國立交通大學

電信工程研究所

博士論文

基於多重敘述編碼理論之無線通訊系統的品質管理研究

QoS Control for Multi-Stream Voice over Mobile IP Networks

研 究 生：吳俊鋒

指導教授：張文輝 博士

中 華 民 國 一百 年 六 月

基於多重敘述編碼理論之無線通訊系統的品質管理研究

QoS Control for Multi-Stream Voice over Mobile IP Networks

研究生：吳俊鋒　　Student: Chun-Feng Wu
指導教授：張文輝　博士　　Advisor: Dr. Wen-Whei Chang

國立交通大學
電信工程研究所
博士論文

A Dissertation
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
Hsinchu, Taiwan

2011 年 6 月

基於多重敘述編碼理論之無線通訊系統的品質管理研究

學生：吳俊鋒　　　　　　指導教授：張文輝 博士

國立交通大學電信工程研究所

摘要

無線通訊的服務品質取決於諸多因素，包括封包漏失、延遲時間、背景雜訊、及語音編碼失真。本篇論文旨在探討多重敘述編碼理論之無線通訊系統的品質管理研究，主要採用多重敘述傳輸系統，一方面利用路徑分集以增加傳輸系統的強健性，另一方面可利用不同敘述間的相關性設計其錯誤隱匿機制。有關傳輸位元錯誤的隱匿機制，前人研究基於強健性能與快速實現的整體考量，根據渦旋碼理論而發展疊代訊源通道解碼演算法，關鍵元件包括軟性輸出通道解碼器和軟性位元訊源解碼器。問題是一般採用的位元層級通道解碼演算法有其限制，不僅無法將相鄰索引之間的相關特性有效整合於訊源事前訊息，與基於索引層級而推導的訊源解碼演算法也存在著相容性的問題。針對這些議題，本論文研究將鎖定索引層級的疊代訊源通道解碼機制。首先開發一個索引層級的 BCJR 通道解碼演算法，可有效整合訊源的事前訊息於其軟性輸出的解碼過程。並且進一步配合多重敘述所屬相關訊息的交叉運用，準確估算不同傳輸索引值的後驗機率，並依最小均方誤差準則求得多重敘述向量量化的最佳解碼輸出。另一個重要的議題則是接收端播放緩衝器的設計。系統設計應整體考量不同關鍵元件的最佳組合，且因應隨時變化的網路傳輸特性作合理調整。首先，我們延伸國際電信聯盟 ITU 針對單一路徑傳輸系統所制訂的 E-model，進一步開發新的音質評量效能指標，可廣泛應用在多重敘述傳輸的系統規劃。有別於前人研究是將播放緩衝器與前向錯誤控制分開設計，本研究基於音質最佳化的設計理念提出一個適應性整合控制演算法。根

據新的音質評量指標，多重敘述傳輸系統的設計規劃成為一個音質損害最小化問題，依據網路動態彈性調整前向錯誤控制與播放排程，進而達到延遲與封包漏失的最佳平衡點。

QoS Control for Multi-Stream Voice over Mobile IP Networks

Student: Chun-Feng Wu Advisor: Dr. Wen-Whei Chang

Institute of Communications Engineering, National Chiao Tung University
Hsinchu, Taiwan, Republic of China

ABSTRACT

Packet loss and network delay are two essential problems to real-time voice communication over mobile IP Networks. The purpose of this dissertation is to develop a multi-stream voice communication system with its quality of service (QoS) control for increased channel robustness. The first part will focus on the error concealment of packet-erasure as well as channel bit errors. The basic strategy is a multiple description scalar quantization (MDSQ) system, in which multiple correlated indexes of the source are assigned and transmitted over channels to take advantage of largely uncorrelated loss and delay characteristics. We propose the use of turbo principle to develop a symbol-based iterative source-channel decoding algorithm for better decoding of multiple descriptions over a noisy channel. We first modify the BCJR algorithm based on sectionalization trellis so that symbol *a posteriori* probabilities can be derived and used as the extrinsic information to improve the iterative decoding between the source and channel decoders. The residual source redundancies are exploited as a priori information and a joint source decoding is formulated in the form of a maximum a posteriori estimation problem. We also formulate a recursive implementation for the source decoder that processes reliability information received on different channels and combines them with inter-description correlation to estimate the transmitted quantizer indexes. Another important issue to address is the playout buffer design which is used at the receiver to smooth out the jitter. As a further step toward perceptual optimization, the error concealing capabilities of multiple description coding can be improved by including an forward error control (FEC) mechanism. We present an objective method for multi-stream voice quality prediction model. Based on the new prediction model, we proposed the use of minimum overall impairment as a perceptually motivated optimization criterion for joint playout buffer and FEC control. Joint playout and FEC adjustment is then formulated as an optimization problem leading to a better balance between end-to-end delay and packet loss.

誌謝

博士班研讀的過程中，首先要感謝我的指導老師張文輝教授悉心的指導，使我在研究上可以很有效的掌握研究技巧與重點，故期許自己對於電信相關領域能有多一分的貢獻。另一方面也感謝老師在生活上也給予我很多的建議與鼓勵。此外，個人非常感謝交通大學所提供優良的學術環境，讓學生在科技與人文的素養均能有所成長。再者，特別感謝新竹教育大學江源泉教授對於研究所提供的協助與指導。在語音通訊實驗室裡的研究生活中，有學長李承龍與同學傅泰魁、曹正宏、林宜德、許忠安、蔡知鑑、何依信、顏廣儀、曾啟翔、潘彥璋、張永樂、陳亞民、戴玲玲、葉葉誠、吳鴻材等相互提攜與砥礪。除此，還要對強大隊整體成員致上敬意，他們在假日時與我一起吃喝玩樂，使我博士班的生活多彩多姿。除此，也要感謝顏雅慧在一路相伴。最後我要特別地感謝我的父母，阿姨還有弟弟，他們是我物質上與心靈上的能量來源。由此，本人才得以順利完成博士論文。

Contents

Abstract (in Chinese) **i**

Abstract (in English) **iii**

Acknowledgement **v**

Contents **vi**

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 Multiple Description Coding 2

1.2 Joint Playout and FEC Control 4

1.3 Iterative Source-Channel Decoding 7

1.4 Dissertation Organization 9

2 Multi-Stream Transmission System and Quality Prediction Model **10**

2.1 Multi-Stream Voice Transmission over a Packet-Erasure Channel 10

2.2 Multi-Stream Voice Quality Prediction Model 13

2.3 Experimental Results 16

3 QoS Control for Multi-Stream Voice over IP Networks **20**

3.1 Perceptual-Based Playout Mechanism 20

3.1.1 Adaptive Playout Scheduling Algorithm 21
3.1.2 Perceptually Motivated Optimization Criterion 22
3.1.3 Perceptual Optimization of Playout Buffer 24
3.1.4 Experimental Results 25
3.2 Jont FEC and Playout Control Mechanisms 27
3.2.1 FEC in a Gilbert-Model Loss Process 28
3.2.2 Joint FEC and Playout Control 32
3.2.3 Experimental Results 34

4 Iterative Symbol Decoding of Convolutionally Encoded Multiple Descriptions 38
4.1 Introduction 39
4.2 Multi-Stream Voice Transmission over a Noisy Channel 41
4.3 Symbol Decoding of Binary Convolutional Codes 43
4.4 The MD-SISO Source Decoder 48
4.5 Experimental Results 53

5 Conclusions and Future Work 57
5.1 Conclusions 57
5.2 Future Work 59
5.2.1 MD-VLC Design 60
5.2.2 Iterative Decoding of Convolutionally-Encoded MD-VLC 60
5.3 SISO Source Decoding of Variable-Length Codes 61
5.4 SISO Channel Decoding of Convolutionally-Encoded VLC 67

Bibliography 74

A 81

B 85

Vita 87

Publication List 88

1896

List of Tables

2.1 I_e comparison for different prediction models. 18

3.1 MOS comparison for different playout algorithms. 27

3.2 Basic Notation. 30

3.3 Average redundant bits comparison for different link loss rates. 37

List of Figures

1.1 Block diagram of MD voice transmission system. 2

1.2 Block diagram of multiple description coding system. 4

2.1 A two-channel VoIP simulation system. 12

2.2 A multi-hop transmission model for network simulations. 12

2.3 Schematic diagram for prediction of I_e model. 15

2.4 $I_{e,k}$ vs. packet erasure rate e. 17

3.1 Performance comparison for different playout algorithms. 26

3.2 A multi-description voice transmission system. 28

3.3 Performance comparison for different playout algorithms. 35

4.1 Block diagram of a two-channel MD communication system. 42

4.2 MD-ISCD scheme for the concatenation of MDSQ and convolutional codes. 42

4.3 Bit-level and merged trellis diagrams 45

4.4 MD-ISCD3 performance for Gauss-Markov sources with $\rho = 0.95$ and $(M, R) = (5, 3)$. 54

4.5 SNR performance of different decoders for $(M, R) = (4, 3)$ and Gauss-Markov sources ($\rho = 0.8, 0.95$) 55

4.6 SNR performance of different decoders for $(M, R) = (5, 3)$ and Gauss-Markov sources ($\rho = 0.8, 0.95$) 56

5.1 Model of the transmission system. 62

5.2 VLC trellis representation for $T = 4$, $N = 10$ and $C = \{c(0) = 11, c(1) = 00, c(2) = 101, c(3) = 010\}$. 64

5.3 Bit-level and merged trellis for $C = \{c(0) = 11, c(1) = 00, c(2) = 101, c(3) = 010\}$. 69

5.4 Three-dimension Trellis digram for $\nu = 2$, $N = 5$ and $T = 2$. 70

Chapter 1

Introduction

Quality of Service (QoS) has been one of the major concerns in the context of real-time multimedia communication over unreliable IP networks. Interactive real-time applications such as telephony and audio/video conferencing require high constraints on packet loss and end-to-end delay. When packet loss rates exceeds 10% and one-way delay exceeds 150 ms, the perceived conversational speech quality can be quite poor. There has been much interest in the use of packet-level forward error correction (FEC) [1] to mitigate the impact of packet losses. Most current FEC mechanisms send additional information along with the media stream so that the lost data can be recovered in part from the redundant information. In FEC schemes, however, loss recovery is performed at the cost of increased end-to-end delay. Multiple description (MD) coding [2]-[4] is another method to gain robustness by taking advantage of the largely uncorrelated loss and delay characteristics on different network paths. In MD coding, multiple descriptions of the speech are created in such a way that each description can be individually decoded for a reduced quality reconstruction, but if all descriptions are available, they can be jointly decoded for a better quality reconstruction. With multiple voice streams, the network delay experienced may vary with each packet depending on the paths taken by different streams and on the level of congestion along the path. The variation in network delay, referred to as jitter, must be smoothed out since it obstructs the proper

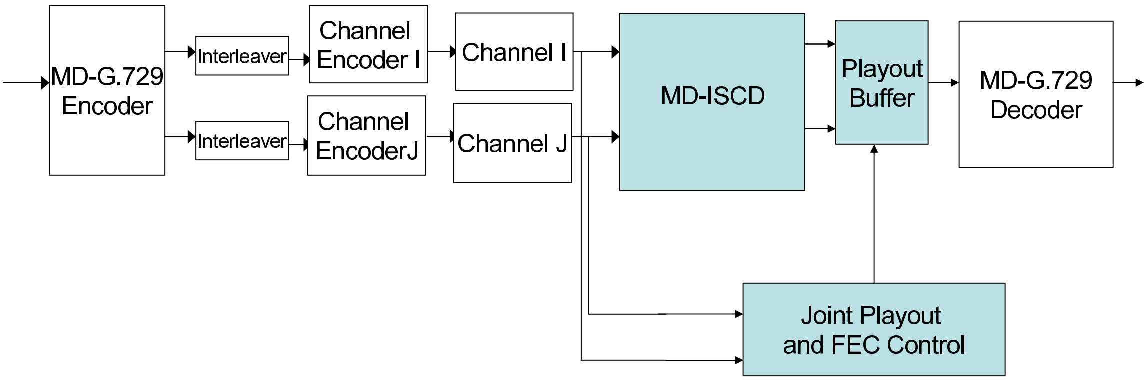

Figure 1.1: Block diagram of MD voice transmission system.

and timely reconstruction of the speech signal at the receiver end. The most common approach is to store recently arrived packets in a buffer before playing them out at scheduled intervals. By increasing the buffer size, the late loss rate is reduced, but the resulting improvement in voice transmission is offset by the accompanying increase in the end-to-end delay.

This dissertation focuses on two important issues in MD voice transmission system as shown in Figure 1.1: (1) From the viewpoint of QoS, we develop a mult-stream playout scheduling technique to improve the delay-loss tradeoff as well as speech reconstruction quality. (2) We also consider the iterative source-channel decoding algorithm to increase error robustness of MD transmission system. In Section 1.1, MD transmission system and some MD Coding techniques are first reviewed. In Section 1.2, some commonly used playout scheduling schemes are reviewed. The concept of Iterative source-channel decoding is introduced in Section 1.3 and finally, Section 1.4 outlines the structure of the dissertation.

1.1 Multiple Description Coding

MD coding [5] is a method of representing a source with multiple correlated descriptions such that any subset of the descriptions can be used to decode the source with a fidelity that increases with the number of received descriptions. The output symbols of an MD encoder exhibit considerable residual redundancy in terms of both nonuniformity of distribution and their dependencies. This redundancy is due to the nonoptimality of the practically designed source encoder in presence of complexity and delay constraints, or by path diversity as a result of MD coding. The ability to exploit path diversity and source residual redundancy for error robustness makes MD coding an attractive option for the multimedia transmission over unreliable IP networks.

In multimedia communication, MD Coding have been applied to efficient compression of voice and image/video signals. For example, Ingle [6] proposed to separate speech samples into odd and even samples for DPCM encoding. Jiang and Ortega [7] proposed a method which quantizes the even samples in PCM (8 bits/sample), encodes the difference between even and odd samples in ADPCM (2 bits/sample) and then packetizes them into stream 1. Proceeding in a similar approach, the odd samples are quantized in PCM (8 bits/sample) and the difference between odd and even samples in ADPCM (2 bits/sample) and then packetized into stream 2. In [4], Gibson proposed two MD-based speech coding approaches, denoted by MD-AMR and MD-G.729, which are extensions of the AMR-WB codec [8] and the G.729 codec [9], respectively. These two MD coders are design to create balanced descriptions in a way that one lost description can be recovered through interpolation in the received description.

