the shrunk version of the interference. This operation is performed by a VQ, while an LDPCC ensures the necessary data protection. In Fig. 7.2 the proposed WDP encoding scheme has been depicted. A block of k message bits $\mathbf{u} = [u_1, ..., u_k]$ is processed by a rate-1/2 LDPCC encoder, whose output is a codeword $\mathbf{c} = [c_{1,0}, c_{1,1}, ..., c_{k,0}, c_{k,1}]$ of length $n = 2k$. Each couple of coded bits $[c_{i,0}, c_{i,1}]$, with $i = 1, ..., k$, is then mapped on a simple quadrature phase shift keying (QPSK) constellation $\mathcal{A}$, according to a Gray labelling. $\mathcal{A}$ is defined as the set $\{\pm\Delta/8 \pm j\Delta/8\}$, where Δ is a real parameter whose meaning will become clear later. The k complex symbols $\mathbf{a} = [a_1, ..., a_k]$ at the output of the mapper will become a sequence of offsets for the constellation on which the VQ works. Each element of the complex interference sequence $\mathbf{s} = [s_1, ..., s_k]$ is properly scaled by the real factor α and processed by the non-linear mod-Δ^2 block. Given a certain input symbol αs_i, it provides the symbol within the set of all complex points $\{\alpha s_i + l\Delta + jr\Delta : l, r \in \mathbb{Z}\}$ falling in the region $\mathcal{R} = \{x + jy : x, y \in [-\Delta/2, \Delta/2)\}$. Thus, the i^{th} output symbol, indicated with s'_i, depends only on the corresponding input and is constrained to be within the two-dimensional square region centered in the origin and with side Δ, i.e.

$$s'_i \triangleq (\alpha s_i) \,\text{mod-}\Delta^2 \tag{7.2}$$

so that $s'_i \in \mathcal{R}$. According to basic lattice theory, $\mathcal{R}$ is a fundamental region for the lattice $\Lambda_\Delta = (\Delta\mathbb{Z})^2$. The mod-$\Delta^2$ block is one of the constituent element of many precoding

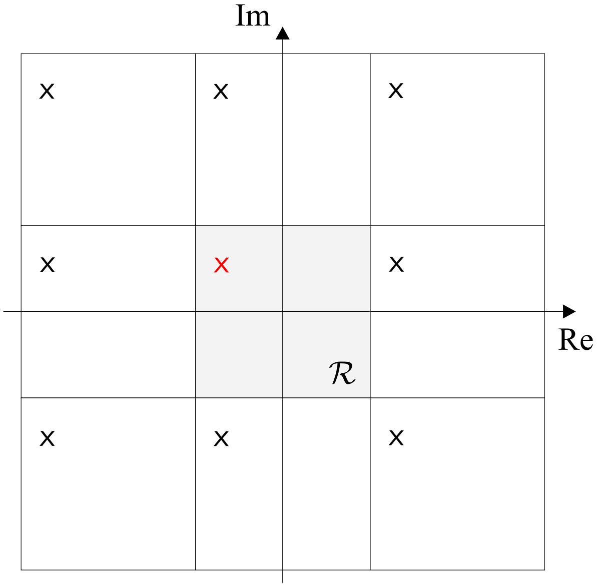


Figure 7.3: Equivalency among points in $\mathbb{R}^2$ as established by the mod-Δ^2 operator.

schemes, like the Tomlinson-Harashima precoding or the flexible precoding [7]. In essence, it allows easily for operating with points in the whole complex plane by establishing an equivalency relation between any two points whose difference is an element of Λ_Δ, as graphically shown in Fig. 7.3.

The scaling factor α, as introduced in section 7.1, is included in $(0, 1)$ and depends on the SNR, according to the relation $\alpha = P_X/(P_X + P_N)$. So for decreasing SNR α tends to zero, while for increasing SNR to one.

As shown in Fig. 7.2, the symbol sequence $\mathbf{s}'$ is then quantized by the VQ. The quantizer is based on a rate-1/2 CC decoder which finds the codeword minimizing the Euclidean distance from $\mathbf{s}'$ through the Viterbi algorithm (VA). The constellation on which the codewords of the CC are mapped is squared and is defined as the set of points $\mathcal{B} = \{\pm\Delta/4 \pm j\Delta/4\}$. During the decoding/quantizing processing $\mathbf{a}$ acts as a sequence of offsets for the constellation $\mathcal{B}$: at the i^{th} time instant the branch metrics of the VA are computed starting from a proper translated version of $\mathcal{B}$, i.e. from the points in the set $\mathcal{B} + a_i = \{b + a_i \colon b \in \mathcal{B}\}$. More exactly, the search is done considering all the complex points which are mod-Δ^2 equivalent to the points in $\mathcal{B} + a_i$. In other words, given a certain a_i, the generic branch metric corresponding to the symbol $b \in \mathcal{B}$ is computed as

$$d(s_i', b, a_i) = \min_{w \in \{b + a_i + l\Delta + jr\Delta \colon l, r \in \mathbb{Z}\}} |s_i' - w|^2. \qquad (7.3)$$

Actually for each branch the searching set is extremely limited, because s'_i is constrained to be in $\mathcal{R}$. As a consequence the output sequence from the quantizer has elements which are always in the set

$$\{a + b + l\Delta + jr\Delta \colon a \in \mathcal{A}, b \in \mathcal{B} \text{ and } l, r \in \{0, \pm 1\}\}\,. \tag{7.4}$$

The overall encoding strategy can be described more clearly using the lattice theory notation. Let's define the lattices

$$\Lambda_A = \left(\frac{\Delta}{4}\mathbb{Z}\right)^2 \quad \text{and} \quad \Lambda_B = \left(\frac{\Delta}{2}\mathbb{Z}\right)^2. \tag{7.5}$$

Now the constellations $\mathcal{A}$ and $\mathcal{B}$ may be viewed as subsets of proper translations of Λ_A and Λ_B respectively:

$$\begin{aligned} \mathcal{A} &\leftrightarrow \left(\Lambda_A + \begin{bmatrix} \Delta/8 \\ \Delta/8 \end{bmatrix}\right) \cap \left(-\frac{\Delta}{4}, \frac{\Delta}{4}\right) \times \left(-\frac{\Delta}{4}, \frac{\Delta}{4}\right) \\ \mathcal{B} &\leftrightarrow \left(\Lambda_B + \begin{bmatrix} \Delta/4 \\ \Delta/4 \end{bmatrix}\right) \cap \left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right) \times \left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right). \end{aligned} \tag{7.6}$$

The lattice partition chain $\Lambda_A/\Lambda_B/\Lambda_\Delta$, reported in Fig. 7.4, is the key of the whole encoding process. Λ_A/Λ_B subdivides Λ_A in four cosets, each one identifiable with a proper translation of Λ_B, i.e. Λ_B, $\Lambda_B + [0, \Delta/4]^T$, $\Lambda_B + [\Delta/4, 0]^T$ and $\Lambda_B + [\Delta/4, \Delta/4]^T$, where $[\cdot]^T$ indicates vector transposition. Translating each coset by $[\Delta/8, \Delta/8]^T$, we obtain four sets of points, to which the four possible couples of LDPC-coded bits are associated. From the co-decoder point of view, all the points in the same set are equivalent. Similarly the lattice partition Λ_B/Λ_Δ induces the subdivision of Λ_B in four cosets. Each branch of the CC trellis is associated to one of these cosets translated by a proper point in the LDPCC-selected set. The VA chooses the best sequence of symbols locating for each branch metric computation the point nearest to s'_i within the correspondent set. Obviously, because of the VA, the best choice is to be intended in the maximum-likelihood sequence estimation (MLSE) sense. An example of the quantization process has been reported in Fig. 7.5, where the quantization alphabets used in the different time instants have been expressly indicated.

The transmitted signal sequence is constituted by the obtained quantization noise, i.e.

$$\mathbf{x} = \mathbf{q} - \mathbf{s}' \tag{7.7}$$

where $\mathbf{q} = \mathbf{q}(\mathbf{s}', \mathbf{a})$ is the quantized output from the VQ. Thus the generic transmitted symbol X_i does not belong to a finite constellation, but has a continuous statistical distribution over the support $[-\frac{\Delta}{2}, \frac{\Delta}{2}] \times [-\frac{\Delta}{2}, \frac{\Delta}{2}]$. The mean transmitted power P_X coincides with the expected distortion per complex symbol, defined as

$$D = \mathbb{E}[|X_i|^2] = \mathbb{E}[|Q_i - S_i'|^2]. \tag{7.8}$$

As it can be easily seen, the quantization distortion D is quadratic in Δ.

The sequence $\mathbf{s}'$ output from the mod-Δ^2 block can be considered as a point in a $2k$-dimensional observation space, which is an hyper-cube with volume Δ^{2k}. Provided that the interference power P_S is large enough, $\mathbf{s}'$ is uniformly distributed over this observation space. Otherwise a dithering sequence of proper power, entirely known by the transmitter and by the receiver, can be added to $\mathbf{s}$ in order to make $\mathbf{s}'$ approximately uniform. The decision region associated with a given codeword of the CC has a volume of

$$V = \frac{\Delta^{2k}}{2^k}. \tag{7.9}$$

In Fig. 7.6, D is plotted for several CCs with different values of constraint length K and for increasing Δ's. All the CCs are optimal in the sense that they have the maximum

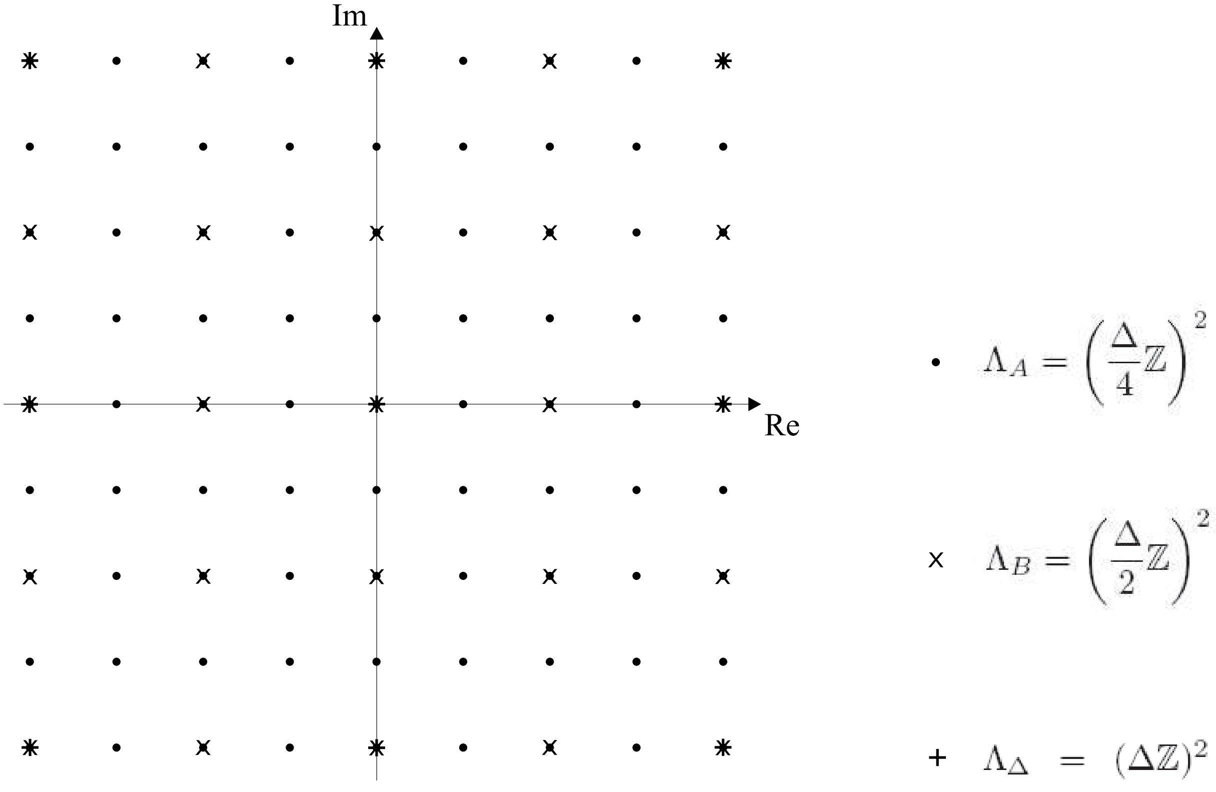
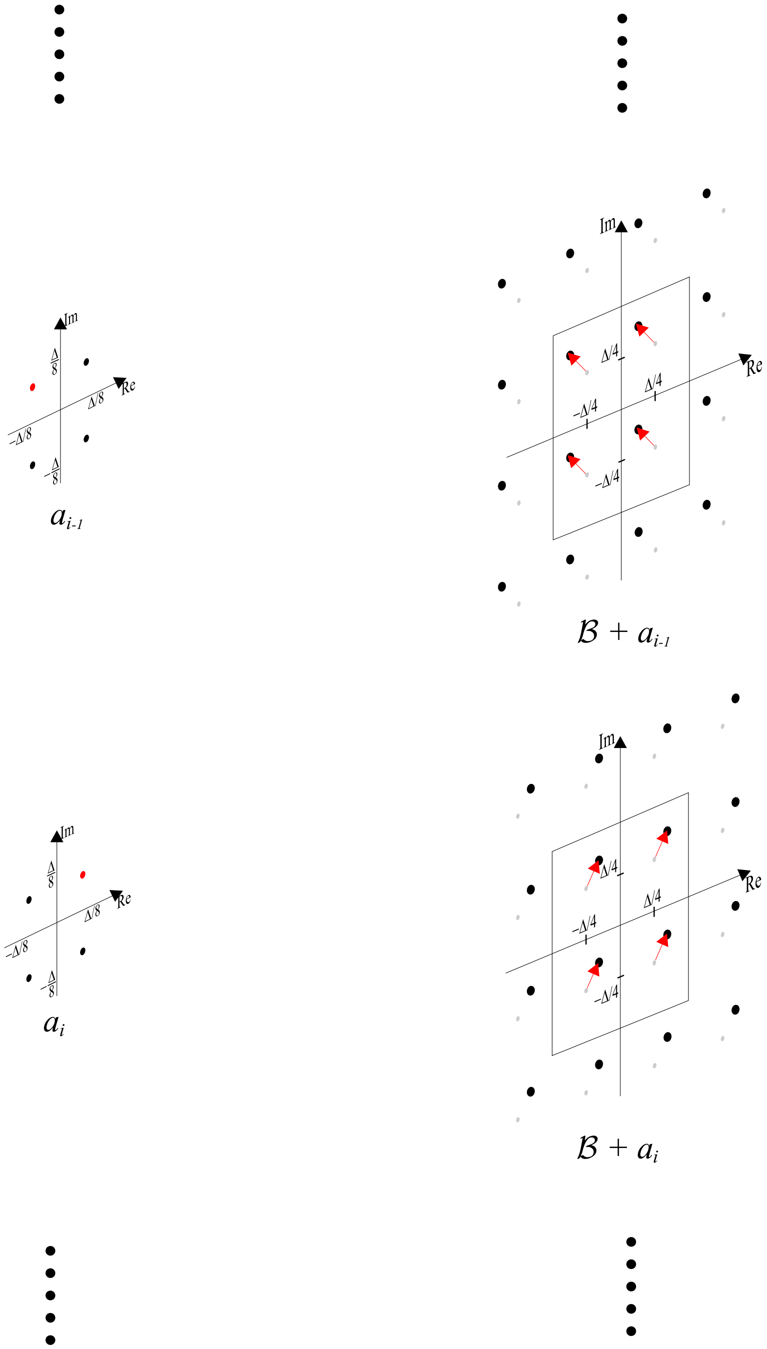

Im
Re
• $\Lambda_A = \left(\frac{\Delta}{4}\mathbb{Z}\right)^2$
x $\Lambda_B = \left(\frac{\Delta}{2}\mathbb{Z}\right)^2$
+ $\Lambda_\Delta = (\Delta\mathbb{Z})^2$

Figure 7.4: The lattice partition chain $\Lambda_A/\Lambda_B/\Lambda_\Delta$.



a_{i-1}

$\mathcal{B} + a_{i-1}$

a_i

$\mathcal{B} + a_i$

Figure 7.5: An example of quantization process: the translations of the basic quantization points have been indicated with arrows.

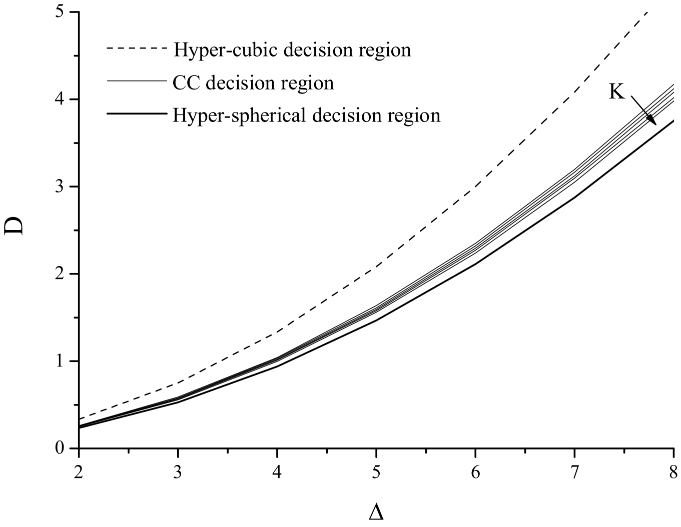


Figure 7.6: Average distorsion per 2-dim (i.e. complex) symbol obtained with CC's with different K.

free distance for the given K's. Approximating the generic decision region with an hyper-sphere with equal volume V, we obtain the lower-bound curve depicted with the thick line. In fact, the volume of an n-dimensional hyper-sphere is $V_s = V_n R^n$, where

$$V_n = \frac{\pi^{n/2}}{(n/2)!} = \frac{2^n \pi^{(n-1)/2}((n-1)/2)!}{n!} \tag{7.10}$$

as it can be found in [8]. Thus, the radius of a $2k$-dimensional sphere with volume V becomes

$$R = \sqrt[2k]{\frac{V}{V_{2k}}} = \frac{\Delta}{\sqrt{2}\,\sqrt[2k]{V_{2k}}} \tag{7.11}$$

and the expected distortion per 2-dimensions obtained in case of points uniformly distributed over the hyper-sphere is

$$D_s = \frac{1}{2k+2}\,\frac{\Delta^2}{\sqrt[k]{V_{2k}}}. \tag{7.12}$$

As it can be noted from Fig. 7.6, increasing the constraint length (and so also the free distance) of the code, the decision regions tend to become more and more "spherical" and, consequently, the mean transmitted power decreases. The dashed line is a loose upper-bound curve, representing the average distortion if the decision region had a perfect $2k$-dimensional cubic shape. According to [9], the quantization noise obtained by an optimal lattice quantizer would converge to a white Gaussian distribution increasing the

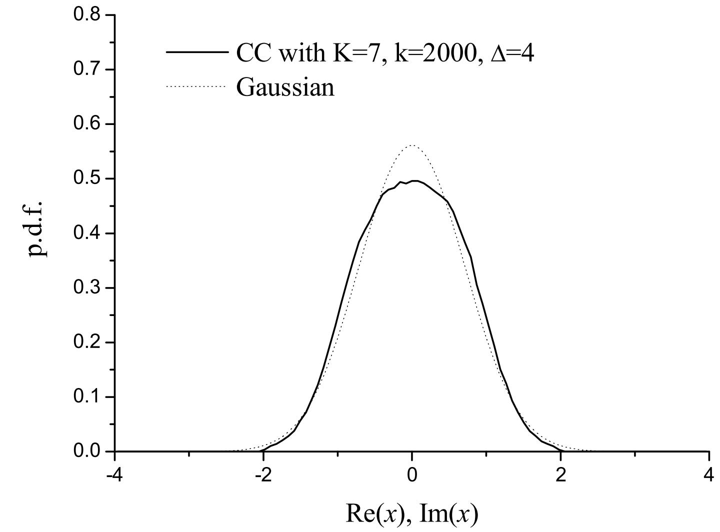


Figure 7.7: p.d.f. obtained through simulations.

number of dimensions. In our case the $2k$-dimensional quantizer is surely sub-optimal, but $\mathrm{Re}(X_i)$ and $\mathrm{Im}(X_i)$ have a p.d.f. which can be roughly approximated with a Gaussian distribution, as it can be seen in Fig. 7.7. So, approximately, X_i has a zero-mean complex Gaussian distribution. In particular the p.d.f. of X_i may impact on the structure of the iterative decoding scheme [4]. As we will see in section 7.3, the pseudo-Gaussianity of the transmitted symbol will allow us to exploit classical convolutional and LDPCC decoders.

7.3 Decoding Scheme

In Fig. 7.8 it is depicted the proposed decoding scheme, mainly composed by the two decoder blocks: the CC and the LDPCC decoders. According to the channel model described in section 7.1, the i^{th} received symbol can be written as

$$y_i = x_i + s_i + z_i \tag{7.13}$$

where s_i is the interference and z_i is the noise sample. The received sample is scaled by the real factor α and processed by a mod-Δ^2 block. Exploiting the properties of the

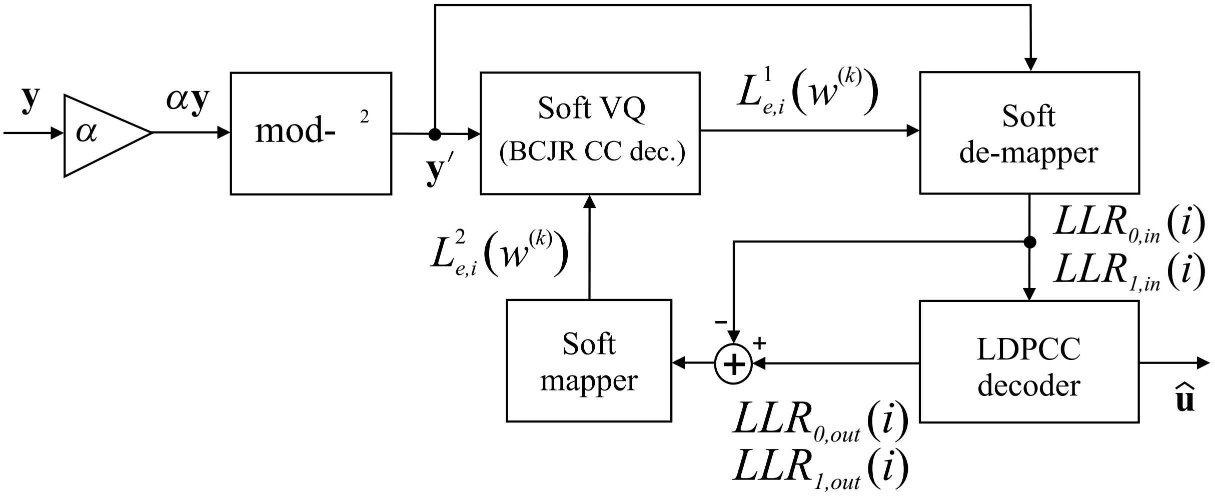


Figure 7.8: Scheme of the WDP decoder (receiver side).