The block diagram of an MD coding system is shown in Figure 1.2. The system has two major components: MD encoder and MD decoder. MD encoder [1][10] splits source samples into two descriptions by using scalar quantizer (SD) followed by index assignment. The index assignment can be represented by a mapping of each reproduction level of the SQ to a unique element in an index assignment matrix. The choice of the index assignment matrix determines the correlation between the descriptions and is the key to realize an MDSQ. Design algorithms for good index assignments are presented in [10]. In these, the inter-description correlation is controlled by choosing the number of diagonals covered by the index assignment. The MDSQ has been extensively studied

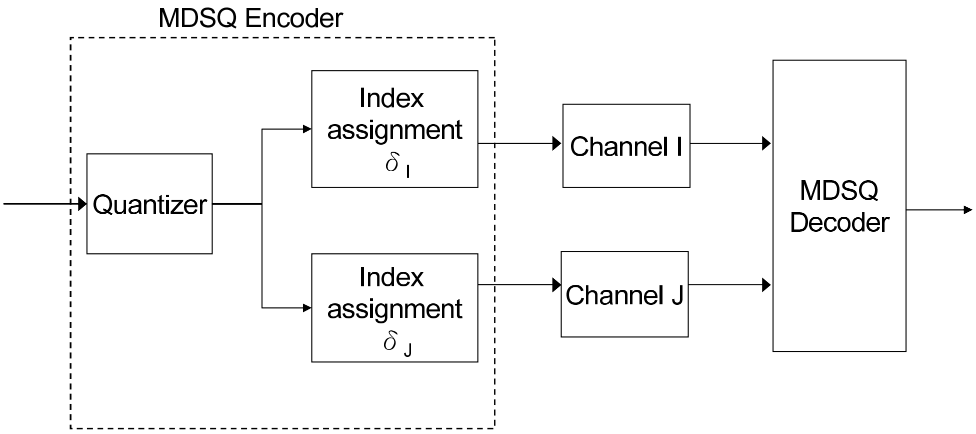

Figure 1.2: Block diagram of multiple description coding system.

for noiseless channels with packet loss, assuming that there exists multiple independent channels that either provide error-free transmission or experience packet erasure. In many practical situations, however, multiple descriptions of the source signals are transmitted over channels that are subject to noise as well as packet loss.

1.2 Joint Playout and FEC Control

Packet loss and delay are the major network impairments for transporting real-time voice over IP networks. The network delay experienced may vary for each packet depending on the level of congestion along the path. The variation in network delay, referred to as jitter, must be smoothed out since it obstructs the proper and timely reconstruction of the speech signal at the receiver end. The most common approach is to store recently arrived packets in a buffer before playing them out at scheduled intervals. By increasing the buffer size, the late loss rate is reduced, but the resulting improvement in voice transmission is off-set by the accompanying increase in the end-to-end delay. In balancing the impairment due to delay and packet loss, two current coding strategies, single and multiple description transmissions, have used different playout buffer algorithms. In single description (SD) coding, a number of adaptive playout buffer algorithms have been proposed that react to changing network conditions by dynamically adjusting the playout delay. Most of them work by taking measurements on the network delays and either compressing or expanding silent periods between consecutive talkspurts. Although there are methods which focused on the delay-loss performance [11], better algorithms have been proposed along with voice quality prediction models for perceptual optimization of playout buffer [12][13]. Taking a different approach, MD coding [2][3][4] exploits the packet path diversity such that each description can be individually decoded for a reduced quality reconstruction, but if all descriptions are available, they can be jointly decoded for a better quality reconstruction. For multi-stream voice transmission, Liang et al., [3] proposed an algorithm which uses the Lagrangian cost function to trade delay versus loss by following a a play-first strategy; that is, it plays out early-arriving descriptions while discarding the later ones. Such a design was based on the assumption that human perceptual experience is more strongly impaired by high latency than packet loss. They neither consider the quality degradation due to frequent switching among playout scenarios nor try to optimize the perceived speech quality by way of a prediction model.

Packet loss in MD voice transmission is a result of not only network loss, but also late loss, which greatly impairs communication quality. Due to the stringent delay budget and the need to output speech continuously, packets experiencing sudden high delay have to be discarded at the receiver end if they arrive later than the scheduled playout deadline. There has been much interest in the use of packet-level forward error control (FEC) to mitigate the impact of packet losses [14]. Most current FEC mechanisms send additional information along with the media stream so that the lost data can be recovered in part from the redundant information. In many applications, however, the losses of successive packets are correlated and a packet loss may be followed by a burst packet loss, which significantly decreases the efficiency of FEC. Furthermore, the loss recovery of FEC is performed at the cost of increased end-to-end delay. This has motivated our investigation into trying to exploit the largely uncorrelated characteristics of packet loss and delay variation on multiple network paths using a joint control of MD and FEC. With an MD scheme coded with FEC we have now more freedom to trade off delay, late loss, and speech reconstruction quality.

Traditionally, the study of FEC for loss recovery and playout buffer adaptation for jitter compensation have proceeded independently. Most packet-level FEC mechanisms send some redundant information along with the media stream so that the lost data can be recovered in part from the redundant information embedded in the later arriving packets. In waiting for the arrival of a minimum required number of packets at the receiving end, loss recovery is performed at the cost of increased end-to-end delay. In view of this potential limitation and the coupling between FEC and playout buffer adaptation [15][16], there is a need to develop a joint FEC and playout control scheme such that the additional delay due to FEC application is dealt within the same optimization framework as for regular MD schemes. Previous efforts toward linking FEC with playout buffer for single-stream transmission can be found in [16], but the assumption on which their algorithm was based may limit its applicability. Specifically, it was assumed that the single-stream network over which the voice packets are sent delivers packets in sequence, and thus if a given packet arrives after its playout time, then all the following packets will also arrive after the playout time of the given packet. This line of reasoning has been challenged by a number of related studies [17] that addressed the possibility of packets delivered out of sequence because of network jitter. As such, the joint FEC and playout control scheme proposed in this work will ignore the constraints imposed by the no-reordering assumption made in [16].

The concept of perceptual optimization is usually realized through the use of E-model [18] to predict the conversational speech quality. However, the E-model does not consider the dynamics of transmission impairments because it relies on the static transmission parameters such as average packet loss and average end-to-end delay. Thus, the E-model may make invalid predictions in dealing with the overall quality issues that MD transmission is focused on. For example, the E-model may only suit single-path transmission with two conceivable playout scenarios; i.e., total loss vs. no-loss of packets. A third scenario, partial loss, however, would rise with MD transmission. That is, with multiple streams sent along two paths, if packets from one path experience erasure or excessive delay, packets from the other path can often be used to conceal the lost packets. Although the partial loss is concealed, the resulting degraded playout quality may be not. In dealing with such reconstruction scheme, the E-model is expected to show two limitations. First, it may fail to register impairments due to reconstruction based on information from a single path as opposed to from both paths, when no packets from either path are lost. Moreover, the resulting detrimental effects that accompany the change in the playout scenarios may thus be ignored and harm its prediction of the overall quality.

In this work, we propose a new objective method for predicting the perceived quality of multi-stream voice transmission. In addition to delay and packet loss, the model also takes into account the quality impairments due to frequent switch of playout scenarios. Based on the new model, we then propose the use of minimum overall impairment as a criterion for perceptual optimization of joint playout buffer and FEC adjustment.

1.3 Iterative Source-Channel Decoding

For MD communication over noisy channels with packet loss, a channel encoder may be used on each description to deal with random bit errors. When the MDSQ is concatenated with convolutional codes, iterative source-channel decoding (ISCD) [19][20] inspired by turbo principle has been shown effective using the source residual redundancy and assisted with the reliability information provided by the soft-output channel decoder. In the so-called MD-ISCD schemes [21][22], source residual redundancy and channel-code redundancy are exploited alternatively by exchanging extrinsic information between the constituent decoders. An iterative decoder consisting of two maximum *a posteriori* probability (MAP) detectors is proposed in [21] for joint decoding of MDSQ and convolutional codes. In [22], a cross decoding strategy was stated that exploits not only the reliability information of every bit in one description but also the extrinsic information from the other description according to the chosen index assignment. In the decoding procedure, MAP detectors operating on soft channel outputs were used for each of the two descriptions in such a way that the output of one MAP detector is combined with inter-description correlation to compute the *a priori* information for the other detector.

In previous works[21][22], MD-ISCD schemes are expected to show two limitations. Firstly, as the source decoder uses two separate MAP detectors with each detector operating on one description, it may report invalid codeword combinations corresponding to the empty cells of the index assignment matrix. In dealing with such situations, an invalid codeword combination is treated as an uncorrectable error and the mean of the source is reconstructed. Secondly, the major part of the iterative decoding process runs on bit-level, but the source decoder itself is realized on symbol-level. This is in part due to the fact that binary convolutional codes are commonly used, so the soft-output channel decoding can be implemented efficiently by the BCJR algorithm [14][23]. It causes the problem that only bitwise source *a priori* knowledge can be exploited by the channel decoder, since the BCJR algorithm is derived based on a bit-level code trellis. For the purpose of applicability, it requires the symbol-to-bit and bit-to-symbol probability conversion in each passing of the extrinsic information between the source and channel decoders. This processing step destroys the bit-correlations within a symbol, thus reducing the effectiveness of iterative decoding.

Recognizing this, we will focus on symbol-based trellis decoding algorithms throughout this paper since they allows to exchange between the source and channel decoders the whole symbol extrinsic information. The first step toward realization is to use sectionalized code trellises rather than bit-level trellises as the bases for soft-output channel decoding of binary convolutional codes. Performance is further improved by using a joint MAP source decoder that processes reliability information received on different channels and combines them with inter-description correlation to provide a better estimate of the transmitted quantizer index.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 introduces some multiple description coding schemes. The source is encoded into multiple redundant descriptions that are separately transmitted over independent network path. Also proposed is a multi-stream voice quality prediction model. In Chapter 3, we propose the use of minimum overall impairment as a criterion for perceptual optimization of joint playout buffer and FEC adjustment. When the MDSQ is cancatenated with channel codes, the concept of extrinsic information from turbo decoding can be adopted for MD iterative source-channel decoding (MD-ISCD) [21]-[22]. Unlike previous works which focused in bit-level ISCD, we present in Chapter 4 a symbol-based iterative decoding of convolutionally encoded multiple descriptions. Finally, Chapter 5 summarizes this dissertation and outlines some directions for future research.

Chapter 2

Multi-Stream Transmission System and Quality Prediction Model

The MD Coding is a technique to generate two or more descriptions, which are sent separately over multiple independent channels. When two or more descriptions are received at the receiver, they can be decoded for acceptable quality reconstruction of the source. A number of MD coding techniques have been proposed for voice communication over mobile IP networks. In this chapter, the MD-G.729 based speech packetization scheme described in [4] was considered for the development of joint playout and FEC control. This section also presents a new objective method for predicting the perceived quality of multi-stream voice transmission.

2.1 Multi-Stream Voice Transmission over a Packet-Erasure Channel

A block diagram of the proposed multi-stream VoIP simulation system is shown in Figure 2.1. The system has four major components: MD speech coder, Internet traffic simulator, delay distribution modelling and adaptive playout buffer. The implementation procedure consisted of description generation and description transmission over two independent network paths. For description generation, the MD-G.729 based on speech packetization scheme described in [4] was used to generate two descriptions from the bitstream of the ITU-T G.729 codec [9]. G.729 is a conjugate-structure algebraic code-excited linear prediction (CS-CELP) codec for encoding narrowband speech at the rate of 8 kbps. It operates on 10-ms speech frames and each speech frame is divided into two subframes and all the parameters except the LPC coefficients are determined once per subframe. The MD-G.729 coder is designed to create two balanced descriptions; i.e., each description is of equal rate 4.6 kbps and speech decoded from either description is of similar quality. During description transmission, the best-effort nature of IP networks results in packets experiencing varying amounts of delay and loss due to different levels of network congestion. To characterize this, we used the ns-2 network simulator [14] to generate the traces of VoIP traffic for different network topologies and varying network load. Meanwhile, traces were also extended for varying link loss rates. A value ranging from 0-30% was used to simulate losses with different degrees of severity. Figure 2.2 shows a two path multi-hop network topology for our simulation, with transmission control protocol (TCP) data traffic on both paths contending simultaneously for network resources. The three nodes situated between source and destination on each path (N1 through N3 on the top path and N4 through N6 on the bottom), represent the data access points, each with a number of data sources attached, thus channelling in a large amount of incoming TCP traffic heading for different destinations. On each path a constant bit rate (CBR) voice stream is transmitted in 10-ms UDP packets at a rate of 4.6 kbps. The running time for each simulation is 15 seconds.

At the receiver, a playout buffer is employed to improve the tradeoff among delay, late loss rate, and speech reconstruction quality. We focused on adaptive algorithms which adjust the playout buffer at the beginning of each talkspurt and subsequent pack-

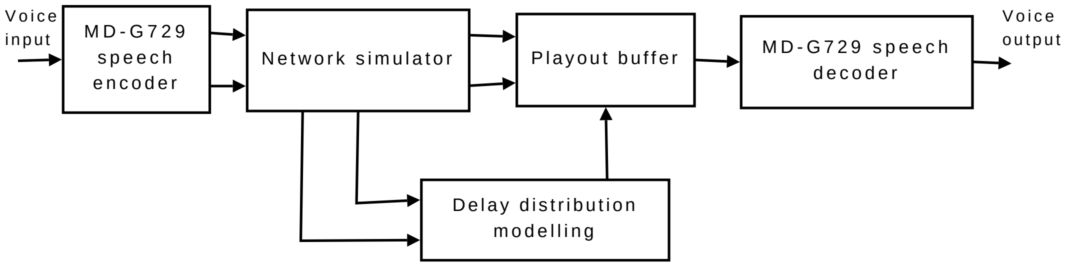

Figure 2.1: A two-channel VoIP simulation system.

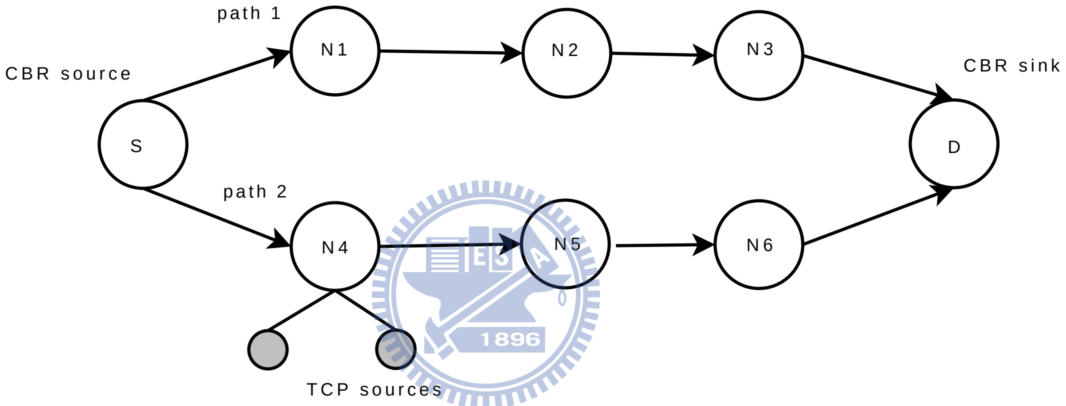

Figure 2.2: A multi-hop transmission model for network simulations.

ets of that talkspurt are played out with the generation rate at the sender. Scheduling the playout of multiple voice streams is formulated as an optimization problem on the basis of a minimum overall impairment criterion. In addition to packet loss and delay, it takes into account the dynamics of transmission impairments due to frequent switch of playout scenarios. To proceed with this, it is a prerequisite to establish a delay distribution model as it provides a direct link to late loss rate in the presence of jitter. Previous work in [13] has found that the delay characteristics of VoIP traffic can be represented by statistical models which follow Pareto, Normal and Exponential distributions depending on applications. Finally, the MD-G.729 bit stream is decoded to generate the degraded speech. In our experiments, the decoder deals with the loss of two descriptions by using the error concealment algorithm of G.729 [9], while in other situations speech packets are reconstructed depending on how many descriptions are received by the playout deadline. If both descriptions are received, the central decoder performs the standard G.729 decoding process after combining the two descriptions into one bitstream. If only one description is lost, the side decoder substitutes the missing information by using received parameters from the other description or information from the most recent correctly received frame [4].