mod-Δ^2 operator, we obtain

$$\begin{aligned}
y'_i &\triangleq (\alpha y_i)\text{mod-}\Delta^2 = \\
&= (\alpha x_i + \alpha s_i + \alpha z_i)\text{mod-}\Delta^2 = \\
&= (\alpha x_i + (\alpha s_i)\text{mod-}\Delta^2 + \alpha z_i)\text{mod-}\Delta^2 = \\
&= (\alpha x_i + (q_i - x_i)\text{mod-}\Delta^2 + \alpha z_i)\text{mod-}\Delta^2 = \\
&= (q_i - (1-\alpha)x_i + \alpha z_i)\text{mod-}\Delta^2 = \\
&= (q_i + z_{eq,i})\text{mod-}\Delta^2
\end{aligned} \tag{7.14}$$

where $z_{eq,i} = -(1-\alpha)x_i + \alpha z_i$ is the equivalent noise seen by the receiver. It is composed by two statistically independent parts: the first contribute due to the transmitted quantization noise and the second one to the thermal noise. For decreasing SNRs, α lowers and the term $(1-\alpha)x_i$ tends to prevail on αz_i. On the contrary, for high SNRs, α converges to 1 and $(1-\alpha)x_i$ tends to vanish. As far as the p.d.f. of X_i can be approximated by a Gaussian one, $Z_{eq,i}$ has a p.d.f. which is also complex Gaussian, with zero mean and power $P_{Neq} = \mathbb{E}[|Z_{eq,i}|^2] = \alpha^2 P_N + (1-\alpha)^2 P_X$. In any case, even if we consider that the actual p.d.f. of the transmitted symbol is only approximately Gaussian, its weighted convolution with the p.d.f. of the noise Z_i tends to approximate quite well a Gaussian one. It is now more clear why the choice $\alpha = \alpha_{opt} = P_X/(P_X + P_N)$ is the optimal strategy: as it can be easily proved, this is the value which minimizes the power of $Z_{eq,i}$.

Because of the mod-Δ^2 operator, each y'_i is in $\mathcal{R}$. However, at least in principle, a correct decoding should take into account the infinite constellation represented by the

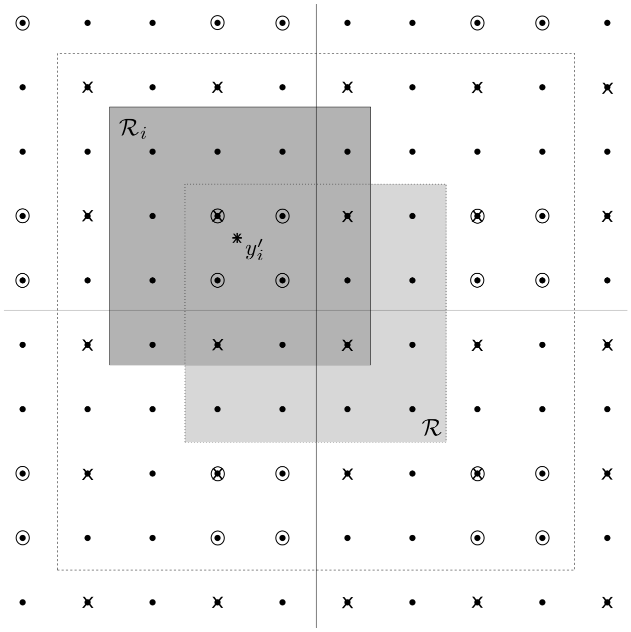


Figure 7.9: A portion of the infinite constellation at the receiver side and the search window corresponding to y'_i: in particular, one of the 4 sets of points equivalent from the point of view of the CC has been represented with small circles and one of the 4 sets of points equivalent for the LDPCC with crosses.

translated set $\Lambda_A + [\Delta/8, \Delta/8]^T$. In fact, in the presence of noise (and this is particularly true for low SNRs), it becomes fundamental to consider that the mod-Δ^2 processing always determines a loss of information, because it cannot distinguish between any couple of points which differs of $(l\Delta + jr\Delta)$, with $l, r \in \mathbb{Z}$. A good trade-off between complexity and performance can be reached considering only the points in a finite search-window. Let's define the search-window as $\mathcal{R}_i = \mathcal{R} + y'_i$, i.e. as the squared region centered in y'_i and with side Δ, as depicted in Fig. 7.9. We indicate with $\mathcal{W}_i$ the QAM constellation obtained as

$$\mathcal{W}_i = \left(\Lambda_A + \left[\begin{array}{c} \Delta/8 \\ \Delta/8 \end{array} \right] \right) \cap \mathcal{R}_i. \tag{7.15}$$

So, according to our scheme, the 16QAM constellation within $\mathcal{R}_i$ is considered for the demodulating/decoding process. By numerical simulations, we found that such a window size is sufficient in order to obtain good results. In the following we indicate with $w^{(k)}$ the k^{th} point in $\mathcal{W}_i$, with $k = 1, ..., 16$. The basic principle on which the decoding algorithm is based follows from the observation that the quantized sequence $\mathbf{q}$ represents a modulated codeword for the CC and for the LDPCC at the same time. This is possible thanks to the equivalency properties existing among the points in the observation space, which has been described in section 7.2. Thus the two decoders can exchange iteratively soft information about the sequence of received symbols.

The CC decoder is based on a log-domain soft-output Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm which computes the a-posteriori information about the symbol sequence $\mathbf{y}'$

$$\widetilde{\Pr}\left(Q_i = w^{(k)}|\mathbf{y}'\right) = \ln \Pr\left(Q_i = w^{(k)}|\mathbf{y}'\right). \tag{7.16}$$

From these quantities the extrinsic information are extracted and passed to the LDPCC encoder. Let's now introduce some notation in order to explain the small changes needed by the convolutional decoding algorithm. We indicate with σ and σ' two generic states of the CC and with

$$\begin{aligned} \widetilde{\alpha}_i(\sigma) &= \ln \alpha_i(\sigma) \\ \widetilde{\beta}_i(\sigma) &= \ln \beta_i(\sigma) \end{aligned} \tag{7.17}$$

the logarithmic forward and backward metrics respectively, computed for the state σ at time instant i, and with

$$\widetilde{\gamma}_i\left(w^{(k)}\right) = \ln \gamma_i\left(w^{(k)}\right) \tag{7.18}$$

the logarithmic branch metric at the same time and referring to a state transition which provides the symbol $w^{(k)}$ as a possible output. In fact it is important to remember that the same state transition now corresponds to several output symbols in $\mathcal{W}_i$, according to the equivalency among points for the CC. Adhering to the classical notation, we define the function

$$\begin{aligned} \max{}^*(x, y) &= \ln(e^x + e^y) = \\ &= \max(x, y) + \ln\left(1 + e^{-|x-y|}\right). \end{aligned} \tag{7.19}$$

According to [10], from several applications of the Bayes' rule and from some properties of the max* function, it results that

$$\begin{aligned}
&\widetilde{\Pr}\left(Q_i = w^{(k)}|\mathbf{y}'\right) = \\
&\quad = \ln p\left(w^{(k)}, \mathbf{y}'\right) - \ln p\left(\mathbf{y}'\right) = \\
&\quad = \ln \sum_{\sigma' \xrightarrow{w^{(k)}} \sigma} p\left(\sigma', \sigma, w^{(k)}, \mathbf{y}'\right) - \ln p\left(\mathbf{y}'\right) = \\
&\quad = \ln \sum_{\sigma' \xrightarrow{w^{(k)}} \sigma} \exp\left[\widetilde{\alpha}_{i-1}(\sigma') + \widetilde{\beta}_i(\sigma) + \widetilde{\gamma}_i\left(w^{(k)}\right)\right] - \ln p\left(\mathbf{y}'\right) = \\
&\quad = \max_{\sigma' \xrightarrow{w^{(k)}} \sigma}{}^{*} \left[\widetilde{\alpha}_{i-1}(\sigma') + \widetilde{\beta}_i(\sigma) + \widetilde{\gamma}_i\left(w^{(k)}\right)\right] - \ln p\left(\mathbf{y}'\right) = \\
&\quad = \max_{\sigma' \xrightarrow{w^{(k)}} \sigma}{}^{*} \left[\widetilde{\alpha}_{i-1}(\sigma') + \widetilde{\beta}_i(\sigma)\right] + \widetilde{\gamma}_i\left(w^{(k)}\right) - \ln p\left(\mathbf{y}'\right)
\end{aligned} \tag{7.20}$$

where $\sigma' \xrightarrow{w^{(k)}} \sigma$ indicates the set of all the possible state transitions which may correspond to the output $w^{(k)}$. If we denote with $\mathcal{W}_{i,\sigma'}^{\sigma}$ the set of all the points in $\mathcal{W}_i$ equivalently corresponding to the state transition $\sigma' \rightarrow \sigma$ and with $\Pr_i\left(w^{(k)}\right) = \Pr\left(Q_i = w^{(k)}\right)$ the a-priori symbol probability, it is easy to show that

$$\begin{aligned}
\widetilde{\alpha}_i(\sigma) &= \max_{\sigma', w^{(k)} \in \mathcal{W}_{i,\sigma'}^{\sigma}}{}^{*} \left[\widetilde{\alpha}_{i-1}(\sigma') + \widetilde{\gamma}_i\left(w^{(k)}\right)\right] \\
\widetilde{\beta}_{i-1}(\sigma') &= \max_{\sigma, w^{(k)} \in \mathcal{W}_{i,\sigma'}^{\sigma}}{}^{*} \left[\widetilde{\beta}_i(\sigma) + \widetilde{\gamma}_i\left(w^{(k)}\right)\right]
\end{aligned} \tag{7.21}$$

where the branch metric (assuming a complex Gaussian p.d.f. for $Z_{eq,i}$), is

$$\begin{aligned}
&\widetilde{\gamma}_i\left(w^{(k)}\right) = \\
&\quad = -\ln\left(\pi P_{N_{eq}}\right) - \frac{1}{P_{N_{eq}}}\left|y_i' - w^{(k)}\right|^2 + \ln \Pr_i\left(w^{(k)}\right).
\end{aligned} \tag{7.22}$$

The initial condition for the iterative computation of the forward and backward metrics are the same as in [10].

The extrinsic information to be passed to the LDPCC decoder block become

$$\begin{aligned}
L_{e,i}^{1}\left(w^{(k)}\right) &= \widetilde{\Pr}\left(Q_i = w^{(k)}|\mathbf{y}'\right) - \widetilde{\gamma}_i\left(w^{(k)}\right) = \\
&= \max_{\sigma' \xrightarrow{w^{(k)}} \sigma}{}^{*} \left[\widetilde{\alpha}_{i-1}(\sigma') + \widetilde{\beta}_i(\sigma)\right] - \ln p\left(\mathbf{y}'\right).
\end{aligned} \tag{7.23}$$

From Eq. (7.23) (as we may expect), it is clear that the decoder needs to compute only four $L_{e,i}^{1}$ at each time instant, one for each class of points equivalent with respect to the

CC. Moreover, as it will be evident in the following, it is not necessary to calculate the quantities $\ln p\left(\mathbf{y}'\right)$.

The soft de-mapper exploits the extrinsic information provided by the BCJR decoder as a-priori information on the symbols in order to compute the log-likelihood ratio for each bit of the LDPCC codeword, i.e. the couple $(LLR_{0,in}(i), LLR_{1,in}(i))$ for each $i = 1, ..., k$:

$$\begin{aligned}
LLR_{0,in}(i) &= \ln \frac{\Pr\left(c_{0,i}=0|y'_i\right)}{\Pr\left(c_{0,i}=1|y'_i\right)} = \\
&= \ln \frac{\sum_{w^{(k)}\in\mathcal{A}^{(0)}_{i,0}} p\left(y'_i|w^{(k)}\right)\Pr\left(w^{(k)}\right)}{\sum_{w^{(k)}\in\mathcal{A}^{(1)}_{i,0}} p\left(y'_i|w^{(k)}\right)\Pr(w^{(k)})} = \\
&= \ln \sum_{w^{(k)}\in\mathcal{A}^{(0)}_{i,0}} \exp\left[-\frac{1}{P_{N_{eq}}}\left|y'_i - w_i^{(k)}\right|^2 + \ln \Pr_i\left(w^{(k)}\right)\right] + \\
&- \ln \sum_{w^{(k)}\in\mathcal{A}^{(1)}_{i,0}} \exp\left[-\frac{1}{P_{N_{eq}}}\left|y'_i - w_i^{(k)}\right|^2 + \ln \Pr_i\left(w^{(k)}\right)\right] = \\
&= \max_{w^{(k)}\in\mathcal{A}^{(0)}_{i,0}}{}^{*}\left[-\frac{1}{P_{N_{eq}}}\left|y'_i - w_i^{(k)}\right|^2 + L^1_{e,i}\left(w^{(k)}\right)\right] + \\
&- \max_{w^{(k)}\in\mathcal{A}^{(1)}_{i,0}}{}^{*}\left[-\frac{1}{P_{N_{eq}}}\left|y'_i - w_i^{(k)}\right|^2 + L^1_{e,i}\left(w^{(k)}\right)\right]
\end{aligned} \tag{7.24}$$

where we have indicated with $\mathcal{A}^{(0)}_{i,0}$ and $\mathcal{A}^{(1)}_{i,0}$ the set of all the points in $\mathcal{W}_i$ corresponding to $c_{i,0} = 0$ and $c_{i,0} = 1$ respectively and, again, we have modelled $Z_{eq,i}$ as complex Gaussian. Obviously, an analog expression holds for $LLR_{1,in}(i)$. The LDPCC decoder can now iteratively process the input LLR's. In our simulations we adopted a classical belief propagation (BP) algorithm: at each iteration step the decoder checks the parity equations and, if a valid codeword is reached, the decoding stops and the decoded bits $\widehat{\mathbf{u}}$ are provided as output. However, if it is not able to converge after a pre-fixed number of iterations, the LDPCC decoder sends back extrinsic information about the modulated symbol sequence to the convolutional decoder, and all the process restarts. According to Fig. 7.8, starting from the differences $LLR_{0,out}(i) - LLR_{0,in}(i)$ and $LLR_{1,out}(i) - LLR_{1,in}(i)$ the soft mapper calculate the extrinsic information $L^2_{e,i}\left(w^{(k)}\right)$ acting as a-priori information for the BCJR algorithm. Only if, after a given number of information exchanges between the decoders, the BP algorithm still does not converge, the process stops and the partially decoded sequence $\widehat{\mathbf{u}}$ is given as output.

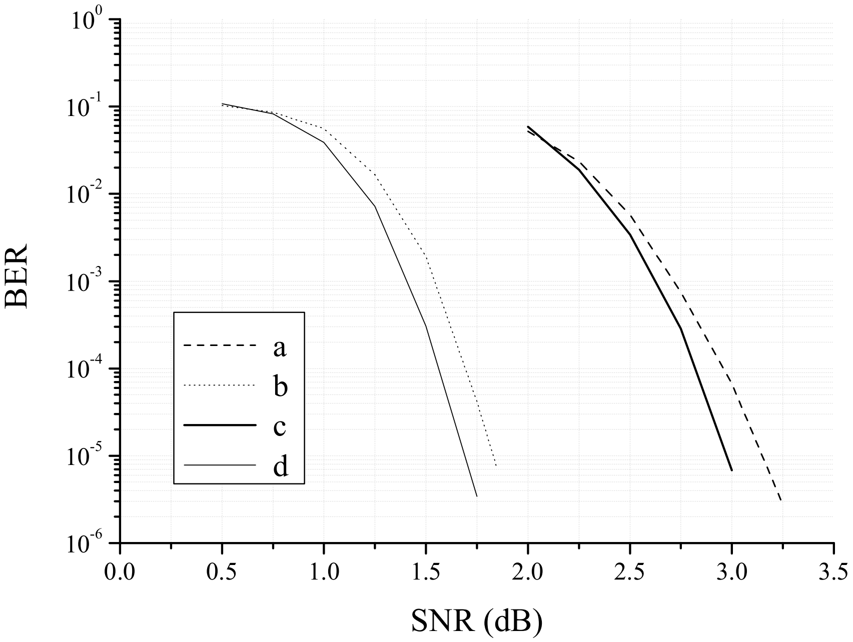


Figure 7.10: BER comparison between LDPCC's in the WDP scenario and their correspondents in AWGN channels: a) WDP with Reg. LDPCC and CC with $K = 7$, b) Reg. LDPCC over AWGN, c) WDP with IRA LDPCC and CC with $K = 7$, d) IRA LDPCC over AWGN; for all the curves $k = 2000$.

7.4 Simulation results

In this section we show some numerical results obtained with the proposed algorithm. We considered a WDP scenario, with $P_S >> P_X$. It is worth recalling that the performance of the proposed scheme are completely unaffected by the interference power. We considered different CC's, all with code rate 1/2, but characterized by different constraint lengths. Moreover, we have run simulations with both regular and irregular LDPCCs with rate 1/2. Following from the previous sections, the co-decoding scheme provides on the whole 0.5 bits/dimension. Of course, increasing K and the word length n=2k, the overall performance improve, at cost of an increase of complexity. Moreover, also the number of internal iterations of the LDPCC decoder and the number of external iterations between the decoders influence the performance. A good trade-off may be the argument of further studies. We found that 100 iteration for the LDPCC and 2-3 external iterations with the CC permit to reach quite good performance also at low SNRs, while keeping a low-complexity with respect to other proposed schemes [5], because of the reduced number of BCJR decoding processes required. In Fig. 7.10 we illustrate a comparison between the BER curves obtained with our scheme for WDP and the corresponding BER curve obtained with the same LDPCCs over AWGN channels at the same rate of 0.5 bits/dimension. $K = 7$ and $k = 2000$ have been considered and the generator polynomials for the CC are $(133, 171)$ [34]. The regular LDPCC is a (3,6) code, while the irregular one is a systematic IRA code [38][39][43]. At a BER of 10^{-5} the gap between the WDP curves and the AWGN case is about 1.3 dB, showing that it is possible to obtain good results by exploiting classical and already existing decoding blocks. In Fig. 7.11, it is shown how the performance change if we consider different values for K and k. In particular the curves relative to a CC with $K = 4$, generator polynomials $(15, 17)$ [34], $k = 2000$ and $k = 4000$ have been reported. For all these curves regular LDPCC's have been considered. From the plot it is evident that, for very low SNRs, an increase of code complexity should be done in the direction of higher K's, while increasing the SNR, the LDPCC begins to work in the waterfall region and considerable improvements may be obtained when k increases.

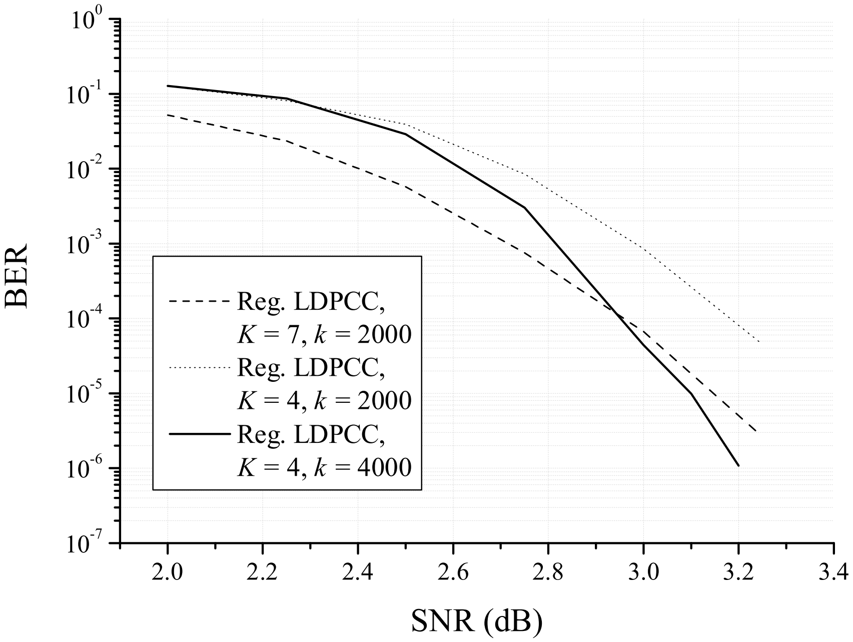


Figure 7.11: BER comparison among different WDP schemes.

Chapter 8

Binary DPC and DTC with NoSI at the decoder

The classical problems of DPC and DTC can be further generalized to the case of some SI available also at the decoder. In this chapter we focus on this topic, showing how in many important scenarios the channel capacity can be improved if the decoder can take advantage of the knowledge of some non-perfect (or noisy) version of the interference. A possible application of such a situation is watermarking for video sequences. In fact, in a video sequence, each frame at the decoder is highly correlated to the previous ones and, while the current frame can be interpreted as the known interference at the encoder, the already received frames can be seen as its noisy version available at the decoder. The results presented in this Chapter constitutes part of the research activities followed during the studying period I spent at the *Multimedia Communication Laboratory* of the *Texas A&M University*, Texas, USA.

8.1 Channels with SI known by the encoder and noisy side information (NoSI) known by the decoder

The general expression for the capacity of discrete memoryless channels when the SI S is available non-causally at the encoder and the SI S_d at the decoder is

$$C = \max_{p(x,u|s)} [I(U;Y,S_d) - I(U;S)] \tag{8.1}$$

where U represents, as usual, a proper auxiliary r.v.. Let's now consider the particular case in which S_d represents a noisy version of S, i.e. with $S_d \leftrightarrow S \leftrightarrow (U,Y)$ forming a

Markov chain. Obviously we have

$$\begin{aligned} I(U;Y,S) - I(U;S) &\geq I(U;Y,S_d) - I(U;S) \\ &= I(U;Y) + I(U;S_d|Y) - I(U;S) \\ &\geq I(U;Y) - I(U;S) \end{aligned}$$

where the first inequality follows from the data processing inequality ($U \leftrightarrow S \leftrightarrow S_d$ form a Markov chain by the definition of S_d) and the second one follows from the non-negativity of mutual information. This chain of inequalities is valid for each $p(x,u|s)$ and thus also if we maximize separately each term. From the first term, we obtain the capacity $C_{SI-TR}^{non-caus} = \max_{p(x|s)} I(X;Y|S)$, i.e. the capacity of the channel with SI perfectly known at the encoder and at the decoder, while the third one becomes $C_{SI-T}^{non-caus}$. In [14] it was shown that $C_{SI-TR}^{non-caus} = C_{SI-T}^{non-caus}$ if and only if the maximizing distributions for U, X and Y in the two cases are the same and if with these distributions $I(U;S|Y) = 0$, i.e. $U \leftrightarrow Y \leftrightarrow S$ form a Markov chain. This is a necessary and sufficient condition for the SI at the decoder side being completely useless in terms of capacity. Analogously, the degraded SI S_d at the decoder cannot increase the DPC capacity if and only if, for the maximizing distribution, $I(U;S_d|Y) = 0$, i.e. $U \leftrightarrow Y \leftrightarrow S_d$.