2.2 Multi-Stream Voice Quality Prediction Model

In Section 1.2 we stated two limitations to E-model to predict the conversation speech quality in the third scenario. First, it may fail to register impairments due to reconstruction based on information from a single path as opposed to from both paths, when no packets from either path are lost. Moreover, the resulting detrimental effects that accompany the change in the playout scenarios may thus be ignored and harm its prediction of the overall quality. Recognizing this, we propose a new objective method for predicting the perceived quality of multi-stream voice transmission. In addition to delay and packet loss, the model also takes into account the quality impairments due to frequent switch of playout scenarios.

Conceptually the proposed model followed the commonly used ITU E-model [18] in defining factors that affect the perceptual quality of the MD voice transmission. As an analytical model of conversational speech quality used for network planning purposes, the E-model combines individual impairments due to the signal's properties and the network characteristics into a single R-factor, ranging from 0 to 100. In VoIP applications [24], the R-factor may be simplified as follows: $R = 94.2 - I_d - I_e$, where I_d represents the delay impairment. I_e is known as the equipment impairment and accounts for impairments due to speech coding and packet loss. The delay impairment can be derived by a simplified fitting process in [24] with the following form

$$I_d(d) = 0.024d + 0.11(d - 177.3)H(d - 177.3) \tag{2.1}$$

where d is the end-to-end delay and $H(x)$ is the step function. The E-model, originally proposed for single-stream transmission, is only applicable to a limited number of speech codecs and network conditions, since it requires time-consuming subjective tests to derive the I_e model. With multiple voice streams, any subset can be used for signal reconstruction, and the transmission quality improves with the size of the subsets. In addition to delay and packet loss, a good quality prediction model should take into account the impairments due to dynamic size allocations during the speech playout.

For two-path transmission, each channel can either deliver or erase the transmitted description, so the two channels will always be in one of four possible states: no loss, loss in channel 1, loss in channel 2, and loss in both channels (packet erasure). Among them, only the speech resulting from the packet-erasure state is not affected by playout buffer operations. The receiver deals with the loss of both descriptions by using the error concealment algorithm of G.729 codec to conceal the erased packet. If, additionally, speech decoded from either MD-G.729 description is assumed to be of similar quality, we only need to consider two kinds of playout scenarios at the receiver end. Specifically, a packet is 1) fully restored with two descriptions and thus played with high quality; and 2) partially restored with one description and thus played with degraded quality. For brevity, let S_k denote the scenario that k descriptions are received before the playout time. Conditioned on the event that the packet can be restored, we let q_k be the probability to play out the packet using k descriptions. Formally, it is given by $q_k = P(S_k)/(P(S_1) + P(S_2))$. It is improtant to notice that quality degradation resulting from S_1 and S_2 are different perceptual experiences. For scenario S_2, the standard G.729 decoding process is carried out after combining the two descriptions into one bitstream. Let $I_{e,k}$ denote the equipment impairment as a result of playing out k received descriptions. From the perceived QoS perspective, the MD-G.729 codec may be viewed as operating at two coding rates: 4.6 kbps for S_1 and 8 kbps for S_2. By taking frequent switch of coding rates into account, we define the average equipment

1896

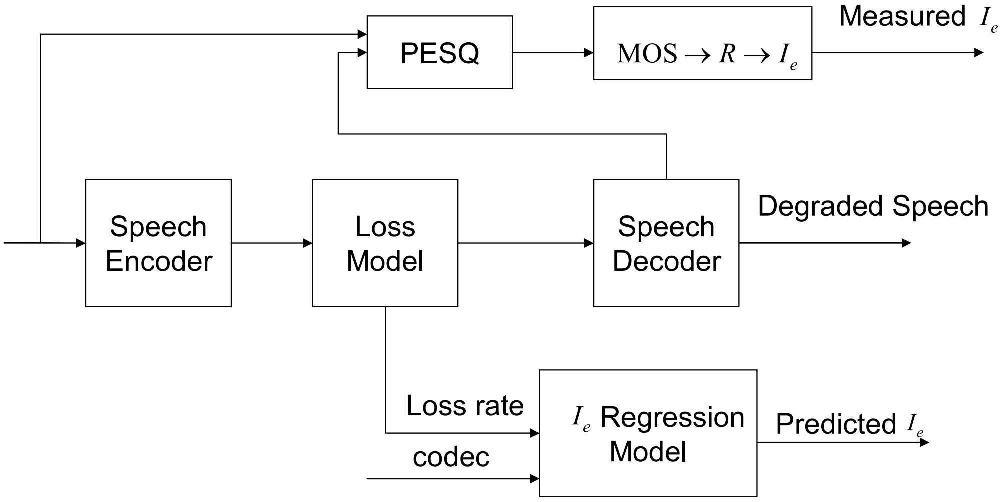

Figure 2.3: Schematic diagram for prediction of I_e model.

impairment due to MD-G.729 coding as follows:

$$I_e(e) = q_1 I_{e,1}(e) + q_2 I_{e,2}(e). \tag{2.2}$$

1896

The next issue to be addressed is how to derive an equipment impairment $I_{e,k}$ corresponding to each playout scenario S_k. We followed the work of [12], which describes an objective method for prediction of $I_{e,k}$ regression model using the PESQ algorithm [25]. As shown in Fig. 2.3, each single measurement consists of three steps and is repeated several times with different transmission configurations. First, a speech sample is selected from an English speech database that contains 16 sentential utterances spoken by eight males and eight females. Each sample has a duration of 8 seconds and sampled at 8 kHz. Second, the speech sample is encoded using MD-G.729 codec and then processed in accordance with the simulated loss model to generate the degraded speech. In our experiments, the decoder deals with packet erasure by using the error concealment algorithm of G.729 [9] to conceal erased packets, while in other scenarios speech packets are reconstructed depending on how many descriptions are received by the playout deadline. Third, the reference speech and degraded speech are processed by the PESQ to obtain a mean opinion score (MOS). For each speech sample, a MOS value for one packet-erasure rate is obtained by averaging over 30 different erasure locations in order to remove the influence of erasure location. Further, these MOS values are averaged over all speech samples and then converted to a rating R to give an equipment impairment value $I_{e,k} = 94.2 - R$. The R-factor can be obtained from the average MOS with a conversion formula as follows:

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.06MOS - 57.336. \tag{2.3}$$

Fig. 2.4 shows that impact of transmission scenario S_k and packet-erasure rate e on the equipment impairment $I_{e,k}$ with a packetization of one frame per packet. The $I_{e,k}$ value for zero packet-erasure rate represents the codec impairment itself. It is obvious that the speech playout resulting from S_2 has a lower codec impairment and has a high robustness to packet loss. By inspecting Figure 2.4, we observe that our measured $I_{e,2}$ value for zero packet erasure, 21.96, is inconsistent with the ITU-published I_e value, 10, for codec G.729 [9]. One possible reason for this discrepancy may lie in the codec algorithm. As the G.729 is a CELP-based codec, the use of linear predictive model of speech production can lead to variations in codec performance with different talkers or languages [26]. Support for such a speculation can be found in at least two studies using the same codec [12][27], which, in case of zero packet loss and using different speech samples from the ITU-T data set [28], rendered measured I_e values of 21.14 and 17.128, respectively, similar to the value obtained for this study. From the curves, a nonlinear regression model can be derived for each $I_{e,k}$ by the least-squares data fitting method. The fitting curves are also shown in Figure 2.4. The derived $I_{e,k}$ model for scenario S_k has the following form: $I_{e,k}(e) = \gamma_{1,k} + \gamma_{2,k} \ln(1 + \gamma_{3,k} e)$, where e is the packet-erasure rate in percentage. Our findings indicate that the regression model parameters $(\gamma_1, \gamma_2, \gamma_3)$ for S_1 are $(52.61, 7.52, 10)$ and $(21.96, 17.02, 16.09)$ for S_2.

2.3 Experimental Results

A set of experimental conditions was designed for the use of artificially degraded speech samples to verify the detrimental effects estimated by the proposed I_e regression model in relation to the traditional E-model. The two models, despite their agreement in

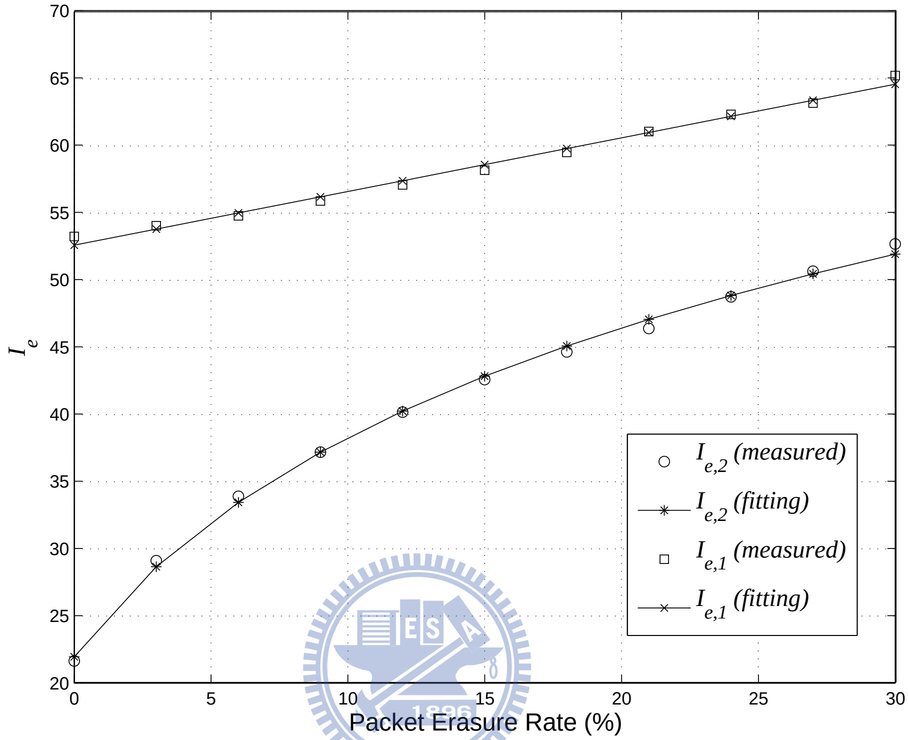

Figure 2.4: $I_{e,k}$ vs. packet erasure rate e.

including packet loss as a main impairment factor, differ in how reconstruction in conditions with partial packet losses is treated. The proposed model differentiates partial reconstruction with one description from full reconstruction with two descriptions. The three states of frame reconstruction dictated by the model are 1) fully restored, when both descriptions are available and thus played with high quality, 2) partially restored, when only one description is available and thus played with less than optimal quality, and 3) restored by the G.729 error concealment algorithm, when both descriptions are lost during transmission. In contrast, the traditional model treats the full and the partial reconstruction states uniformly as the no-loss state, leaving out any differentiation of the processes involved that lead to the no-loss at the receiver end. It is thus reasonable to hypothesize that the traditional model fails to register any quality impairment due to partial reconstruction. As such, if the I_e's estimated with the two models show

Table 2.1: I_e comparison for different prediction models.

Speech	$e\%$	$q_1\%$	Traditional I_e	Proposed I_e	Measured I_e
Female	9.88	6.48	43.16	44.35	44.54
	4.93	22	34.41	39.48	40.84
Male	4.84	14	31.78	34.97	36.57
	12	31	40.37	45.67	46.27

significant differences in their closeness to the I_e's measured, then adding such a differentiation scheme into the modelling process should prove a valid approach. The speech samples considered here were one male and one female utterance. The G.729 speech codec and the proposed MD coding scheme were used sequentially, which turned each utterance into a bitstream of frames with two identical descriptions to be transmitted along separate dynamically-changing paths. At the receiver end, each utterance was artificially degraded to render two tokens, each with its own composition of frames of the three reconstruction states. Since the proposed model diverges from the traditional model by treating the loss of one packet as a separate state from either total loss or no loss, the underlying variable being manipulated in the frame composition was the rate q_1 of partial loss. Thus, there was a total of four test conditions.

1896

Table 2.1 lists for each condition the percentages of frames that are erased and restored with only one description, followed by the three corresponding I_e's as estimated by the traditional model, by the proposed model, and as measured then converted with PESQ. The results showed that, unlike the traditional model that yielded poorer estimations for samples containing higher percentages of one description loss, the proposed model gave estimations that are quite robust regardless of the sample frame composition. For example, given the same percentage increases from 6.84% to 22% and from 14% to 31% in the female and the male utterance respectively, the traditional model showed deviations from the measured I_e's that were increased from 1.38 to 6.43 and from 4.79 to 5.9, respectively, while the proposed model yielded across conditions more stable and smaller deviations that ranged from 0.6 to 1.6. Taken together, these comparison data suggest that independent evaluation of impairments due to loss of one vs. both descriptions adds to the robustness of the proposed model.

Chapter 3

QoS Control for Multi-Stream Voice over IP Networks

In this chapter, we study the QoS Control of Multi-Stream Voice over IP Networks. The proposed MD system model has been presented in Chapter 2. Section 3.1 presents a perceptual-based playout mechanism which uses optimization criterion based on new quality prediction model proposed in Section 2.2. A further step toward perceptual optimization, a packet-level FEC channel encoder is included into our proposed MD system to strengthen the error concealing capacities of MD system. Also proposed is a joint playout and FEC control scheme in Section 3.2.

3.1 Perceptual-Based Playout Mechanism

Packet loss and delay are the major network impairments for transporting real-time voice over IP networks. In the proposed system, multiple descriptions of the speech are used to take advantage of the packet path diversity. Our goal is to develop is a multi-stream playout buffer algorithm, together with an adaptive parameter adjustment scheme, that maximizes the perceived speech quality via delay-loss trading. Experimental results showed that, compared to FEC-protected single-path transmission, the proposed multi-stream transmission scheme achieves significant reductions in delay and packet loss rates as well as improved speech quality.

3.1.1 Adaptive Playout Scheduling Algorithm

The main attraction of multi-stream transmission arises from its flexibility to trade off different sources of impairments against each other. Waiting for the arrival of both descriptions results in lower equipment impairment I_e, but at the cost of higher delay impairment I_d. On the other hand, playing out the voice description with lower delay avoids latency, but increases the equipment impairment. Since playout scheduling aims to improve the overall conversational speech quality, which hangs on the balance between delay and packet loss, full reconstruction of both descriptions may not always be the priority if the overall impairment does not justify the extra delay from waiting. Given that, the design of a playout buffer must play around with switching between different playout scenarios in order to maximize the benefits of packet path diversity. To accomplish this goal, the proposed voice quality prediction model is applied on adaptive control of the multi-stream playout buffer. Prior to the arrival of each packet i, the playout delay for that packet is determined according to the past recorded delays. The playout delay of packet i is denoted by $d_{play,i}$, which is defined as the time from the moment that packet is delivered to the network until it has to be played out. A packet may get lost due to its late arrival, if its network delay is larger than the playout delay.