The general expression for the capacity of discrete memoryless channels when SI S are available causally at the encoder and the degraded SI S_d are known by the decoder has not been explicitly given in [15] but the extension of (6.44) to this case is straightforward:

$$C = \max_{p(u)} I(U;Y,S_d) = \max_{p(u)} I(U;Y|S_d). \tag{8.2}$$

In fact it is always possible to see the channel output as constituted by the couple of r.v.s (Y,S_d) and thus to apply the classical result by Shannon. The second equality follows from the observation that, since S_d is a degraded SI, $0 \leq I(U;S_d) \leq I(U;S) = 0$, given the data processing inequality and the independence between U and S. As in the DPC case, if and only if for the maximizing distribution $I(U;S_d|Y) = 0$, then the availability of S_d at the decoder is useless in terms of capacity.

Besides the increment in capacity, it may be interesting to consider when the NoSI at the receiver determines a different optimal distribution for U with respect to the classical DPC/DTC. In the case of a discrete memoryless channel, the Karush-Kuhn-Tucker (KKT) theorem offer necessary and sufficient conditions for the DPC/DTC capacity achieving distribution to be optimal also with NoSI at the decoder. For instance, let's consider DTC subject to the transmitter constraint $E\{f(X)\} \leq c$ (but similar considerations hold

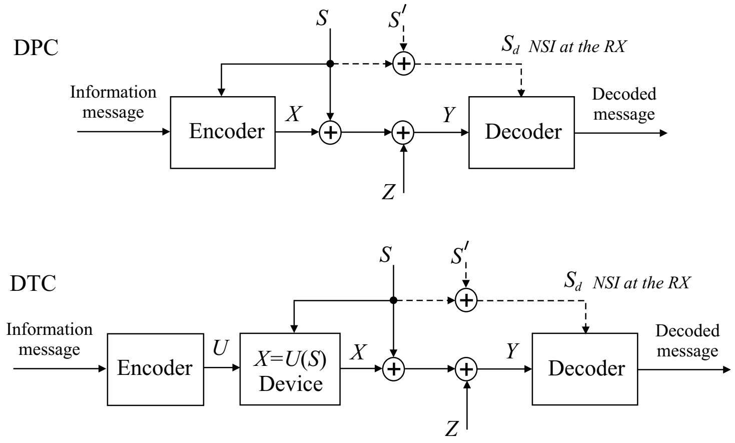


Figure 8.1: General scheme of DPC and DTC coding with NoSI at the decoder.

for DPC): if $\mathcal{X}$ and $\mathcal{S}$ represent the sets of input and SI symbols respectively, the set of possible strategies $\mathcal{U}$ has size $|\mathcal{U}| = |\mathcal{X}|^{|\mathcal{S}|}$. Indicating with $p_o(u)$ the optimal probability mass function (PMF) without any SI at the decoder and applying the KKT conditions, we obtain that if and only if $\forall u \in \mathcal{U}$

$$\begin{cases} I(U=u;Y|S_d) = \mu + \lambda E\{f(u(S))\} & \text{if } p_o(u) > 0 \\ I(U=u;Y|S_d) < \mu + \lambda E\{f(u(S))\} & \text{if } p_o(u) = 0 \end{cases}$$

where μ and λ are real constants, then $p_o(u)$ is an optimal distribution also with NoSI at the decoder.

8.2 DPC on Binary channel NoSI at the decoder

Let's now consider the memoryless channel of Fig. 8.1 in the binary case, i.e. with X, S, S_d, Z and Y binary r.v.s. Let's assume the noise $Z \sim \text{Ber}(p)$ and the interference known at the encoder $S \sim \text{Ber}(1/2)$, statistically independent. Moreover, let's define the NoSI known at the decoder as $S_d = S + S'$, where $S' \sim \text{Ber}(q)$ is independent of S. In the following we report the capacity of this channel in case of both non-causal and causal SI.

Proposition 1. The capacity of the described binary channel with non-causal SI (DPC) and subject to the constraint (6.40) is given by

$$\begin{aligned} C^{dpc}_{b,NoSI}(d) = \sup_{d_1,d_2,\theta} \{ & h(d * p * q) - (1-\theta)h(p * q) \\ & + \theta[h(d_2) - h(d_2 * q) - h(p)]\} \end{aligned} \tag{8.3}$$

where $d_1, d_2 \in [0, 1/2]$ and $\theta \in [0, 1]$, satisfying $(1-\theta)d_1 + \theta d_2 = d$.

From the expression above it is clear that capacity in (6.42) is a particular case of (8.3), corresponding to $q = 1/2$, i.e. S_d and S completely uncorrelated. On the contrary, (8.3) turns into (6.41) if $q = 0$, i.e. if the SI S are perfectly known at the decoder.

Proof.
Let's divide the proof of (8.3) in two parts: first we show that it constitutes an upper bound for any achievable rate and then we provide a distribution capable of achieving (8.3). Following the technique in [14], we consider only deterministic mapping $x = f(u, s)$, i.e. $p(x|u, s) = \delta_{x,f(u,s)}$, where $\delta_{i,j}$ is the Kronecker delta function. Consider a generic alphabet $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$, where $\mathcal{U}_1$ and $\mathcal{U}_2$ are disjoint sets defined as

$$\begin{aligned} \mathcal{U}_1 &= \{u \in \mathcal{U} : \ f(u, s=0) = f(u, s=1)\} \\ \mathcal{U}_2 &= \{u \in \mathcal{U} : \ f(u, s=0) \neq f(u, s=1)\}. \end{aligned} \tag{8.4}$$

Note that with this partitioning of $\mathcal{U}$ the constraint (6.40) can be written as

$$\begin{aligned} d \ &\geq \ E\{d_H(X,0)\} = \sum_{u \in \mathcal{U}_1} p(u) E[d_H(X,0)|U=u] \\ &\quad + \sum_{u \in \mathcal{U}_2} p(u) E\{d_H(X,0)|U=u\} \\ &= \ (1-\theta) \sum_{u \in \mathcal{U}_1} \lambda_{u,1} d_u + \theta \sum_{u \in \mathcal{U}_2} \lambda_{u,2} d_u \\ &= \ (1-\theta) d_1 + \theta d_2 \end{aligned} \tag{8.5}$$

where we have defined $\theta = \Pr\{U \in \mathcal{U}_2\}$, $d_u = E\{d_H(X,0)|U=u\}$, $\lambda_{u,i} = \Pr\{U = u|U \in$

$\mathcal{U}_i\}$ and $d_i = \sum_{u\in\mathcal{U}_i} \lambda_{u,i} d_u$ for $i = 1, 2$. We have

$$\begin{aligned}
I(U;Y,S_d) - I(U;S) &= I(U;Y) + I(U;S_d|Y) - I(U;S) \\
&= H(Y) - H(Y|U) + H(S_d|Y) - H(S_d|U,Y) - H(S) + H(S|U) \\
&\overset{\text{a)}}{\leq} H(S_d|Y) - H(S_d|U,Y) + H(S|U) - H(Y|U) \\
&\overset{\text{b)}}{=} H(X+Z+S'|Y) - H(S_d|U,Y) + H(S|U) - H(Y|U) \\
&\overset{\text{c)}}{\leq} H(X+Z+S') \\
&\quad + \sum_{u\in\mathcal{U}_1} \Pr\{U=u\}[H(S|U=u) - H(S_d|Y,U=u) - H(Y|U=u)] \\
&\quad + \sum_{u\in\mathcal{U}_2} \Pr\{U=u\}[H(S|U=u) - H(S_d|Y,U=u) - H(Y|U=u)]
\end{aligned} \tag{8.6}$$

where a) follows from $H(S)-H(Y) \geq 0$ since $S \sim \text{Ber}(1/2)$, b) depends on the observation that $S_d = Y + X + Z + S'$ and c) follows from the definition of conditioned entropy and the fact that conditioning reduces entropy. Let's now consider separately the terms in (8.6):

$$H(X+Z+S') = h\left([(1-\theta)d_1 + \theta d_2] * p * q\right) \tag{8.7}$$

given the independence among X, Z and S' and the definitions in (8.5). Now consider the terms in the first sum: if $u \in \mathcal{U}_1$, x is determined uniquely by u for the definition of $\mathcal{U}_1$, i.e. $x = f(u,s) = g(u)$. Thus

$$\begin{aligned}
&H(S|U=u) - H(Y|U=u) \\
&\quad = \quad H(S|U=u) - H(S+X+Z|U=u) \\
&\quad = \quad H(S|U=u) - H(S+Z|U=u) \leq 0
\end{aligned} \tag{8.8}$$

where we have considered that the noise Z, independent of S and U, cannot decrease the conditioned entropy. Moreover, with similar considerations,

$$\begin{aligned}
H(S_d|Y,U=u) &= H(Y+X+Z+S'|Y,U=u) \\
&= H(Z+S'|Y,U=u) = H(Z+S') = h(p*q).
\end{aligned} \tag{8.9}$$

Considering the terms in the second sum, if $u \in \mathcal{U}_2$ the sum $x+s = f(u,s)+s$ is uniquely determined by the value of u. Thus, for $u \in \mathcal{U}_2$, it can be shown that (see [14]),

$$H(S|U=u) - H(Y|U=u) = h(d_u) - h(p). \tag{8.10}$$

And also

$$\begin{aligned} H(S_d|Y,U=u) &= H(S+X+S_d|Y,U=u) \\ &= H(X+S'|Y,U=u) = h(d_u * q). \end{aligned} \tag{8.11}$$

Combining (8.6), (8.7), (8.8), (8.9), (8.10) and (8.11) we obtain:

$$\begin{aligned} &I(U;Y,S_d) - I(U;S) \\ &\quad \leq h(d*p*q) + (1-\theta)\sum_{u\in\mathcal{U}_1} \lambda_{u,1} h(p*q) \\ &\qquad + \theta \sum_{u\in\mathcal{U}_2} \lambda_{u,2}\left[h(d_u) - h(d_u*q) - h(p)\right] \\ &\quad = h(d*p*q) + (1-\theta)h(p*q) + \theta\sum_{u\in\mathcal{U}_2}\lambda_{u,2}\left[h(d_u) - h(d_u*q) - h(p)\right]. \end{aligned} \tag{8.12}$$

Defining $G(x) = h(x) - h(x*q) - h(p)$ we have:

$$\frac{d^2}{dx^2}G(x) = -\frac{q(1-q)\log_2 e}{x(1-x)(x*q)(1-x*q)} < 0 \tag{8.13}$$

for $x, q \in (0, 1/2)$. So $G(x)$ is concave in the interval of interest and we obtain

$$\sum_{u\in\mathcal{U}_2}\lambda_{u,2}G(d_u) \leq G\left(\sum_{u\in\mathcal{U}_2}\lambda_{u,2}d_u\right) = G(d_2). \tag{8.14}$$

Substituting this result in (8.12) we have that, for each $p(x,u|s)$, we can find $d_1, d_2 \in [0,1/2]$ and $\theta \in [0,1]$ satisfying $d \geq (1-\theta)d_1 + \theta d_2$ such that

$$\begin{aligned} &I(U;Y,S_d) - I(U;S) \\ &\quad \leq h\left([(1-\theta)d_1 + \theta d_2] * p * q\right) \\ &\quad - (1-\theta)h(p*q) + \theta[h(d_2) - h(d_2*q) - h(p)] \\ &\quad \leq h\left([(1-\theta)d_1' + \theta d_2'] * p * q\right) \\ &\quad - (1-\theta)h(p*q) + \theta[h(d_2') - h(d_2'*q) - h(p)] \end{aligned}$$

where $d_1' \in [d_1, 1/2]$, $d_2' \in [d_2, 1/2]$ such that $d = (1-\theta)d_1' + \theta d_2'$. The second inequality follows from the observation that the RHS of the first inequality is strictly increasing in d_1 and d_2 for each fixed θ.