The basic adaptive playout algorithm estimates two statistics characterizing the network delay, and uses them to calculate the playout delay as follows:

$$d_{play,i} = \hat{d}_i + \beta \hat{v}_i. \tag{3.1}$$

where $\hat{d}_i$ and $\hat{v}_i$ are running estimates of the mean and variation of network delay seen up to the arrival of the ith packet. The safety factor β has a critical impact on the tradeoff between delay and late packet loss, which in turn influences the conversational speech quality. From (3.1) it can be deduced that increasing β leads to lower late loss rate as more packets arrive in time, however the end-to-end delay increases. All of the algorithms [11-13] used a fixed value of β, e.g., $\beta = 4$, to set the buffer size, so that only a small fraction of the arriving packets should be lost due to late arrival. In this work, a β-adaptive algorithm is instead used to control the playout buffer so that the reconstructed voice quality is maximized in terms of delay and loss. The idea behind our algorithm is to adaptively adjust the value of β with each incoming talkspurt, depending on the variation in the network delays.

3.1.2 Perceptually Motivated Optimization Criterion

Next, we formulated the parameter adjustment as a perceptually motivated optimization problem and the adopted criterion relies on the use of the proposed multi-stream voice quality prediction model. Let d_i be the end-to-end delay experienced by the ith packet, which consists of encoding delay d_c and playout delay $d_{play,i}$. e_i is the packet-erasure probability to lose two descriptions, no matter if the description is dropped by the network or discarded due to its late arrival, and is given by

$$\begin{aligned} e_i = \; & e_n^{(1)} e_n^{(2)} + e_n^{(1)}(1 - e_n^{(2)}) e_{b,i}^{(2)} + e_n^{(2)}(1 - e_n^{(1)}) e_{b,i}^{(1)} \\ & + (1 - e_n^{(1)})(1 - e_n^{(2)}) e_{b,i}^{(1)} e_{b,i}^{(2)} \end{aligned} \tag{3.2}$$

where $e_n^{(l)}$ and $e_{b,i}^{(l)}$ represent the link loss probability and estimated late loss probability of packet i in stream l, respectively. Now, we define an overall impairment function I_m which is a function of both d_i and e_i, with $I_m(d_i, e_i) = I_d(d_i) + I_e(e_i)$. Using (2.2) and (2.3), I_m can be expressed as

$$I_m(d_i, e_i) = 0.024 d_i + 0.11(d_i - 177.3) H(d_i - 177.3) + \textstyle\sum_{k=1,2} q_k I_{e,k}(e_i). \tag{3.3}$$

where $q_1 + q_2 = 1$ and the probability to receive both descriptions is given by

$$q_2 = \frac{1}{1 - e_i}(1 - e_n^{(1)})(1 - e_n^{(2)})(1 - e_{b,i}^{(1)})(1 - e_{b,i}^{(2)}). \tag{3.4}$$

Our optimization framework requires an analytic expression for the packet erasure probability e_i as a function of the single parameter β_i. Notice that $e_{b,i}^{(l)}$ and the playout delay $d_{play,i}$ are strongly correlated, and to find out their relationship, the network delays of stream l are assumed to follow a Pareto distribution which is defined as $F_l(x) = 1 - (g_l/x)^{\alpha_l}$. The parameters of Pareto distribution α_l and g_l can be estimated from past recorded delays using the maximum likelihood estimation method [8]. Then, the late loss probability of packet i in stream l can be computed as follows:

$$e_{b,i}^{(l)} = 1 - F_l(d_{play,i}) = (g_l/d_{play,i})^{\alpha_l} \tag{3.5}$$

This reduces the expression of the packet-erasure probability e_i to be a function of the playout delay $d_{play,i}$, which in turn is a function of the parameter β_i. Its gradient with respect to β_i is given by

$$\begin{aligned} \frac{de_i}{d\beta_i} &= \tfrac{-\hat{v}_i}{d_{play,i}}\{(1-e_n^{(1)})(1-e_n^{(2)})e_{b,i}^{(1)}e_{b,i}^{(2)}(\alpha_1+\alpha_2) \\ &\quad +e_n^{(1)}(1-e_n^{(2)})e_{b,i}^{(2)}\alpha_2 + e_n^{(2)}(1-e_n^{(1)})e_{b,i}^{(1)}\alpha_1\} \end{aligned} \tag{3.6}$$

The overall impairment function I_m is a function of the playout delay $d_{play,i}$ and the probability q_k as well as the packet-erasure probability e_i. Since these parameters are all functions of the parameter β_i, the overall impairment I_m is also a function of β_i, i.e., $I_m(d_i, e_i) = I_m(\beta_i)$. By differentiating it with respect to β_i, we get the following equation for the gradient:

$$I_m'(\beta_i) = c\hat{v}_i + \textstyle\sum_{k=1,2}\{q_k \frac{\gamma_{2,k}\gamma_{3,k}}{1+\gamma_{3,k}e_i}\frac{de_i}{d\beta_i} + \frac{dq_k}{d\beta_i}I_{e,k}(e_i)\}. \tag{3.7}$$

where

$$c = \{ \begin{array}{ll} 0.024, & \beta_i < (177.3 - d_c - \hat{d}_i)/\hat{v}_i; \\ 0.134, & \beta_i > (177.3 - d_c - \hat{d}_i)/\hat{v}_i. \end{array} \tag{3.8}$$

$$\begin{aligned} &\tfrac{dq_2}{d\beta_i} = \tfrac{\hat{v}_i}{d_{play,i}(1-e_i)}(1-e_n^{(1)})(1-e_n^{(2)})[\alpha_1 e_{b,i}^{(1)}(1-e_{b,i}^{(2)}) \\ &+\alpha_2 e_{b,i}^{(2)}(1-e_{b,i}^{(1)})] + \tfrac{1}{(1-e_i)^2}\tfrac{de_i}{d\beta_i}(1-e_n^{(1)})(1-e_n^{(2)})(1-e_{b,i}^{(1)})(1-e_{b,i}^{(2)}) \end{aligned} \tag{3.9}$$

3.1.3 Perceptual Optimization of Playout Buffer

Our general problem can be stated as follows: Given estimates of the parameters characterizing the delay distribution and I_e regression model, find the optimal value of β_i so as to minimize the overall impairment function $I_m(\beta_i)$. This task belongs to the class of set-constrained optimization problems, which can be solved efficiently by means of one-dimensional search methods [29]. For computational purposes, we applied the secant method [29] to search for the minimizer $\hat{\beta}_i$ of I_m over the constraint set $\{\beta_i \in R, \beta_i > 0\}$. Starting with two initial values $\beta_i(-1)$ and $\beta_i(0)$, the iterative formula for the secant algorithm at the j-th iteration has the form

$$\beta_i(j+1) = \beta_i(j) - \frac{\beta_i(j) - \beta_i(j-1)}{I'_m(\beta_i(j)) - I'_m(\beta_i(j-1))} I'_m(\beta_i(j)). \tag{3.10}$$

The new value $\beta_i(j+1)$ is then used in the next iteration and the estimation process is repeated until the difference $|\beta_i(j+1) - \beta_i(j)|$ is smaller than a threshold. Finally, we summarize the proposed multi-stream playout buffer algorithm as below.

1. Apply an autoregressive algorithm [11] to estimate the delay mean $\hat{d}_i^{(l)}$ and variance $\hat{v}_i^{(l)}$ for individual stream l $(l = 1, 2)$ as follows:

   $$\hat{d}_i^{(l)} = \mu \hat{d}_{i-1}^{(l)} + (1-\mu) n_i^{(l)}. \tag{3.11}$$

   $$\hat{v}_i^{(l)} = \mu \hat{v}_{i-1}^{(l)} + (1-\mu) |n_i^{(l)} - \hat{d}_i^{(l)}|. \tag{3.12}$$

   where $n_i^{(l)}$ is the network delay of packet i in stream l and $\mu = 0.998002$ is a weighting factor for convergence control.

2. At the beginning of each talkspurt, update network delay records for the past $L = 200$ packets in every stream l $(l = 1, 2)$, and use them to calculate the Pareto distribution parameters (α_l, g_l) by the maximum likelihood estimation method. Given a set of past network delays $\{n_{i-1}^{(l)}, n_{i-2}^{(l)}, \ldots, n_{i-L}^{(l)}\}$, we compute

   $$g_l = \min\{n_{i-1}^{(l)}, n_{i-2}^{(l)}, \ldots, n_{i-L}^{(l)}\} \tag{3.13}$$

$$\alpha_l = L/\Sigma_{j=i-1}^{i-L} \log(\frac{n_j^{(l)}}{g_l}) \tag{3.14}$$

3. Use the values of (α_l, g_l) in the secant method to determine the minimizer $\hat{\beta}_i^{(l)}$ of the utility function,

$$I_m(\beta_i^{(l)}) = I_d(d_c + \hat{d}_i^{(l)} + \beta_i^{(l)} \hat{v}_i^{(l)}) + I_e(e_i(\beta_i^{(l)})). \tag{3.15}$$

4. Set the playout delay to

$$\begin{aligned} &d_{play,i} = \hat{d}_i^{(l^*)} + \hat{\beta}_i^{(l^*)} \hat{v}_i^{(l^*)}, \\ &l^* = \arg\min\{I_m(\hat{\beta}_i^{(l)}), l = 1, 2\} \end{aligned} \tag{3.16}$$

3.1.4 Experimental Results

Computer simulations were carried out to evaluate the performances given by three examples, MD1-3, of the MD voice transmission scheme, which used the 9.2 kbps MD-G.729 codec for the generation of two balanced descriptions. An FEC-protected single description (SD) transmission scheme was also tested for its comparative strength. The SD scheme applied the 8 kbps G.729 codec and performed packet-level (9,8) Reed-Soloman channel code, a condition in which an FEC packet was generated for every 8 packets and whenever any 8 of the 9 packets (8 + the resulting FEC packet) had been received over a period of time, the 8 packets were fully recovered at the receiver end. It was hypothesized that the performances of these four schemes being tested would be set apart mainly by the value of β (fixed or dynamically changing) they each assumed during the test period, and that the best performance should come with β values whose calculation was based on link loss, packet-erasure loss and various transmission scenarios. MD1 had a fixed $\beta = 4$, and MD2 took values of β that were dynamically adjusted by the playout buffer according to the proposed voice quality prediction model. MD3 differed from the previous two by having its β set following the play-first strategy proposed by Liang et al [3]. The SD scheme, with the FEC feature,

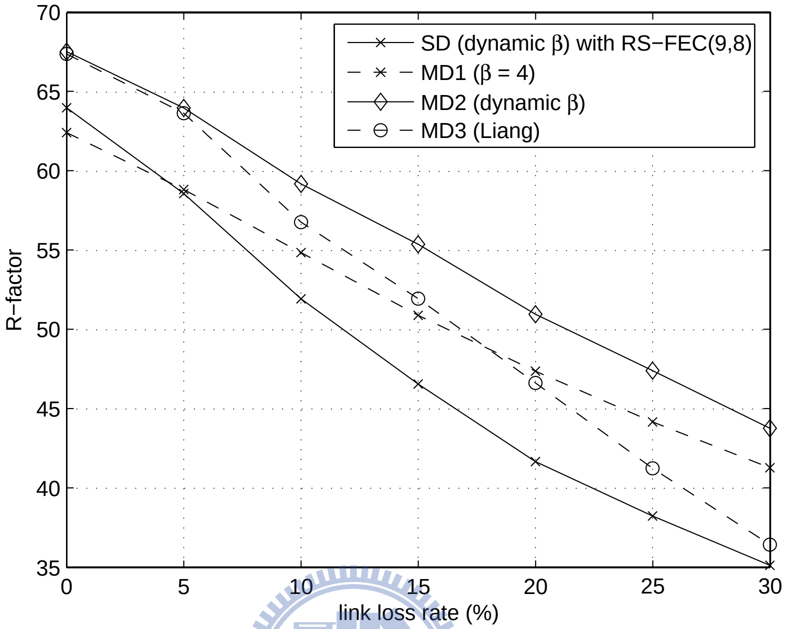

Figure 3.1: Performance comparison for different playout algorithms.

assumed a dynamic β value as determined by the E-model [30]. The speech data fed into the simulations were two sentential utterances spoken by one male and one female, each sampled at 8 kHz. and 8 seconds in duration. Both samples were encoded and then processed in accordance with the delay and loss characteristics of the trace data to degrade the speech. Figure 3.1 plots the perceived speech quality for the SD and the 3 MD schemes as a function of the link loss rate. As described in Section 2.3, the perceived quality was gauged by calculating the predicted average R-factor according to the E-model, and the link loss rate was varied from 0-30%. It can be seen that, although the quality deteriorated for all four schemes as the link loss rate increased, the three MD schemes yielded better speech quality than the SD scheme, especially at increased link loss rate. At rates slightly beyond the minimum (eg.,5%), the SD scheme, despite its FEC feature, started showing incapability of recovering the lost packets in facing link losses. Among the three MD schemes that showed three levels of dynamics in making decisions about delay, MD1, with its fixed β value, yielded the worst quality

Table 3.1: MOS comparison for different playout algorithms.

Link loss rate (%)	0	5	10	15	20	25	30
SD with RS-FEC(9,8)	3.305	3.028	2.678	2.396	2.147	1.979	1.833
MD1	3.226	3.040	2.832	2.623	2.439	2.274	2.128
MD2	3.481	3.305	3.059	2.860	2.627	2.441	2.254
MD3	3.473	3.288	2.942	2.677	2.399	2.126	1.893

at 0% link loss rate, yet showed better results at rates above 20% than MD3, suggesting a limitation of the Lagrangian cost function in predicting the actual perceived speech quality. The best results, as hypothesized, were obtained with the currently proposed scheme MD2, which can be attributed to its all encompassing algorithm and thus the overall function that takes into account the impairment impacts as a result of delay, packet-erasure loss and various transmission scenarios. To elaborate further, MOS performances of various playout algorithms were examined for MD transmission with link loss rates ranging from 0% to 30%. As shown in Table 3.1, the results indicate that the proposed playout algorithm is preferable to other algorithms in all the tests and its performance gain tends to increase with increasing link loss rates.

3.2 Jont FEC and Playout Control Mechanisms

This Section presents a joint playout buffer and FEC adjustment scheme that maximizes the perceived speech quality via delay-loss trading. Figure 3.2 shows a block diagram of the simulation system with the first two components, MD speech coder and channel coder, responsible for description generation and the rest, for transmission and signal reconstruction. After source coding, packet-level Reed-Solomon (N, K) codes [14] are used for channel coding of individual descriptions. The channel encoder takes a codeword of K speech packets and generate $N - K$ additional FEC check packets for the transmission of N packets over the network. Such a code, denoted as a RS (N, K)

code, is able to recover all losses in the block if and only if at least K out of N packets are received correctly. The receiver end features an adaptive playout buffer that smooths out the network jitter. The algorithm adjusts the playout buffer at the beginning of each talkspurt and subsequent packets of that talkspurt are played out with the generation rate at the sender. A joint design of FEC and playout buffer adaptation was further formulated as an optimization problem on the basis of a minimum overall impairment criterion. In addition to packet loss and delay that traditional systems sought to control, this design takes into account the dynamics of transmission impairments due to frequent switch of playout scenarios. Experimental results showed that the proposed multi-stream voice transmission scheme achieves significant reductions in delay and packet loss rates as well as improved speech quality.