Now we provide a distribution for U and a mapping for X capable of achieving the bound above. Let's consider $d_1, d_2 \in [0, 1/2]$, $\theta \in [0, 1]$ and $\mathcal{U} = \{u_0, u_1, u_{id}, u_{not}\}$, where

$$\begin{aligned}
p(U = u_0|S = 0) &= p(U = u_0|S = 1) = (1 - \theta)(1 - d_1)\\
p(U = u_1|S = 0) &= p(U = u_1|S = 1) = (1 - \theta)\ d_1\\
p(U = u_{id}|S = 0) &= p(U = u_{not}|S = 1) = \theta(1 - d_2)\\
p(U = u_{not}|S = 0) &= p(U = u_{id}|S = 1) = \theta d_2
\end{aligned}$$

and the mapping $x = f(u, s)$:

$$\begin{aligned}
f(u_0, 0) = f(u_0, 1) = f(u_{id}, 0) = f(u_{not}, 1) = 0\\
f(u_1, 0) = f(u_1, 1) = f(u_{id}, 1) = f(u_{not}, 0) = 1.
\end{aligned}$$

With these assumptions, for θ, d_1 and d_2 satisfying

$$d = E\{d_H(X, 0)\} = (1 - \theta)d_1 + \theta d_2 \tag{8.15}$$

it is easy to verify that

$$\begin{aligned}
I(U; Y, S_d) - I(U; S) =& h(d * p * q) - (1 - \theta)h(p * q)\\
&+ \theta[h(d_2) - h(d_2 * q) - h(p)].
\end{aligned} \tag{8.16}$$

□

In Fig. 8.2, the capacity increment due to NoSI at the decoder has been graphically reported. The depicted curves refer to the case $p = 0.1$.

8.3 DTC on Binary channel NoSI at the decoder

Similarly, when the SI are causally available at the encoder, it is possible to take advantage of some NoSI available at the decoder. In particular, the capacity region is increased with respect to the classical DTC case.

Proposition 2. The capacity of the binary channel described in the previous section with causal SI (DTC) and subject to the constraint (6.40) is given by

$$C^{dtc}_{b,NoSI}(d) = h(d * p * q) - h(p * q) + 2d\ [h(p * q) - h(p)]. \tag{8.17}$$

Again, note that if $q = 1/2$, (8.17) turns into (6.45) and if $q = 0$ (8.17) becomes (6.41).

Proof.

The coding over strategies technique suggested by Shannon in [15] is extremely useful here, since in the binary case the possible strategies $X = U(S)$ belong to the finite set $\mathcal{U} = \{u_0, u_1, u_{id}, u_{not}\}$ (where $u_0(S) = 0$, $u_1(S) = 1$, $u_{id}(S) = S$ and $u_{not}(S) = 1 + S$). The average weight of X is given by

$$\tilde{d} = E\{d_H(X, 0)\} = \Pr\{u_1\} + \frac{1}{2}\Pr\{u_{id}\} + \frac{1}{2}\Pr\{u_{not}\}.$$

Posing $\Pr\{u_0\} = \alpha$, $\Pr\{u_1\} = \beta$ and $\Pr\{u_{id}\} = \Pr\{u_{not}\} = \theta/2$, the constraint (8.5) becomes $d \geq \tilde{d} = \beta + \theta/2$. Note that the assumption $\Pr\{u_{id}\} = \Pr\{u_{not}\}$ is justified from the fact that the two symbols are cost-equivalent and the channel with the input $U \in \{u_{id}, u_{not}\}$ is perfectly symmetric. So a maximizing distribution for U must have the two symbols equiprobable.

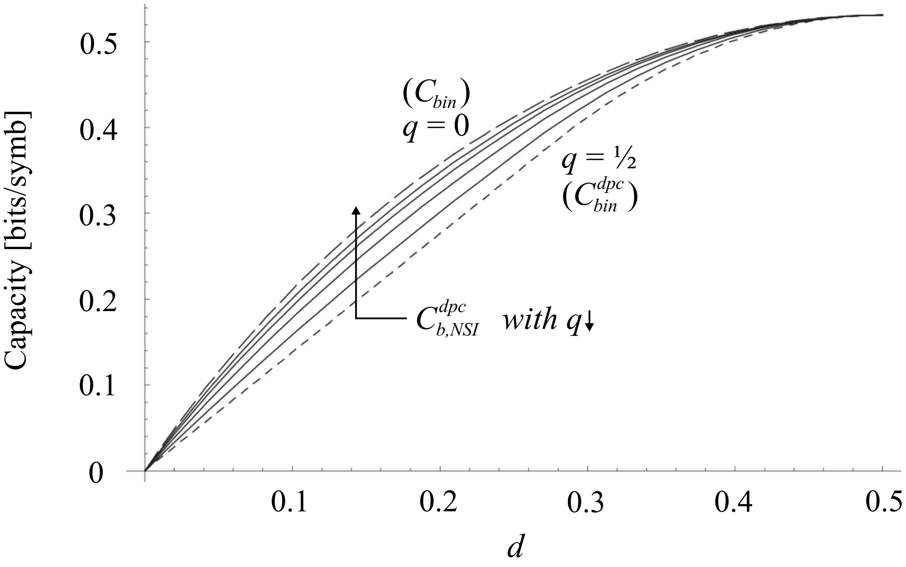


Figure 8.2: DPC capacity curves (p=0.1) corresponding to different values of q.

	u			
(y, s_d)	u_0	u_1	u_{id}	u_{not}
(0,0)	$\frac{1}{2}(1 - p * q)$	$\frac{1}{2}p * q$	$\frac{1}{2}(1 - p)$	$\frac{1}{2}p$
(0,1)	$\frac{1}{2}p * q$	$\frac{1}{2}(1 - p * q)$	$\frac{1}{2}(1 - p)$	$\frac{1}{2}p$
(1,0)	$\frac{1}{2}p * q$	$\frac{1}{2}(1 - p * q)$	$\frac{1}{2}p$	$\frac{1}{2}(1 - p)$
(1,1)	$\frac{1}{2}(1 - p * q)$	$\frac{1}{2}p * q$	$\frac{1}{2}p$	$\frac{1}{2}(1 - p)$

Table 8.1: Conditioned joint PMF $p(y, s_d|u)$ in DTC with NoSI.

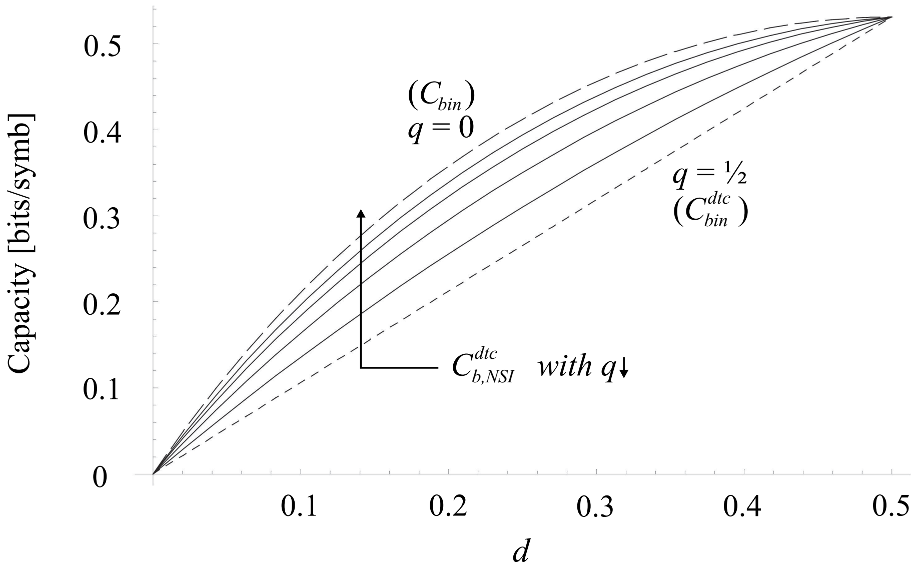


Figure 8.3: DTC capacity curves (p=0.1) corresponding to different values of q.

Computing $I(U;Y,S_d) = \sum_u p(u) \sum_{y,s_d} p(y,s_d|u) \log_2 \frac{p(y,s_d|u)}{p(y,s_d)}$ with the values reported in Tab. 8.1 and exploiting the normalization $\alpha + \beta + \theta = 1$, we have

$$I(U;Y,S_d) = h(\tilde{d} * p * q) - h(p * q) + \theta[h(p * q) - h(p)].$$

Thus, the maximum $I(U;Y,S_d)$ is attained in correspondence of the maximum values allowed for $\tilde{d}$ and θ, i.e. $\tilde{d} = d$ and $\theta = 2d$, which determines also $\beta = 0$ and $\alpha = 1 - 2d$. And this concludes the proof.

□

In Fig. 8.3 we have reported some capacity curves obtained with and without the described NoSI at the decoder, in the binary DTC scenario. In particular, the examples depicted refer to the case $p = 0.1$.