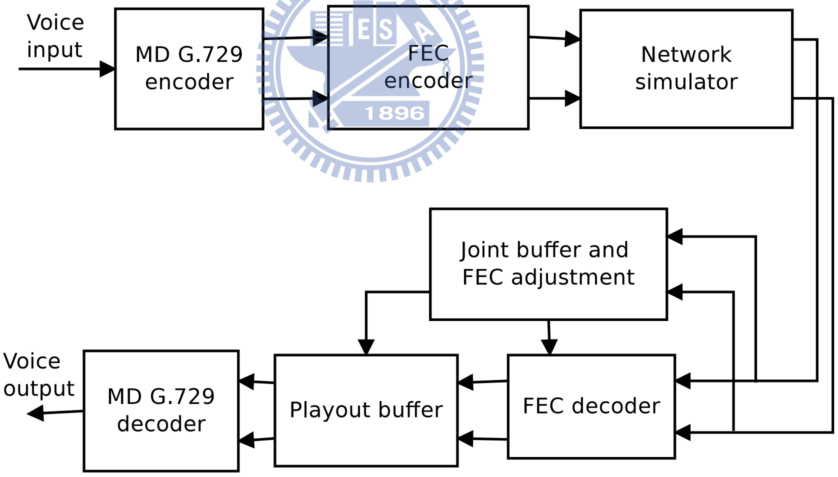

Figure 3.2: A multi-description voice transmission system.

3.2.1 FEC in a Gilbert-Model Loss Process

In Section 1.2 we stated the rationale in combining FEC into the playout buffer algorithm without following the no-reordering assumption underlying the work in [16]. Assume that multiple descriptions of the speech are transmitted over independent network paths and each path is characterized by a Gilbert-model loss process. The Gilbert model is a two-state Markov chain model in which state B represents a network loss and state G represents a packet reaching the destination. For each stream l, the parameters $p^{(l)}$ and $q^{(l)}$ denote respectively the probabilities of transitions from G to B states and from B to G states. A packet is said to be missing so long as the packet is either dropped in the network or discarded due to its late arrival. For the sake of clarity, every packet i is assigned a variable $W_i \in \{0, 1, 2\}$, corresponding to the following 3 arrival scenarios: $W_i = 0$, arriving before its playout time, $W_i = 1$, a network loss, $W_i = 2$, arriving after its playout time. Following the development of (Boutremans and Boudec, 2003), let $R^{(l)}(m, n, D_{F,i})$ denote the probability that $m - 1$ packets are missing (dropped or received late) in the next $n - 1$ packets following the network loss of packet i, and let $S^{(l)}(m, n, D_{F,i})$ denote the probability that $m - 1$ packets are missing in the next $n - 1$ packets following the late loss of packet i. Similarly, let $\tilde{R}^{(l)}(m, n, D_{F,i})$ and $\tilde{S}^{(l)}(m, n, D_{F,i})$ denote the probability that $m - 1$ missing packets occur in the last $n - 1$ packets preceding packet i which is dropped and received late, respectively. As shown in the Appendix A, these probabilities can be computed by recurrence as follows:

$$
\begin{aligned}
&R^{(l)}(m, n, D_{F,i}) = \qquad\qquad (3.17)\\
&\begin{cases}
q^{(l)}(1 - p^{(l)})^{n-2} \cdot \prod_{h=1}^{n-1}(1 - e_{b,i+h}^{(l)}), \quad m = 1, n \geq 1 \\[2ex]
(1 - q^{(l)})R^{(l)}(m - 1, n - 1, D_{F,i+1}) \\
+ \displaystyle\sum_{j=1}^{n-m} \{q^{(l)}(1 - p^{(l)})^{j-1} \prod_{h=1}^{j}(1 - e_{b,i+h}^{(l)}) \cdot \{p^{(l)} R^{(l)}(m - 1, n - j - 1, D_{F,i+j+1}) \\
+(1 - p^{(l)}) e_{b,i+j+1}^{(l)} S^{(l)}(m - 1, n - j - 1, D_{F,i+j+1})\}\}, \quad 2 \leq m \leq n
\end{cases}
\end{aligned}
$$

and

$$S^{(l)}(m,n,D_{F,i}) = \tag{3.18}$$
$$\begin{cases} e_{b,i}^{(l)}(1-p^{(l)})^{n-1}\cdot\prod_{h=1}^{n-1}(1-e_{b,i+h}^{(l)}), \quad m=1, n\geq 1 \\ \\ \displaystyle\sum_{j=0}^{n-m}\{e_{b,i}^{(l)}(1-p^{(l)})^{j}\prod_{h=1}^{j}(1-e_{b,i+h}^{(l)})\cdot\{p^{(l)}R^{(l)}(m-1,n-j-1,D_{F,i+j+1}) \\ +(1-p^{(l)})e_{b,i+j+1}^{(l)}S^{(l)}(m-1,n-j-1,D_{F,i+j+1})\}\}, \quad 2\leq m\leq n \end{cases}$$

where $D_{F,i}$ is the FEC delay and $e_{b,i}^{(l)}$ is the estimated late loss probability of packet i in stream l. Table 3.2 summarizes the basic notation used in the Appendix.

Table 3.2: Basic Notation.

Notation	Description
$D_{F,i}$	FEC delay of packet i
e_i	Packet-erasure probability of packet i
$e_{b,i}^{(l)}$	Late loss probability of packet i in stream l
$P_L^{(l)}(i)$	Resudual loss probability of packet i in stream l after FEC is used
$P_{R1}^{(l)}(i)$	Probability to recover a dropped packet i in stream l
$P_{R2}^{(l)}(i)$	Probability to recover a late lost packet i in stream l
$R^{(l)}(m,n,D_{F,i})$	Probability that $m-1$ packets are missing in the next $n-1$ packets following the network loss of packet i
$S^{(l)}(m,n,D_{F,i})$	Probability that $m-1$ packets are missing in the next $n-1$ packets following the late loss of packet i
$\tilde{R}^{(l)}(m,n,D_{F,i})$	Probability that $m-1$ packets are missing in the last $n-1$ packets proceeding the network loss of packet i
$\tilde{S}^{(l)}(m,n,D_{F,i})$	Probability that $m-1$ packets are missing in the last $n-1$ packets proceeding the late loss of packet i

With RS (N,K) code, each code takes a codeword of K voice packets and generates $N-K$ additional FEC packets for the transmission of N packets over the network. Such a code is able to recover any missing packet in the block if and only if at least K out of N packets in this block are received before their playout time. Viewed from this perspective, the probability to recover a dropped packet is given by

$$
\begin{aligned}
& P_{R1}^{(l)}(i) \\
& = \Pr(\text{packet } i \text{ can be recovered} \mid \text{packet } i \text{ is dropped in the network}) \\
& = \sum_{L=1}^{N-K} \Pr(L \text{ packets are missing in } W_1^N | W_i = 1) \\
& = \sum_{L=1}^{N-K} \sum_{m=0}^{min(L-i,i-1)} \Pr(\ m \text{ packets are missing in } W_1^{i-1} | W_i = 1) \\
& \cdot \Pr(\ L-m-1 \text{ packets are missing in } W_{i+1}^N | W_i = 1) \\
& = \sum_{L=1}^{N-K} \sum_{m=0}^{min(L-i,i-1)} \tilde{R}^{(l)}(m+1, i, D_{F,i}) \cdot R^{(l)}(L-m, N-i+1, D_{F,i})
\end{aligned}
\tag{3.19}
$$

and the probability to recover a late lost packet is given by

$$
\begin{aligned}
& P_{R2}^{(l)}(i) \\
& = \Pr(\text{packet } i \text{ can be recovered} \mid \text{packet } i \text{ is received late}) \\
& = \sum_{L=1}^{N-K} \Pr(L \text{ packets are missing in } W_1^N | W_i = 2) \\
& = \sum_{L=1}^{N-K} \sum_{m=0}^{min(L-i,i-1)} \Pr(m \text{ packets are missing in } W_1^{i-1} | W_i = 2) \\
& \cdot \Pr(L-m-1 \text{ packets are missing in } W_{i+1}^N | W_i = 2) \\
& = \sum_{L=1}^{N-K} \sum_{m=0}^{min(L-i,i-1)} \tilde{S}^{(l)}(m+1, i, D_{F,i}) \cdot S^{(l)}(L-m, N-i+1, D_{F,i})
\end{aligned}
\tag{3.20}
$$

Using these probabilities, we can compute the residual loss probability (after FEC is used) as follows:

$$
P_L^{(l)}(i) = e_n^{(l)}(1 - P_{R1}^{(l)}(i)) + (1 - e_n^{(l)}) e_{b,i}^{(l)} (1 - P_{R2}^{(l)}(i)) \tag{3.21}
$$

where $e_n^{(l)}$ represents the network loss probability measured in stream l. The packet-erasure probability e_i is defined as the probability that none of the descriptions of packet i arrives on time, and is given by

$$
e_i = \prod_{l=1}^{2} P_L^{(l)}(i) \tag{3.22}
$$

3.2.2 Joint FEC and Playout Control

The main attraction of multi-stream transmission arises from its flexibility in trading different sources of impairments against each other. Waiting for the arrival of both descriptions results in lower equipment impairment, but at the cost of higher delay impairment. On the other hand, playing out the voice description with lower delay avoids latency, but increases the equipment impairment. Since playout scheduling aims to improve the overall conversational speech quality, which hangs on the balance between delay and packet loss, full reconstruction of both descriptions may not always be the priority if the overall impairment does not justify the extra delay from waiting. Given that, the joint playout and FEC control must play around with switching between different playout scenarios in order to maximize the benefits of packet path diversity. To accomplish this goal, we formulated the system design as a perceptually motivated optimization problem and the adopted criterion relies on the use of the proposed multi-stream voice quality prediction model. Our efforts began by estimating the playout delay, which is defined as the time from the moment that packet is delivered to the network until it has to be played out. We applied an autoregressive algorithm (Moon et al. 1998) to estimate the mean $\hat{d}$ and variance $\hat{v}$ of network delay, and use them to calculate the buffer delay $d_b = \hat{d} + \beta\hat{v}$. Waiting for the FEC check packets results in additional delay and, consequently, the playout delay is given by

$$d_{play} = \hat{d} + \beta\hat{v} + (N-1)T_p \tag{3.23}$$

where T_p is the packet generation interval. The parameter β has a critical impact on the tradeoff between delay and late packet loss, which in turn influences the conversational speech quality. From (3.23) it can be deduced that increasing β leads to lower late loss rate as more packets arrive in time, and yet the end-to-end delay also increases. Most playout buffer algorithms [11][12][13] used a fixed value of β; e.g., $\beta = 4$, to set the buffer size, so that only a small fraction of the arriving packets should be lost due to late arrival. In this work, a β-adaptive algorithm is instead used to control the buffer size so that the reconstructed voice quality is maximized in terms of delay and loss.

Our general problem can be stated as follows: Given estimates of the parameters characterizing the packet loss and delay distribution, find the optimal values of β and $\{N, K\}$ so as to minimize the overall impairment function subject to the rate constraint. Let d_i be the end-to-end delay experienced by the ith packet, which consists of encoding delay d_c and playout delay d_{play}. Now, we define an overall impairment function I_m as a function of both d_i and $e_1^K = (e_1, \cdots, e_K)$ with the following form

$$I_m(d_i, e_1^K) = \; I_d(d_i) + \tfrac{1}{K}\textstyle\sum_{j=1}^{K}\sum_{l=1,2} r_l I_{e,l}(e_j) \tag{3.24}$$

where $r_1 + r_2 = 1$ and the probability to receive both descriptions is given by

$$r_2 = \frac{1}{1-e_i}\prod_{l=1}^{2}(1 - P_L^{(l)}(i)). \tag{3.25}$$

Our optimization framework requires an analytic expression for the packet erasure probability e_i as a function of the parameter β. Notice that $e_{b,i}^{(l)}$ and the playout delay d_{play} are strongly correlated, and to find out their relationship, the network delays of stream l are assumed to follow a Pareto distribution which is defined as $F_D^{(l)}(d) = 1-(g_l/d)^{\alpha_l}$. The parameters of Pareto distribution α_l and g_l can be estimated from past recorded delays using the maximum likelihood estimation method [13]. More specifically, given a set of past network delays $\{n_{i-1}^{(l)}, n_{i-2}^{(l)}, \ldots, n_{i-M}^{(l)}\}$, we compute $g_l = min\{n_{i-1}^{(l)}, n_{i-2}^{(l)}, \ldots, n_{i-M}^{(l)}\}$ and $\alpha_l = M/\Sigma_{j=i-1}^{i-M}\log(\frac{n_j^{(l)}}{g_l})$. Then, the late loss probability of packet i in stream l can be computed as follows:

$$e_{b,i}^{(l)} = 1 - F_D^{(l)}(D_{F,i}) = (g_l/D_{F,i})^{\alpha_l}. \tag{3.26}$$

where $D_{F,i} = d_{play} - (i-1)T_p$.This reduces the expression of the packet-erasure probability e_i to be a function of the playout delay d_{play}, which in turn is a function of the parameter β.

Finally, we summarize the proposed multi-stream joint playout and FEC adjustment algorithm as below.

1. Apply an autoregressive algorithm [11] to estimate the delay mean $\hat{d}_i^{(l)}$ and variance $\hat{v}_i^{(l)}$ for individual stream l ($l = 1, 2$) as follows:

   $$\hat{d}_i^{(l)} = \mu \hat{d}_{i-1}^{(l)} + (1-\mu) n_i^{(l)}. \tag{3.27}$$

   $$\hat{v}_i^{(l)} = \mu \hat{v}_{i-1}^{(l)} + (1-\mu) |n_i^{(l)} - \hat{d}_i^{(l)}|. \tag{3.28}$$

   where $n_i^{(l)}$ is the network delay of packet i in stream l and $\mu = 0.998002$ is a weighting factor for convergence control.

2. At the beginning of each talkspurt, update network delay records for the past $M = 200$ packets in every stream l ($l = 1, 2$), and use them to calculate the Pareto distribution parameters (α_l, g_l) by the maximum likelihood estimation method.

3. Use the values of (α_l, g_l) to compute the late loss probability in (3.26) and the packet erasure probability e_i in (3.22). Apply an exhaustive search method to determine the minimizer $(\hat{\beta}_i^{(l)}, \hat{N}^{(l)}, \hat{K}^{(l)})$ of the overall impairment function in (3.24) subject to the code rate constraint $\frac{N}{K} \times \frac{9.2}{8} \leq R_{max}$. Here, the maximum overall code rate R_{max} is chosen to be 2.