8.4 Gaussian channel with SI at the encoder and NoSI at the decoder

Now we consider the Gaussian version of the memoryless channel in Fig. 8.1, where S, S_d and Z are Gaussian r.v.s. The noise $Z \sim \mathcal{N}(0, \sigma_Z^2)$ and the interference known at the encoder $S \sim \mathcal{N}(0, \sigma_S^2)$ are statistically independent. Similarly to the binary case, let's consider the NoSI known at the decoder defined as $S_d = k(S + S')$, where $S' \sim \mathcal{N}(0, \sigma_d^2)$ is independent of S and k is a real constant. Moreover the transmission is subject to the power constraint $E\{X^2\} \leq P$. The capacity of this channel in the DPC case has been implicitly given in [13], since for the optimal (U, X) the r.v.s $U \to Y \to S$ form a Markov chain. Thus NoSI at the decoder are useless. More interesting is the Gaussian DTC. As already stated, the capacity of this channel when S is causally available at the encoder is not known. The inflated lattice precoding scheme [11] offers a good practical solution, independent of the interference distribution and optimal for asymptotically high SNR. However, if the interference statistics is fixed and known, the possibility to achieve better performance exploiting this knowledge has been observed in different works. In particular, in [17] and [18] it has been shown how it is possible to obtain higher capacity-per-unit cost [19] (and thus lower values for the minimum achievable E_b/N_0) considering sets of strategies $u(\cdot)$ including a *zero-cost* function. It is easy to see that with a proper choice for the signalling strategies, some NoSI at the decoder can be useful to further extend the achievable region. Let us consider, for instance, the "on-off" signalling described in [17] (but similar considerations may hold for [18])

$$u(s) = \begin{cases} u_0(s) = 0 & \text{if "off" input} \\ u_1(s) = as & \text{if "on" input} \end{cases} \tag{8.18}$$

where a is a real parameter and the duty cycle is properly reduced if the interference power increases. For asymptotically high noise-to-interference ratios (NIRs) the minimum achievable E_b/N_0, denoted with $(E_b/N_0)_{min}$, approaches -1.59 dB in the "on-off" case, while it is bounded to ≈ 2.4 dB with lattice precoding. However, a real improvement can be observed only for high NIRs, while in general the specific "on-off" signalling is not robust to the interference power. In this situation, some NoSI at the decoder turn out to be useful. Thanks to the theorem by Verdú in [19], if the signalling alphabet has a zero-cost symbol u_0, the capacity-per-unit cost C_U can be computed as

$$C_U = \sup_{u \in \mathcal{U} \setminus u_0} \frac{D(p_{Y,S_d|U=u} || p_{Y,S_d|U=u_0})}{E[u(S)^2]} \tag{8.19}$$

where $\mathcal{U}$ is the set of all the possible strategies of the form $u : R \to R$ and $D(\cdot||\cdot)$ is the Kullback-Leibler divergence. Restricting $\mathcal{U}$ to the "on-off" functions in (8.18), we obtain a lower bound for C_U. In particular, defining the vector $\mathbf{R} = [Y, S_d]^T$ and representing with $\mathbf{r}$ its generic sample, the couple (Y, S_d) can be described in terms of multivariate normal distribution, according to the conditioned PDF

$$p_{\mathbf{R}|U=u_i}(\mathbf{r}|u_i) = \frac{1}{(2\pi\sqrt{|\mathbf{M_i}|})} e^{-(1/2)\mathbf{r}^T \cdot \mathbf{M_i}^{-1} \cdot \mathbf{r}}$$

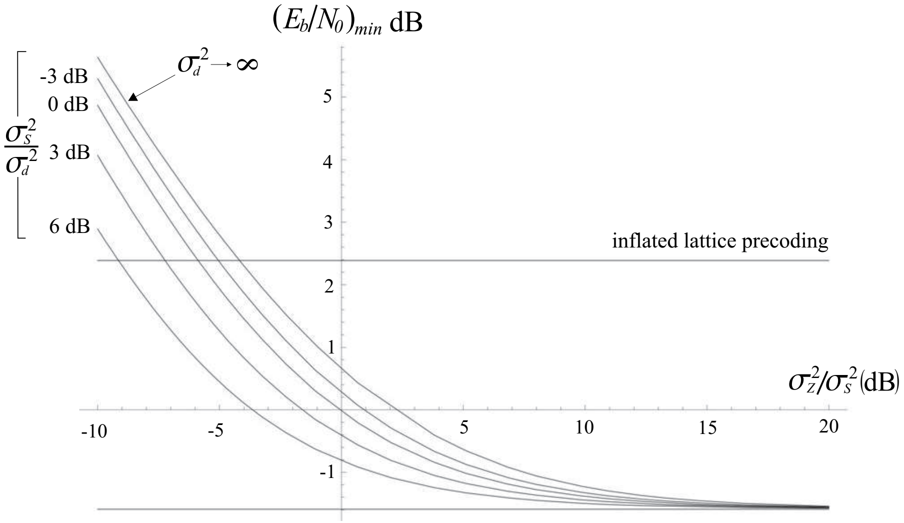


Figure 8.4: $(E_b/N_0)_{min}$ corresponding to "on-off" signalling with and without NoSI at the decoder.

where $i \in \{0, 1\}$, $| \cdot |$ indicates the matrix determinant and $\mathbf{M_i} = E\{\mathbf{R} \cdot \mathbf{R}^T | U = u_i\}$ are the covariance matrices. Defining $\gamma_1 = \sigma_Z^2/\sigma_S^2$ and $\gamma_2 = \sigma_S^2/\sigma_d^2$, after some computations we obtain

$$C_U \geq \sup_{a \geq -1} \frac{\log_2 e}{2a^2\sigma_S^2} \left[\frac{(1+a)^2 - 1 + a^2\gamma_2}{1 + \gamma_1 + \gamma_1\gamma_2} - \ln\left(1 + \frac{(1+a)^2 - 1}{1 + \gamma_1 + \gamma_1\gamma_2}\right)\right].$$

In Fig. 8.4, it is graphically reported the comparison between the minimum achievable E_b/N_0 with the "on-off" signalling (correspondent to the inverse of the respective capacity-per-unit cost) versus the NIR, for different values of γ_2.

Chapter 9

Conclusion

In the present dissertation we have presented the main results obtained within the research activity in these years. The work has been characterized by a logic continuity in terms of considered subjects and applied methodology.

The *Phoenix* project, which I could follow since its beginning, has provided many interesting results starting from the contributions of all the partners. In particular, I have worked on the development of the JSCC Controller models and algorithms, as well as on the MPEG-4 source co-decoding and on many parts of the radio link. My activity comprehended the development of software which has been included in the *Phoenix* common simulation chain. In particular, the interface definition required many coordination efforts with the project partners.

In general, the results of the project and the strategies I proposed appear very promising, as it has been extensively shown in this dissertation.

In 2006 I spent 6 months at the *Multimedia Communication Laboratory* at the *Texas A&M University*, Texas, USA, where I could study and obtain some results in the field of DPC/DTC with noisy side information available at the decoder. In particular I found the capacity regions of the proposed binary channels and showed the improvements provided by the additional side information. This topic is strictly connected to the problem of watermarking for video sequences.

This activity constituted the continuation of my research in the field of DPC: in particular, I proposed a low-complexity co-decoding technique capable of achieving performance within 1.2-1.3 dB from the correspondent channels without any interference.

The main results of my research activity have been published in several scientific papers and some of them are currently under revision.

Future works may continue with the application of the results obtained in the DPC and

DTC fields to the JSCC approach for video communications.

Bibliography

[1] M. Costa, "Writing on Dirty Paper (Corresp.)," *IEEE Trans. on Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[2] T.M. Cover, J.A. Thomas, "Elements of information theory," *John Wiley and Sons Ed.*, 1991

[3] R. Zamir, S. Shamai (Shitz) and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. on Information Theory*, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

[4] W. Yu, D. P. Varodayan and J. M. Cioffi, "Trellis and convolutional precoding for transmitter-based interference presubtraction," *IEEE Trans. on Commun.*, vol. 53, no. 7, pp. 1220–1230, Jul. 2005.

[5] U.Erez and S. Ten Brink, "A close-to-capacity dirty paper coding scheme," *IEEE Trans. on Information Theory*, vol. 51, no. 10, pp. 3417–3432, Oct. 2005.

[6] S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," *Probl. Pered. Inform. (Probl. Inform. Transm.)*, vol. 9, no. 1, pp. 19–31, 1980.

[7] R. F. H. Fischer, "Precoding and Signal Shaping for Digital Transmission," John Wiley and Sons, New York, 2002.

[8] J. H. Conway and N. J. A. Sloane, "Sphere packings and groups," Springer-Verlag, New York, 1988.

[9] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Trans. on Information Theory*, vol. 42, no. 4, pp. 1152–1159, Jul. 1996.

[10] W. E. Ryan, "Concatenated convolutional codes and iterative decoding," from the *Wiley Encyclopedia of Telecommunications*, John Wiley and Sons, 2003.

[11] U. Erez, S.Shamai and R. Zamir, "Capacity and Lattice Strategies for canceling known interference," *IEEE Trans. on Information Theory*, vol. 51, no. 11, pp. 3820–3833, Nov. 2005.

[12] S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," *Probl. Pered. Inform. (Probl. Inform. Transm.)*, vol. 9, no. 1, pp. 19–31, 1980.

[13] M. Costa, "Writing on Dirty Paper (Corresp.)," *IEEE Trans. on Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[14] R.J. Barron, B. Chen and G.W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. on Information Theory*, vol. 49, no. 5, pp. 1159–1180, May 2003.

[15] C.E. Shannon, "Channels with side information at the transmitter," *IBM Journal*, pp. 289–293, Oct. 1958.

[16] A. Bennatan, D. Burshtein, G. Caire and S. Shamai, "Superposition coding for side-information channels," *IEEE Trans. on Information Theory*, vol. 52, no. 5, pp. 1872–1889, May 2006.

[17] G. Caire, S. Shamai, "Writing on dirty tape with LDPC codes," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 62, pp. 123–140, Nov. 2003

[18] D. Hoesli, "On the capacity per unit cost of the dirty tape channel," in Winter School on Coding and Information Theory, Monte Verita, Switzerland, Feb. 2003.

[19] S. Verdú, "On the capacity per unit cost," *IEEE Trans. on Information Theory*, vol. 36, no. 5, pp. 1019–1030, Sep. 1990.

[20] G.D. Forney Jr., "On the role of MMSE estimation in approaching the information-theoretic limits of linear Gaussian channels: Shannon meets Wiener," *CoRR*,cs.IT/0409053, 2004

[21] IEEE802.11, part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band," P802.11a/D7.0, July 1999.

[22] ETSI TS 101 475, "Broadband Radio Access Networks (BRAN) Hiperlan Type2 - Physical layer," V 1.1.1 April 2000.

[23] T. Keller and L. Hanzo, "Adaptive multicarrier modulation: a convenient framework for time-frequency processing in wireless communications," IEEE Proc. of the IEEE, vol. 88, no. 5, May 2000, pp. 611-640.

[24] J.A.C.Bingham, *ADSL, VDSL, and Multicarrier Modulation*, John Wiley and Sons, 2000.

[25] P.S. Chow, J. M. Cioffi, J.A.C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," IEEE Transactions on Communications, vol. 43, n. 2/3/4, pp. 773-775, Feb/Mar/Apr 1995.

[26] J. Campello, "Optimal discrete bit loading for multicarrier modulation systems," ISIT 1998, Cambridge, MA, USA, pag. 193, Aug. 1998.

[27] D. Dardari, M. G. Martini, M. Milantoni and M. Chiani, "MPEG-4 video transmission in the 5GHz band through an adaptive OFDM wireless scheme" Proceedings of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC 2002, vol. 4, Sept. 2002, Lisboa, Portugal, pp. 1680-1684.

[28] M.Budagavi, W. Rabiner Heinzelman, J. Webb, R. Talluri, "Wireless MPEG-4 Video Communication on DSP Chips," IEEE Signal Processing Magazine, January 2000.

[29] M.G. Martini, M. Chiani, "Rate distorsion models for Unequal Error Protection for wireless videotransmission," IEEE VTC 2004 Conference, Milan, Italy, May 2004.

[30] M.G.Martini, M.Chiani, "Wireless transmission of MPEG-4 video: performance evaluation of Unequal Error Protection over a block fading channel," IEEE Veh. Technol. Conf. (VTC 2001), Rhodes, May 2001.

[31] H. Zheng and K.J.R Liu, "Robust Image and Video Transmission over Spectrally Shaped Channels Using Multicarrier Modulation," IEEE Trans. on Multimedia, Vol. 1, No.1, pp. 88-103, March 1999.

[32] S.B.Weinstein, P.M. Ebert, "Data transmission by frequency division multiplexing using the discrete Fourier transform," IEEE Transaction on Communications, vol.19, Oct. 1971, pp. 628-634.