4. Set the playout delay and RS code parameters to

   $$\begin{aligned} d_{play} &= \hat{d}^{(l^*)} + \hat{\beta}_i^{(l^*)} \hat{v}^{(l^*)} + (\hat{N}^{(l^*)} - 1) T_p, \\ (N, K) &= (\hat{N}^{(l^*)}, \hat{K}^{(l^*)}) \end{aligned} \tag{3.29}$$

   with $l^* = \arg\min\{I_m(\hat{\beta}^{(l)}, \hat{N}^{(l)}, \hat{K}^{(l)}), l = 1, 2\}$

3.2.3 Experimental Results

Computer simulations were carried out to evaluate the performances given by the four MD voice transmission schemes, MD1-4, which all used the MD-G.729 for source coding and RS(N, K) code for channel coding. The speech data fed into the simulations were two sentential utterances spoken by one male and one female, each sampled at

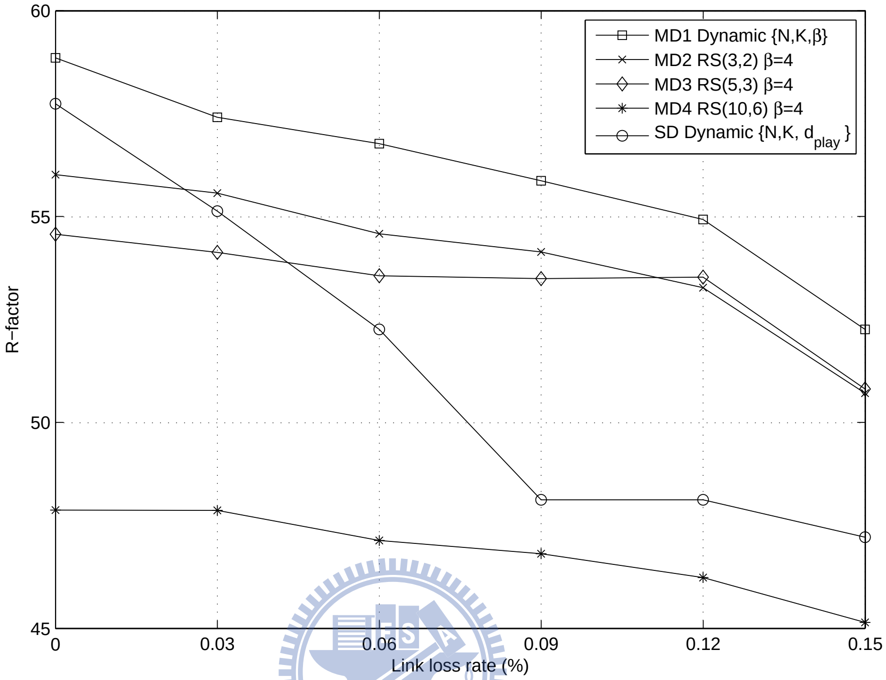

Figure 3.3: Performance comparison for different playout algorithms.

8 kHz and 8 seconds in duration. Both samples were encoded and then processed in accordance with the delay and loss characteristics of the trace data to degrade the speech. Among the four schemes, MD1 had its parameters $\{\beta, N, K\}$ dynamically adjusted according to the proposed voice quality prediction model, while MD2-4 shared a fixed $\beta = 4$ with (N, K) set at (3,2), (5,3), and (10,6) respectively. It should be pointed out that the last two (N, K) sets allowed MD3 and MD4 to perform at the same FEC coding ratio but with different lengths of delay, which gave us the opportunity to evaluate in our test environment the effect of packet loss vs. delay. It was hypothesized that the performances of these schemes would be set apart mainly by the values of $\{\beta, N, K\}$ they each assumed, and that the best performance should come with the adaptive parameter adjustment scheme, or MD1 in the current case, whose calculation was based on link loss, packet-erasure loss and various transmission scenarios.

The performances of MD transmission schemes were also compared with an FEC-protected single description (SD) transmission scheme, which consists of an 8 kbps G.729 speech coder followed by a RS(N, K) channel coder. Following the work of [16], the SD scheme applied a joint playout buffer and FEC adjustment scheme which jointly chooses both the playout delay d_{play} and the FEC scheme RS(N, K) so as to maximize the perceived voice quality. Figure 3.3 plots the perceived speech quality associated with the SD and four MD schemes for the case where the network paths are subjected to Gilbert-model loss process with link loss rate ranging from 0% to 15%. As described in Section 2.3, the perceived quality was gauged by calculating the predicted average R-factor according to the E-model. It can be seen that the R-factor was decreased as the link loss rate was increased regardless of the scheme used. When applying a joint playout buffer and FEC control scheme, the results obtained using the MD1 is clearly demonstrated an improvement over those obtained using the SD scheme, especially at high link loss rates. As link loss rates slightly beyond 6%, the SD scheme, despite its FEC feature, started showing incapability of recovering the lost packets in facing Gilbert-model link loss process. Among the four MD schemes, MD4, with the longest end-to-end delay, yielded the lowest R-factors, while MD3, with the same FEC coding ratio but shorter delays than those set for MD4, yielded higher R-factors than MD4, but lower R-factors than MD2. MD2 with the lower delay impairment allowed it to outperform MD3 and MD4, but its strength of packet recovery, as seen in Figure 3.3, receded faster as the link loss rate was increased, and at link loss rates greater than 12%, yielded lower R-factors than MD3. The best results in the plot, as hypothesized, were obtained with the currently proposed scheme MD1. Table 3.3 presents some of the varying parameters that shaped its performance and demonstrates the dynamic aspects of this scheme. At link loss rate = 12%, 10.25% of the descriptions were recovered with $(N, K) = (5, 3)$ while 89.7% (the rest) were recovered with $(N, K) = (3, 2)$; when the loss was increased to 15%, 25.6% of the descriptions were recovered with $(N, K) = (5, 3)$ and 74.35% (the rest) were recovered with $(N, K) = (3, 2)$. The average redundant bits thus obtained at the two link loss rates were $1.14 (= 10.25\% \cdot 2 + 89.7\% \cdot 1)$ and $1.255 (= 25.6\% \cdot 2 + 74.35\% \cdot 1)$, respectively. The plot showed that these settings allowed MD1 to outperform schemes with fixed settings in view of the transmission scenarios during testing. It follows that in multi-stream voice transmission scheme design, the pursuit of high performance of FEC does not guarantee high perceptual speech quality if delay fails to be jointly considered. The best performance seen in MD1 should therefore be taken as evidence attesting to the supremacy of using an all encompassing algorithm proposed here that aims to lower the total impairment impacts by making adjustments adaptive to the on-going interplay of delay, packet-erasure loss and various transmission scenarios.

Table 3.3: Average redundant bits comparison for different link loss rates.

Link loss rate %	RS(3,2)	RS(5,3)	Average redundant bits
12	89.7%	10.25%	1.14
15	74.35%	25.6%	1.255

Chapter 4

Iterative Symbol Decoding of Convolutionally Encoded Multiple Descriptions

In this chapter, we develop a new MD-ISCD technique which allows to exploit the source residual redundancy as well as the inter-description correlation to the fullest extent. The proposed MD system model is presented in Section 4.2. In Section 4.3, a log-MAP symbol decoding scheme is proposed to decode binary convolutional codes and is shown to be superior to the bit-level BCJR algorithm. Performance of the MD-ISCD is further enhanced by exchanging between its constituent decoders the whole symbol extrinsic information. Also a proposed in Section 4.4 is a joint MAP source decoder which processes the total channel outputs and combines them with inter-description correlation to improve the estimation of transmitted quantizer indexes. Computer simulations were conducted to compare various iterative decoders for transmission of convolutionally encoded MDSQ data over AWGN channels.

4.1 Introduction

With the rapid development of wireless multimedia communications, reliable transmission of speech and video signals over bandlimited noisy channels are becoming more and more widespread. MD coding [21] is a method of representing a source with multiple correlated descriptions such that any subset of the descriptions can be used to decode the source with a fidelity that increases with the number of received descriptions. The output symbols of an MD encoder exhibit considerable residual redundancy in terms of both nonuniformity of distribution and their dependencies. This redundancy is due to the nonoptimality of the practically designed source encoder in presence of complexity and delay constraints, or by path diversity as a result of MD coding. The ability to exploit path diversity and source residual redundancy for error robustness makes MD coding an attractive option for the multimedia transmission over unreliable IP networks. A typical example is the MD scalar quantization (MDSQ) [21]-[22] which splits source samples into two descriptions by using scalar quantizer and index assignment. The index assignment can be represented by a mapping of each reproduction level of the scalar quantizer to a unique element in an index assignment matrix. The choice of the index assignment matrix determines the correlation between the descriptions and is key to realize an MDSQ. Design algorithms for good index assignments are presented in [22]. In these, the inter-description correlation is controlled by choosing the number of diagonals covered by the index assignment. The MDSQ has been extensively studied for noiseless channels with packet loss, assuming that there exists multiple independent channels that either provide error-free transmission or experience total failure. In many practical situations, however, multiple descriptions of the source signals are transmitted over channels subject to noise and packet loss.

Transmission of convolutionally encoded multiple descriptions (MD) over noisy channels can benefit from the use of iterative source-channel decoding (ISCD) methods [21][22]. In MD-ISCD schemes, the iterative decoder consisted of two constituent decoders which exploited alternatively the source residual redundancy and channel-code redundancy according to the turbo-principle. With respect to an implementation of MD-ISCD, it has to be emphasized that the major part of the iterative decoding process runs on bit-level [3], but the source decoder itself is realized on symbol-level. For the purpose of applicability, the conversion of the symbol-probabilities to bit-probabilities is required in each passing of the extrinsic information between the source and channel decoders. This processing step destroys the mutual information between the symbol-bits, thus reducing the effectiveness of iterative decoding. In this chapter, we attempt to capitalize more fully on the source residual redundancy and then develop an MD-ISCD scheme which permits to exchange between its two constituent decoders the whole symbol extrinsic information. The first step toward realization is to derive the modified BCJR algorithm based on sectionalized code trellises that provides reliability information on each transmitted symbol rather than on index-bits. To further reduce the computation, we also apply the concept of Jacobian logarithm to formulate the algorithm in the logarithmic domain.

Also proposed in this chapter is a recursive implementation for the MD-SISO source decoder, which combines reliability information received on different channels and exploits the inter-description correlation to improve the estimation of the transmitted symbols. The first algorithmic step consists in the computation of the a posteriori probabilities (APP) for each of possibly transmitted quantizer index. In the second step, identical for both descriptions, these index APPs have to be combined with a priori knowledge of the index assignment to extract the extrinsic information on every symbol of each description. The extrinsic information contains the new part of information obtained from MD source coding and will be passed to the corresponding channel decoder as new a priori information for the next iteration. Experimental results indicate that the combined use of a symbol-based channel decoder and a joint MAP source decoder allows the proposed MD-ISCD scheme to achieve high robustness against channel noises.

4.2 Multi-Stream Voice Transmission over a Noisy Channel

The multi-channel transmission of autocorrelated sources over AWGN channels is considered, in which an MDSQ is used for source coding and convolutional codes are used for channel coding of individual descriptions. Figure 4.1 shows our model of a multi-channel transmission system. The MDSQ encoder can be decomposed into a scalar quantizer followed by an index assignment from [10]. Suppose at time t, the input sample v_t is quantized by the M-bit index u_t that, after index assignment, is represented by two descriptions $u_{I,t} = \delta_I(u_t)$ and $u_{J,t} = \delta_J(u_t)$ at an average rate of R bits per symbol per channel. For the scalar quantizer, the reproduction level corresponding to an index $u_t = l$ is denoted by c_l, where $l \in \Gamma = \{0, 1, \ldots, 2^M - 1\}$. We can generally assume that there is a certain amount of residual redundancy remaining in the index sequence due to delay and complexity constraints for the quantization stage. In the following, the time-correlations of quantizer indexes are modelled by a first-order stationary Markov process with index-transition probabilities $P(u_t|u_{t-1})$. The two descriptions resulting from MDSQ can be interpreted as the row and column indexes of an $2^R \times 2^R$ matrix, in which the $(\delta_I(l), \delta_J(l))$-th location is placed with a specified quantizer index $u_t = l$. Since $2^M < 2^{2R}$, these two descriptions contain redundancy and the correlation properties of each possible pair $(\delta_I(l), \delta_J(l))$ can be computed from the knowledge of index assignment. The amount of inter-description correlation decreases as M becomes larger and more diagonals are occupied by the quantizer indexes. In describing the index assignment matrix, let $R_k = \{l|\delta_I(l) = k\}$ and $C_m = \{l|\delta_J(l) = m\}$ represent the subset of quantizer indexes located in row k and in column m of the matrix, respectively. We will denote the output symbols of MDSQ by $u_{D,t}$, where for simplicity, $D \in \{I, J\}$ stands for one of the descriptions. After MDSQ encoding a block of T symbols of description D, written as $U_{D,1}^T = (u_{D,1}, \ldots, u_{D,t}, \ldots, u_{D,T})$, are interleaved by a symbol interleaver Φ. The interleaved symbol sequence, denoted by

1896

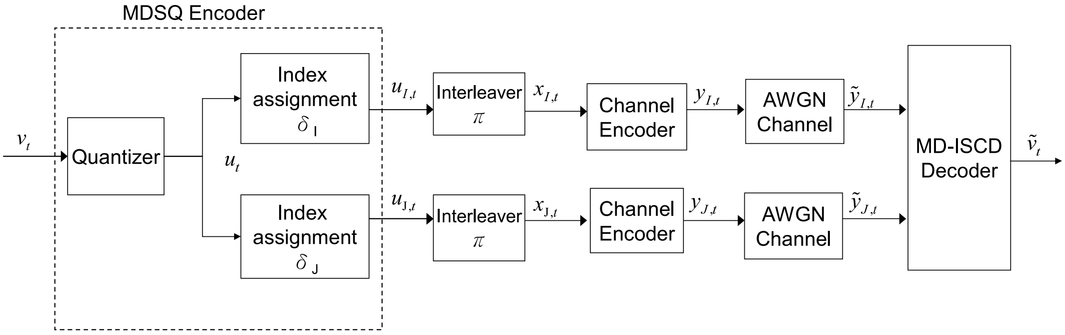

Figure 4.1: Block diagram of a two-channel MD communication system.

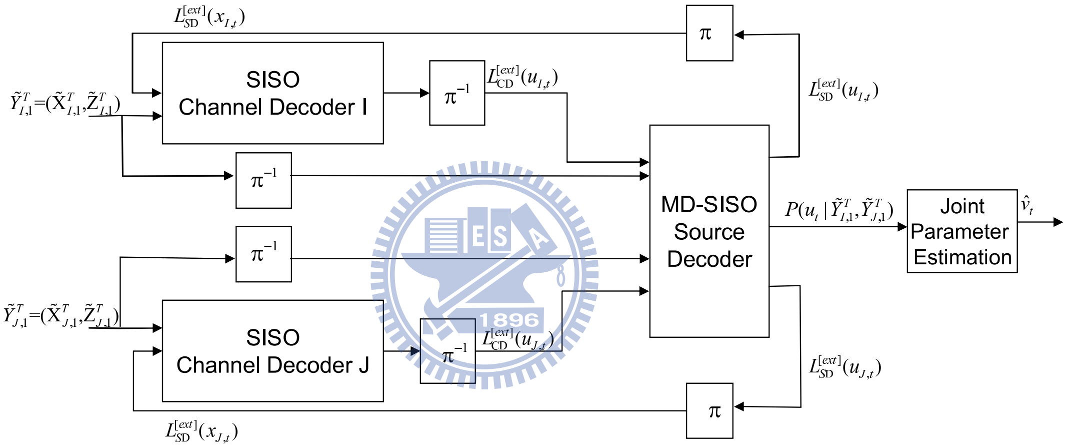
$L_{SD}^{[ext]}(x_{I,t})$
$\tilde{Y}_{I,1}^T=(\tilde{X}_{I,1}^T,\tilde{Z}_{I,1}^T)$
SISO Channel Decoder I
π^{-1}
$L_{CD}^{[ext]}(u_{I,t})$
π
$L_{SD}^{[ext]}(u_{I,t})$
π^{-1}
π^{-1}
MD-SISO Source Decoder
$P(u_t\,|\,\tilde{Y}_{I,1}^T,\tilde{Y}_{J,1}^T)$
Joint Parameter Estimation
$\hat{v}_t$
$\tilde{Y}_{J,1}^T=(\tilde{X}_{J,1}^T,\tilde{Z}_{J,1}^T)$
SISO Channel Decoder J
π^{-1}
$L_{CD}^{[ext]}(u_{J,t})$
$L_{SD}^{[ext]}(u_{J,t})$
π
$L_{SD}^{[ext]}(x_{J,t})$

Figure 4.2: MD-ISCD scheme for the concatenation of MDSQ and convolutional codes.

$X_{D,1}^T = (x_{D,1}, \ldots, x_{D,t}, \ldots, x_{D,T})$, is then processed by a binary convolutional channel encoder with a code rate of 1/2. If a systematic channel encoder is used, the codeword corresponding to each symbol $x_{D,t}$ can be written as $y_{D,t} = \{x_{D,t}, z_{D,t}\}$, where $x_{D,t}$ and $z_{D,t}$ represent the systematic and parity symbol of the code, respectively. The code sequences are modulated with a BPSK modulator and then transmitted over an AWGN channel. For brevity, denote the input and output sequence of the AWGN channel by $Y_{D,1}^T = \{X_{D,1}^T, Z_{D,1}^T\}$ and $\tilde{Y}_{D,1}^T = \{\tilde{X}_{D,1}^T, \tilde{Z}_{D,1}^T\}$, respectively.

Goal of the MD-ISCD is to jointly exploit the channel information and source *a priori* information for improved estimation of the transmitted quantizer index. When MDSQ is concatenated with a channel coder, the turbo-like evaluation of source residual redundancy and of artificial channel-code redundancy makes step-wise quality gains possible by iterative decoding. As shown in Figure 4.2, the receiver consists of two separate channel decoders and an MD source decoder with soft-inputs and soft-outputs (SISO). In the first step, identical for both descriptions, a channel decoder processes the received code sequence $\tilde{Y}_{D,1}^{T}$ and combines them with source *a priori* information to compute the extrinsic information $L_{CD}^{[ext]}(x_{D,t})$ on individual systematic symbol $x_{D,t}$. The MD-SISO source decoder combines the extrinsic information provided by the two channel decoders to provide a more accurate estimate of the transmitted quantizer index. Joint decoding of the two descriptions produces the *a posteriori* probability (APP) for each of possibly transmitted quantizer index u_t, which is denoted by $P(u_t|\tilde{Y}_{I,1}^{T}, \tilde{Y}_{J,1}^{T})$. Given the knowledge of the index assignment and source encoder statistics, it also generates a symbol-level extrinsic information $L_{SD}^{[ext]}(u_{D,t})$ for each description which will be used as *a priori* information of the corresponding channel decoder in the next iteration. Exchanging extrinsic information between the source and channel decoders is iteratively repeated until the reliability gain becomes insignificant. After the last iteration, the index APPs are used to determine the MAP signal estimates as follows:

$$\hat{v}_t = c_{l^*}, \ l^* = \max_{l \in \Gamma} P(u_t = l|\tilde{Y}_{I,1}^{T}\tilde{Y}_{J,1}^{T}) \tag{4.1}$$

4.3 Symbol Decoding of Binary Convolutional Codes

For the transmission scheme with channel coding, a soft-output channel decoder can be used to provide both decoded bits and their reliability information for further processing to improve the system performance. The commonly used BCJR algorithm is a trellis-based MAP decoding algorithm for both linear block and convolutional codes. The derivation presented in [23] led to a forward-backward recursive computation on the basis of a bit-level trellis diagram, which has two branches leaving each state and every branch represents a single symbol-bit. Proper sectionalization of a bit-level code trellis may result in useful trellis structural properties [31],[32] and allows us to devise SISO channel decoding algorithms which incorporate parameter-oriented extrinsic information from the source decoder. To proceed with this, we propose a modified BCJR algorithm which parses the received code-bit sequence into blocks and computes the APP for each parameter index on a symbol-by-symbol basis. Unlike classical BCJR algorithm that decodes one bit at a time, our scheme proceeds with decoding the parameter indexes as nonbinary symbols that are matched to the number of bits in an index. By parsing the code-bit sequence into M-bit symbols, we are in essence merging M stages of the original bit-level code trellis into one. As an example, we illustrate in Figure 4.3 two stages of the bit-level trellis diagram of a rate 1/2 convolutional encoder with generator polynomial $(7,5)_8$. The solid lines and dashed lines correspond to the input bits of 0 and 1, respectively. Fig. 4.3 also shows the sectionalized trellis diagram when two stages of the original bit-level trellis are merged together. In general, there are 2^M branches leaving and entering each state in a M-stage merged trellis diagram. Having defined the trellis structure as such, there will be one symbol APP corresponding to each branch which represents a particular parameter symbol $x_{D,t}$. For convenience, we say that the sectionalized trellis diagram forms a finite-state machine defined by its state transition function $F_s(x_{D,t}, s_t)$ and output function $F_p(x_{D,t}, s_t)$. Specifically, the code-symbol associated with the branch from state s_t to state $s_{t+1} = F_s(x_{D,t}, s_t)$ can be written as $y_{D,t} = (x_{D,t}, z_{D,t})$, where $z_{D,t} = F_p(x_{D,t}, s_t)$ is the parity symbol given state s_t and systematic symbol $x_{D,t}$.

We next apply sectionalized code trellises to formulate a recursive implementation for computing the APP of a systematic symbol $x_{D,t}$, given the received code sequence $\tilde{Y}_{D,1}^{T} = \{\tilde{y}_{D,1}, \tilde{y}_{D,2}, ..., \tilde{y}_{D,T}\}$. Let $l_D = \delta_D(l)$ represent the symbol of description D corresponding to a specified quantizer index $x_t = l$. Taking the trellis state s_t into

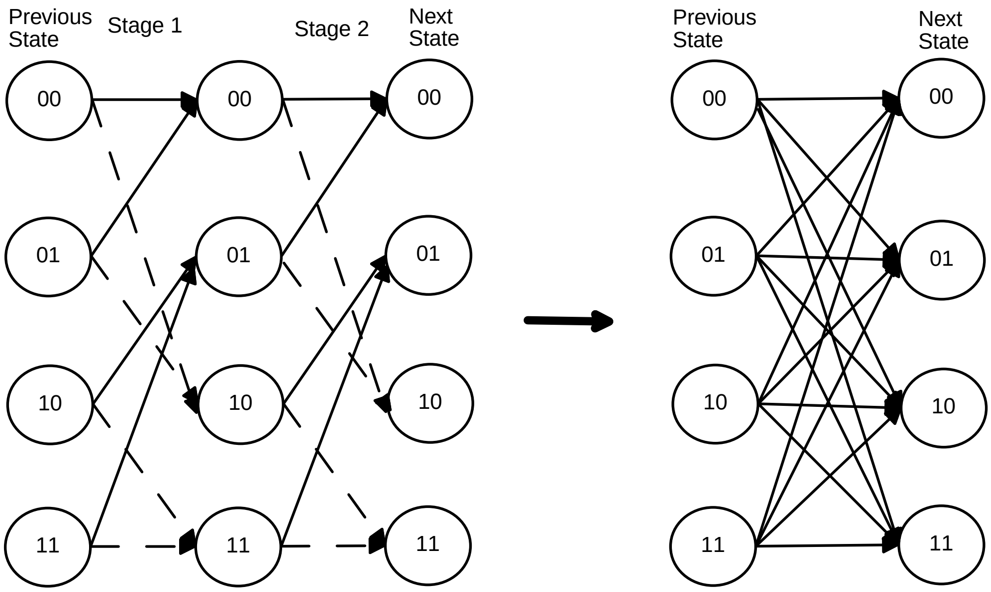

Figure 4.3: Bit-level and merged trellis diagrams

consideration, we rewrite the symbol APP as follows:

$$\begin{aligned} P(x_{D,t} = l_D|\tilde{Y}^T_{D,1}) &= C\sum_{s_t} P(x_{D,t} = l_D, s_t, \tilde{Y}^T_{D,1})/P(\tilde{Y}^T_{D,1}) \\ &= C\sum_{s_t} \alpha^x_t(l_D, s_t)\beta^x_t(l_D, s_t)/P(\tilde{Y}^T_{D,1}), \end{aligned} \tag{4.2}$$

where $\alpha^x_t(l_D, s_t) = P(x_{D,t} = l_D, s_t, \tilde{Y}^t_{D,1})$ and $\beta^x_t(l_D, s_t) = P(\tilde{Y}^T_{D,t+1}|x_{D,t} = l_D, s_t, \tilde{Y}^t_{D,1})$. For the recursive implementation, the forward and backward recursions are to compute the following metrics:

$$\begin{aligned} &\alpha^x_t(l_D, s_t) \\ &= \sum_{s_{t-1}}\sum_{k_D} P(x_{D,t} = l_D, s_t, x_{D,t-1} = k_D, s_{t-1}, \tilde{y}_{D,t}, \tilde{Y}^{t-1}_{D,1}) \\ &= \sum_{s_{t-1}}\sum_{k_D} \alpha^x_{t-1}(k_D, s_{t-1})\gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) \end{aligned} \tag{4.3}$$

$$
\begin{aligned}
&\beta_t^x(l_D, s_t) \\
&= \sum_{s_{t+1}} \sum_{k_D} P(\tilde{Y}_{D,t+1}^T, s_{t+1}, x_{D,t+1} = k_D | x_{D,t} = l_D, s_t, \tilde{Y}_{D,1}^t) \\
&= \sum_{s_{t+1}} \sum_{k_D} \beta_{t+1}^x(k_D, s_{t+1}) \gamma_{k_D, l_D}(\tilde{y}_{D,t+1}, s_{t+1}, s_t)
\end{aligned} \tag{4.4}
$$

and in (4.3)

$$
\begin{aligned}
\gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) &= P(x_{D,t} = l_D, s_t, \tilde{y}_{D,t} | x_{D,t-1} = k_D, s_{t-1}, \tilde{Y}_{D,1}^{t-1}) \\
&= P(s_t | x_{D,t-1} = k_D, s_{t-1}) P(x_{D,t} = l_D | x_{D,t-1} = k_D) \\
&\cdot P(\tilde{y}_{D,t} | x_{D,t} = l_D, s_t).
\end{aligned} \tag{4.5}
$$

The computation of the branch metric $\gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1})$ can be further simplified as follows. First, making use of the merged code trellis, the value of $P(s_t|x_{D,t-1} = k_D, s_{t-1})$ is either one or zero depending on whether symbol k_D is associated with transition from state s_{t-1} to state $s_t = F_s(x_{D,t-1} = k_D, s_{t-1})$. The second term in (4.5) is reduced to $P(x_{D,t} = l_D)$ under the assumption that $x_{D,t}$ is uncorrelated with $x_{D,t-1}$, which is indeed the case as $x_{D,t}$ is the interleaved version of symbols $u_{D,t}$. For AWGN channels, the third term in (4.5) can be computed by

$$
P(\tilde{y}_{D,t} | x_{D,t} = l_D, s_t) = P(\tilde{x}_{D,t} | x_{D,t} = l_D) P(\tilde{z}_{D,t} | z_{D,t} = F_p(x_{D,t} = l_D, s_t)). \tag{4.6}
$$

The MAP algorithm is likely to be considered too complex for real-time implementation in a practical system. To avoid the number of complicated operations and also numerical representation problems, realizations of the MAP algorithm in the logarithmic domain have been proposed in [33][34]. We define the reliability of each nonzero symbol $x_{D,t} = l_D$, $l_D = 1, 2, \ldots, 2^R - 1$, with respect to $x_{D,t} = 0$, by considering log-likelihood ratio (LLR) of the following type

$$
L(x_{D,t} = l_D | \tilde{Y}_{D,1}^T) = \log \tfrac{P(x_{D,t} = l_D | \tilde{Y}_{D,1}^T)}{P(x_{D,t} = 0 | \tilde{Y}_{D,1}^T)}. \tag{4.7}
$$

This definition for the LLR values allow for easy conversion between the *a posteriori* LLRs and APPs. The next step is to reduce the large computational burden complexity which is required for computing the logarithmic values of the $\alpha_t^x(l_D, s_t)$ and $\beta_t^x(l_D, s_t)$ terms in (4.2). This task can be accomplished by using the Jacobian logarithm function [34] defined by the property

$$\log(e^{\delta_1} + e^{\delta_2}) = \max\{\delta_1, \delta_2\} + \log(1 + e^{-|\delta_2 - \delta_1|}). \tag{4.8}$$

For brevity, we use the following shorthand notation $\max_j{}^*\{\delta_j\} = \log(\sum_{j=1}^{l} e^{\delta_j})$. By taking the logarithm of $\alpha_t^x(l_D, s_t)$ in (4.3), we have

$$\begin{aligned} \hat{\alpha}_t^x(l_D, s_t) &= \log \alpha_t^x(l_D, s_t) \\ &= \max_{s_{t-1}}{}^* \max_{k_D}{}^* \{\hat{\alpha}_{t-1}^x(k_D, s_{t-1}) + \hat{\gamma}_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1})\}\} \end{aligned} \tag{4.9}$$

and similarly,

$$\begin{aligned} \hat{\beta}_t^x(l_D, s_t) &= \log \beta_t^x(l_D, s_t) \\ &= \max_{s_{t+1}}{}^* \max_{k_D}{}^* \{\hat{\beta}_{t+1}^x(k_D, s_{t+1}) + \hat{\gamma}_{k_D,l_D}(\tilde{y}_{D,t+1}, s_{t+1}, s_t)\}\} \end{aligned} \tag{4.10}$$

$$\begin{aligned} \hat{\gamma}_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) &= \log \gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) \\ &= \log P(s_t | x_{D,t-1} = k_D, s_{t-1}) + \log P(x_{D,t} = l_D) \\ &\quad + \log P(\tilde{x}_{D,t} | x_{D,t} = l_D) + \log P(\tilde{z}_{D,t} | z_{D,t} = F_p(x_{D,t} = l_D, s_t)). \end{aligned} \tag{4.11}$$