[33] D. Dardari, V. Tralli, “High-speed indoor wireless communications at 60 GHz with coded OFDM,” IEEE Transactions on Communications, vol.47, no.11, pp. 1709-1721, Nov. 99.

[34] J.G. Proakis, *Digital communications*, McGraw Hill, 4th edition, 2001.

[35] C. Bergeron, C. Lamy-Bergot, “Modelling H.264/AVC sensitivity for error protection in wireless transmission,” IEEE MMSP Conference, 2006.

[36] R. G. Gallager, “Low-Density Parity-Check Codes,” Cambridge, MA: M.I.T. Press, 1963.

[37] H. Jin, A. Khandekar, and R. McEliece, “Irregular repeat-accumulate codes,” in Proc. International Symposium on Turbo codes and Related Topics, Sept. 2000, pp. 1-8.

[38] D. Divsalar, H. Jin, R.J. McEliece, “Coding theorems for turbo-like codes,” Proc. 36th Allerton Conference on Communication, Control and Computing, Allerton, Illinois, pp. 201-210, September 1998.

[39] M. Yang, Y. Li, and W. Ryan, “Design of effciently encodable moderate-length high-rate irregular LDPC codes,” IEEE Trans. Commun., vol. 52, pp. 564-571, Apr. 2004.

[40] M. Chiani, A. Ventura, “Some implementation issues and error floor analysis for high-rate low-density parity-check codes,” in Proc. IEEE SoftCOM 2001, Oct. 2001.

[41] T. Richardson, M.A. Shokrollahi, R. Urbanke, “Design of capacity approaching irregular low-density parity-check codes,” IEEE Trans. Inform. Theory, vol.47, no.2, pp.619-637, February 2001.

[42] E. Paolini, G. Liva, M. Chiani, “An overview of some efficient encoding and decoding algorithms for Low-Density Parity-Check Codes,” IEEE SoftCom 2003, October 2003.

[43] E. Paolini, G. Liva, M. Chiani, “Simple reconfigurable low-density-parity-check codes,” IEEE Comm. Letters, vol. 9, no. 3, pp. 258-260, March 2005.

[44] R. Grunheid, E. Bolinth, H. Rohling, “A blockwise loading algorithm for the adaptive modulation technique in OFDM systems,” IEEE Vehicular Technology Conference, VTC 2001 Fall, vol. 2, 7-11 Oct. 2001, pp. 948-951.

[45] W. Yu, J.M. Cioffi, “On constant power water-filling,” IEEE Proc. ICC 2001, pp. 1665-1669.

[46] L. M. C. Hoo, J. Tellado, J. Cioffi, “Discrete dual QoS loading algorithms for multicarrier systems” IEEE International Conference on Communications, ICC ’99, vol. 2, 6-10 June 1999, pp.796-800.

[47] D.Dardari, “A Low Complexity and Low Overhead Adaptive Bit Loading Algorithm for OFDM Based High-speed WLAN Systems,” Cost meeting report, COST 273 TD(03)155 Prague, Czech Republic September 24-26, 2003.

[48] MPEG-4 standard, MPEG-4 Video Group, “Final Draft of International Standard,” ISO-IEC/JTC1/SC29/WG11 N2502a, Atlantic City, October 1998.

[49] MPEG-4 Video Group, *MPEG-4 Overview-(V.15-Beijing Version)* ISO/IEC JTC1/SC29/WG11 N2323, Beijing, 2000, http://cselt.stet.it/mpeg/standard/mpeg-4/mpeg-4.html

[50] Y. Q. Shi, H. Sun, “Image and Video Compression for Multimedia Engineering,” *CRC Press*.

[51] Y. Wang, S. Wenger, J. Wen, A. K. Katsaggelos, “Error Resilient Video Coding Techniques,” *IEEE Signal Processing Magazine*, pp. 61-81, July 2000.

[52] Z. Wang, L. Lu, A.C. Bovik, “Video quality assesment based on structural distortion measurement,” Signal processing: image communication, vol. 29, no. 1, January 2004.

[53] Z. Wang, H.R. Sheikh, A.C. Bovik, “No-reference perceptual quality assessment of JPEG compressed images,” Proc. IEEE international conference on image processing, pp. 477-480, Rochester, New York, September 2002.

[54] C. Lamy-Bergot, N. Chautru, C. Bergeron, “Unequal error protection for H.263+bitstream over a wireless IP network,” IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), Toulouse, France, May 2006.

[55] S. Vembu, S. Verd, “The Source-Channel Separation Theorem Revisited,” *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 44-54, Jan. 1995.

[56] M. G. Martini, “Wireless Multimedia System: Joint Source and Channel Coding for Video Transmission,” PhD dissertation, Univ. of Bologna, January 2002.

[57] M. G. Martini, M. Chiani, “Proportional Unequal Error Protection for MPEG-4 video transmission,” *IEEE International Conference on Communications ’01*, vol. 4, pp. 1033-1037, 2001

[58] M. G. Martini, M. Chiani, “Robust Transmission of MPEG-4 video: Start Codes Substitution and Length Field Insertion Assisted Unequal Error Protection,” *Proceedings PCS-2001*, pp. 155-158, 2001

[59] MoMuSys project website:
http://www.cordis.lu/infowin/acts/rus/projects/ac098.htm

[60] M.G.Martini, M.Chiani, “Joint Source-Channel Error Detection with Standard Compatibility for Wireless Video Transmission,” Proc. IEEE WCNC 2002, Orlando, Florida, March 2002.

[61] J.Medbo, H.Andersson, P.Schramm, “Channel models for Hiperlan/2 in different indoor scenarios,” COST 259, April 1998.

[62] “White paper for Phoenix,” Phoenix Consortium, project proposal for IST FP6, 2003.

[63] “Deliverable D1.2 - Phoenix requirements document,” Phoenix Consortium, March 2004.

[64] “Deliverable D3.3a - Specification and preliminary design of network transparency solutions,” Phoenix Consortium, September 2004.

[65] “Deliverable D1.3 - Phoenix pecifications,” Phoenix Consortium, June 2004.

[66] “Deliverable D2.4a - Preliminary JSCC/D controller model,” Phoenix Consortium, September 2004.

[67] “Deliverable D2.4c - JSCC controller: final design and algorithmic optimisation,” Phoenix Consortium, June 2006.

[68] “Deliverable D4.3 - Test-bed and demos,” Phoenix Consortium, October 2006.

[69] “Soft-input decoding of variable-length codes applied to the H.264 standard,” C. Bergeron, C. Lamy-Bergot, IEEE MMSP, 2004.

[70] J. Hagenauer, N. Seshadri, C. E. W. Sundberg, “The Performance of Rate-Compatible Punctured Convolutional Codes for Digital Mobile Radio,” *IEEE Transactions on Communications*, VOL. 38, NO. 7, pp. 966-980, July 1990

[71] L. Diana, J. M. Kahn, “Rate-Adaptive Modulation Techniques for Infrared Wireless Communications,” *IEEE International Conference on Communications ’99*, vol. 1, pp. 597-603, 1999

[72] S.A. Garawi, R.S.H. Istepanian, M.A.Abu-Rgheff, “3G wireless communication for mobile robotic tele-ultrasonography sistems,” IEEE Comm. Mag., vol. 44, no. 4, pp. 91-96, April 2006.

[73] P.C. Cosman et al., “Thoracic CT images: effect of lossy image compression on diagnostic accuracy,” Radiology, 190:517-524, 1994.

[74] C.E. Shannon, “A mathematical theory of communication,” Bell System Technical Journal, vol. 27, pp. 379-423, 623-656, July-Oct. 1948.

[75] J.L. Massey, “Joint source and channel coding,” in Commun. Systems and Random Process Theory, NATO Advanced Studies Institutes Series E25, J.K. Skwirzynski editor, pp. 279-293, Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[76] S.B. Zahir Azami, P. Duhamel and O. Rioul, “Joint source-channel coding: panorama of methods,” in Proceedings of CNES workshop on data compression, Toulouse, France, Nov. 1996.

[77] J. Hagenauer and T. Stockhammer, “Channel coding and transmission aspects for wireless multimedia,” Proceedings of the IEEE, vol. 87, no. 10, Oct. 1999.

[78] K. Sayood and J.C. Borkenhagen, “Use of residual redundancy in the design or joint source/channel coders,” IEEE Transactions on Communications, vol. 39, pp. 838-846, June 1991.

[79] M. Park and D.J. Miller, “Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation,” IEEE Transactions on Communications, vol. 48, no. 1, pp. 1-6, Jan. 2000.

[80] L. Perros-Meilhac and C. Lamy, “Huffman tree based metric derivation for a low-complexity sequential soft VLC decoding,” in Proceedings of IEEE ICC’02, New York, USA, vol. 2, pp. 783-787, April-May 2002.

[81] M. Mazzotti, M.G. Martini, M. Chiani, “Fixed-Packet-Length Transcoding for Error Resilient Video Transmission over WCDMA Radio,” Proc. of Packet Video 2003, Nantes, April 2003.

[82] D. Dardari, M.G. Martini, M. Mazzotti, M. Chiani, “Layered Video Transmission on Adaptive OFDM Wireless Systems,” Eurasip Journal on Applied Signal Processing, January 2004.

[83] D. Dardari, M.G. Martini, M. Mazzotti, M. Chiani, “Bit-loading for Unequal Error Protection of Video Streams in OFDM Wireless Systems,” IEEE International Conference on Communications (ICC04), Paris, June 2004.

[84] M.G. Martini, M. Mazzotti, M. Chiani, G. Panza, C. Lamy-Bergot, J. Huusko, G. Jeney, G. Feher, S. X. Ng, “Controlling Joint Optimization of Wireless Video Transmission: the PHOENIX Basic Demonstration Platform,” 14th IST Mobile & Wireless Communication Summit, Dresden, 19-23 June 2005.

[85] M.G. Martini, M. Mazzotti, M. Chiani, “Quality driven control for joint optimization of wireless video transmission,” VLBV Workshop, Cagliari (I), September 2005.

[86] M. Mazzotti, M. Chiani, “A simple rate-1/2 co-decoding scheme for Writing on Dirty Paper,” IEEE International Conference on Communications (ICC06), Istanbul, June 2006.

[87] M. Mazzotti, Z. Xiong, “Effects of noisy side information at the decoder in dirty paper and dirty tape coding,” IEEE Information Theory Workchop (ITW06), Chengdu, China, October 2006.

[88] M.G. Martini, M. Mazzotti, “Quality driven wireless video transmission for medical applications,” IEEE International Conference of the Engineering in Medicine and Biology Society (EMBC06), New York, USA.

[89] M.G. Martini, M. Mazzotti, C. Lamy-Bergot, J. Huusko, P. Amon, “Content adaptive network aware joint optimization of wireless video transmission” accepted for pubblication on IEEE Communication Magazine.

[90] C. Lamy-Bergot, C. Bergeron, P. Hammes, M.G. Martini, P. Amon, J. Vehkapera, S.X. Ng, L. Bokor, G. Feher, M. Mazzotti, G. Panza, "Cross-layer optimisation for video transmission over an IP wireless link," proposed for pubblication on IEEE Journal on Selected Topics in Signal Processing.

[91] "Robust multi-layer controller structures for enhanced medical video streaming in robotic teleultraonography applications (ROAM)," M.G. Martini, R. Istepanian, M. Mazzotti, E. Tan, N. Philips, proposed for pubblication on IEEE Transanctions of Mobile Computing.