An iterative process using the log-MAP channel decoder as a constituent decoder is realizable, if the *a posteriori* LLR $L(x_{D,t} = l_D | \tilde{Y}_{D,1}^T)$ can be separated into three additive terms: the *a priori* term $L_a(x_{D,t} = l_D) = \log[P(x_{D,t} = l_D)/P(x_{D,t} = 0)]$, the channel-related term $L_c(x_{D,t} = l_D) = \log[P(\tilde{x}_{D,t} | x_{D,t} = l_D)/P(\tilde{x}_{D,t} | x_{D,t} = 0)]$, and an extrinsic term $L_{CD}^{[ext]}(x_{D,t} = l_D)$. Substituting (4.9) and (4.11) into (4.7) leads to

$$L(x_{D,t} = l_D | \tilde{Y}_{D,1}^T) = L_a(x_{D,t} = l_D) + L_c(x_{D,t} = l_D) + L_{CD}^{[ext]}(x_{D,t} = l_D) \tag{4.12}$$

with the extrinsic LLR

$$\begin{aligned}
&L_{CD}^{[ext]}(x_{D,t}=l_D) \qquad (4.13)\\
&=\max_{s_t}{}^{*}\{\log P(\tilde{z}_{D,t}|x_{D,t}=l_D,s_t)+\beta_t^x(l_D,s_t)\\
&+\max_{s_{t-1}}{}^{*}\{\max_{k_D}{}^{*}\{\log P(s_t|x_{D,t-1}=k_D,s_{t-1})+\alpha_{t-1}^x(k_D,s_{t-1})\}\}\}\\
&-\max_{s_t}{}^{*}\{\log P(\tilde{z}_{D,t}|x_{D,t}=0,s_t)+\beta_t^x(0,s_t)\\
&+\max_{s_{t-1}}{}^{*}\{\max_{k_D}{}^{*}\{\log P(s_t|x_{D,t-1}=0,s_{t-1}).+\alpha_{t-1}^x(k_D,s_{t-1})\}\}\}.
\end{aligned}$$

Notice that the *a priori* LLR in (4.12) is initialized to be $L_a(x_{D,t}=l_D)$ in terms of the source distribution $P(x_{D,t}=l_D)$. Within iterations the precision of the APP estimation can be enhanced by replacing $L_a(x_{D,t}=l_D)$ with the interleaved extrinsic LLR $L_{SD}^{[ext]}(x_{D,t}=l_D)$ provided by the source decoder. Therefore, the extrinsic LLR resulting from the channel decoding can be calculated by

$$L_{CD}^{[ext]}(x_{D,t}=l_D)=L(x_{D,t}=l_D|\tilde{Y}_{D,1}^T)-L_{SD}^{[ext]}(x_{D,t}=l_D)-L_c(x_{D,t}=l_D) \qquad (4.14)$$

and used as new *a priori* information for the source decoder.

4.4 The MD-SISO Source Decoder

Goal of the MD-SISO source decoder is to compute the APPs of transmitted quantizer indexes by jointly exploiting the channel information, the source residual redundancy and the inter-description correlation induced by the MDSQ. In previous work related to this problem [22][35], the source decoder uses two separate MAP detectors with each detector operating on a single description $\tilde{Y}_{D,1}^T$ to compute the APP $P(u_{D,t}|\tilde{Y}_{D,1}^T)$ for a decoded systematic symbol $u_{D,t}=l_D$. Afterwards the source decoder makes an MAP decision on the two symbols $\{l_I^*,l_J^*\}$ and uses their combination to locate the corresponding quantizer index from the index assignment matrix. As the two MAP symbol estimates are decoded separately, it may report invalid codeword combinations corresponding to the empty cells of the index assignment matrix. To compensate for this shortage, we propose a joint MAP decoding algorithm which combines reliability information received on different channels and computes the APP for each of possibly transmitted quantizer index $u_t = l$. For the purpose of applicability, the algorithm of MD-SISO source decoding is separated into two parts. The first algorithmic step consists in the computation of the APP $P(u_t|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1})$ for a decoded quantizer index u_t, given the two received code-symbol sequences $\{\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}\}$. In the second step, identical for both descriptions, these index APPs are combined with *a priori* knowledge of the index assignment to extract the extrinsic information $L^{[ext]}_{SD}(u_{D,t})$ on every symbol $u_{D,t}$ of description D. It contains the new part of information resulting from MD-SISO source decoding and will be delivered back to the corresponding channel decoder as new *a priori* information for the next iteration.

The source decoding algorithm starts by computing the APP for a decoded quantizer index $u_t = l$ as follows

$$P(u_t = l|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}) = P(u_t = l, \tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1})/P(\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}). \tag{4.15}$$

Since the received sequence of systematic symbols are de-interleaved and then processed by the source decoder, we have $P(u_t = l, \tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}) = P(u_t = l, \tilde{U}^T_{I,1}, \tilde{Z}^T_{I,1}, \tilde{U}^T_{J,1}, \tilde{Z}^T_{J,1})$, where $\tilde{U}^T_{D,1} = \Phi^{-1}(\tilde{X}^T_{D,1})$ These probabilities can be further decomposed by using the Bayes theorem as

$$\begin{aligned} P(u_t = l, \tilde{U}^T_{I,1}, \tilde{Z}^T_{I,1}, \tilde{U}^T_{J,1}, \tilde{Z}^T_{J,1}) &= P(u_t = l, \tilde{U}^T_{I,1}, \tilde{U}^T_{J,1}) P(\tilde{Z}^T_{I,1}, \tilde{Z}^T_{J,1}|u_t = l, \tilde{U}^T_{I,1}, \tilde{U}^T_{J,1}) \\ &= \alpha^u_t(l)\beta^u_t(l) \cdot \textstyle\prod_{D\in\{I,J\}} P(\tilde{Z}^T_{D,1}|u_{D,t} = l_D, \tilde{U}^T_{D,1}) \end{aligned} \tag{4.16}$$

where $\alpha^u_t(l) = P(u_t = l, \tilde{U}^t_{I,1}, \tilde{U}^t_{J,1})$ and $\beta^u_t(l) = P(\tilde{U}^T_{I,t+1}, \tilde{U}^T_{J,t+1}|u_t = l, \tilde{U}^t_{I,1}, \tilde{U}^t_{J,1})$. Using the Markov property of the indexes and the memoryless assumption of the channel, the forward-backward recursions of the algorithm in the logarithmic domain can be expressed as

$$\begin{aligned}\hat{\alpha}_t^u(l) &= \log \alpha_t^u(l) \\ &= \log \sum_k P(u_t = l, u_{t-1} = k, \tilde{U}_{I,1}^t, \tilde{U}_{J,1}^t) \\ &= \log \sum_k P(\tilde{u}_{I,t}, \tilde{u}_{J,t} | u_t = l, u_{t-1} = k, \tilde{U}_{I,1}^{t-1}, \tilde{U}_{J,1}^{t-1}) \\ &\cdot P(u_t = l | u_{t-1} = k, \tilde{U}_{I,1}^{t-1}, \tilde{U}_{J,1}^{t-1}) P(u_{t-1} = k, \tilde{U}_{I,1}^{t-1}, \tilde{U}_{J,1}^{t-1}) \\ &= \max_k{}^* \{\hat{\gamma}_{l,k}^t(\tilde{u}_{I,t}, \tilde{u}_{J,t}) + \hat{\alpha}_{t-1}^u(k)\}\end{aligned} \tag{4.17}$$

$$\begin{aligned}\hat{\beta}_t^u(l) &= \log \beta_t^u(l) \\ &= \log \sum_k P(u_t = l, u_{t+1} = k, \tilde{U}_{I,1}^T, \tilde{U}_{J,1}^T) / P(u_t = l, \tilde{U}_{I,1}^t, \tilde{U}_{J,1}^t) \\ &= \max_k{}^* \{\hat{\gamma}_{k,l}^{t+1}(\tilde{u}_{I,t+1}, \tilde{u}_{J,t+1}) + \hat{\beta}_{t+1}^u(k)\}\end{aligned} \tag{4.18}$$

and in (4.17)

$$\hat{\gamma}_{l,k}^t(\tilde{u}_{I,t}, \tilde{u}_{J,t}) = \log P(\tilde{u}_{I,t} | u_{I,t} = l_I) + \log P(\tilde{u}_{J,t} | u_{J,t} = l_J) + \log P(u_t = l | u_{t-1} = k) \tag{4.19}$$

With these metrics, the *a posteriori* LLR corresponding to the index APP $P(u_t = l | \tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T)$ can be expressed as

$$\begin{aligned}L(u_t = l | \tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T) &= \hat{\alpha}_t^u(l) + \hat{\beta}_t^u(l) - \hat{\alpha}_t^u(0) - \hat{\beta}_t^u(0) \\ &\quad + \textstyle\sum_{D \in \{I,J\}} \{L_{CD}^{[ext]}(u_{D,t} = l_D) - L_{CD}^{[ext]}(u_{D,t} = 0_D)\}\end{aligned} \tag{4.20}$$

In the next step, the APP of each decoded symbol in every description is calculated from the temporary values of the index APPs and used for computing the extrinsic information of the source decoder. From the properties of the index assignment matrix, this task was accomplished by summing together the APPs of quantizer indexes being assigned to a certain description. For example, the APP for a decoded symbol $u_{I,t} = l_I$ of description I is given by

$$P(u_{I,t} = l_I | \tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T) = \sum_{n \in R_{l_I}} P(u_t = n | \tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T). \tag{4.21}$$

where $R_{l_I} = \{n|\delta_I(n) = l_I\}$ represents the subset of quantizer indexes located in column l_I of the matrix. Substituting (4.17) and (4.19) into (4.21) leads to

$$\begin{aligned}\log P(u_{I,t} = l_I|\tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T) &= \log P(\tilde{u}_{I,t}|u_{I,t} = l_I) + \log P(\tilde{Z}_{I,1}^T|u_{I,t} = l_I, \tilde{U}_{I,1}^T) \\ &+ \max_{n\in R_{l_I}}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = n_J) + \log P(\tilde{Z}_{J,1}^T|u_{J,t} = n_J, \tilde{U}_{J,1}^T) \\ &+ \hat{\beta}_t^u(n) + \max_k{}^*\{\log P(u_t = n|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\}.\end{aligned} \tag{4.22}$$

This allows us to decompose the *a posteriori* LLR $L(u_{I,t} = l_I|\tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T)$ into three additive terms: *a priori* term $L_a(u_{I,t})$, the channel-related term $L_c(u_{I,t} = l_I)$, and an extrinsic term $L_{SD}^{[ext]}(u_{I,t} = l_I)$. In order to determine each of the three terms, we rewrite (4.22) in log-likelihood algebra as

$$\begin{aligned}L(u_{I,t} = l_I|\tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T) &= \log \frac{\sum_{n\in R_{l_I}} P(u_t{=}n|\tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T)}{\sum_{m\in R_0} P(u_t{=}m|\tilde{Y}_{I,1}^T, \tilde{Y}_{J,1}^T)} \\ &= L_a(u_{I,t} = l_I) + L_c(u_{I,t} = l_I) + L_{SD}^{[ext]}(u_{I,t} = l_I)\end{aligned}$$

where

$$L_a(u_{I,t} = l_I) = \log[P(\tilde{Z}_{I,1}^T|u_{I,t} = l_I, \tilde{U}_{I,1}^T)/P(\tilde{Z}_{I,1}^T|u_{I,t} = 0, \tilde{U}_{I,1}^T)]$$

$$L_c(u_{I,t} = l_I) = \log[P(\tilde{u}_{I,t}|u_{I,t} = l_I)/P(\tilde{u}_{I,t}|u_{I,t} = 0)]$$

and

$$\begin{aligned}&L_{SD}^{[ext]}(u_{I,t} = l_I) \\ &= \max_{n\in R_{l_I}}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = n_J) + \log P(\tilde{Z}_{J,1}^T|u_{J,t} = n_J, \tilde{U}_{J,1}^T) \\ &+ \hat{\beta}_t^u(n) + \max_k{}^*\{\log P(u_t = n|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\} \\ &- \max_{m\in R_0}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = m_J) + \log P(\tilde{Z}_{J,1}^T|u_{J,t} = m_J, \tilde{U}_{J,1}^T) \\ &+ \hat{\beta}_t^u(m) + \max_k{}^*\{\log P(u_t = m|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\}\end{aligned}$$

As shown in the Appendix B, the *a priori* LLR in (4.24) is equal to the de-interleaved sequence of extrinsic information resulting from the channel decoding, i.e, $L_a(u_{I,t} = l_I) = L_{CD}^{[ext]}(u_{I,t} = l_I)$. The extrinsic LLR $L_{SD}^{[ext]}(u_{I,t} = l_I)$ contains the new part of information which has been determined by the source decoder by exploiting the residual source redundancy as well as the inter-description correlation induced by the

MDSQ. With respect to (4.23), the extrinsic LLR resulting from the source decoding can be calculated by

$$L^{[ext]}_{SD}(u_{I,t} = l_I) = L(u_{I,t} = l_I|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}) - L^{[ext]}_{CD}(u_{I,t} = l_I) - L_c(u_{I,t} = l_I) \qquad (4.23)$$

which is used after interleaving as *a priori* information in the next channel decoding round. Finally, we summarize the proposed MD-ISCD scheme as follows:

1. Initialization: Set the extrinsic information of source decoding to $L^{[ext]}_{SD}(x_{I,t}) = L^{[ext]}_{SD}(x_{J,t}) = 0$. Set the iteration counter to $n = 0$ and define an exit condition n_{max}.
2. Read series of received sequences $\tilde{Y}^T_{D,1}$ and map all received systematic symbols $\tilde{x}_{D,t}$ to channel-related LLR $L_c(x_{D,t})$.
3. Perform log-MAP channel decoding on each description to compute the extrinsic LLR $L^{[ext]}_{CD}(x_{D,t})$ using (4.14).
4. Perform MD-SISO source decoding by inserting the de-interleaved extrinsic LLR $L^{[ext]}_{CD}(u_{I,t})$ and $L^{[ext]}_{CD}(u_{J,t})$ into (4.20) to compute the index *a posteriori* LLR $L(u_t|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1})$ and into (4.23) to compute the symbol *a posteriori* LLR $L(u_{I,t}|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1})$. Then, the extrinsic LLR $L^{[ext]}_{SD}(u_{I,t})$ is computed by (4.27) and is forwarded to the channel decoder as *a priori* information. Joint decoding of the two received sequences to extract the extrinsic LLR $L^{[ext]}_{SD}(u_{J,t})$ of description J operates in a similar manner.
5. Increase the iteration counter $n \leftarrow n + 1$. If the exit condition $n = n_{max}$ is fulfilled, then continue with step 6, otherwise proceed with step 3.
6. Compute the APP for each decoded index $u_t = l$ as follows:

$$P(u_t = l|\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1}) = e^{L(u_t=l|\tilde{Y}^T_{I,1},\tilde{Y}^T_{J,1})} / \sum_{j=0}^{2^M-1} e^{L(u_t=j|\tilde{Y}^T_{I,1},\tilde{Y}^T_{J,1})}. \qquad (4.24)$$

7. Estimate the decoder output signals $\hat{v}_t$ by (4.1) using the index APPs obtained from step 6.

4.5 Experimental Results

Computer simulatoions were conducted to compare the performance of various MD-ISCD schemes for transmission of convolutionally encoded multiple descriptions over AWGN channels. First a bit-level iterative decoding scheme MD-ISCD1 [22] is considered for error mitigation using the classical BCJR algorithm for soft-output channel decoding and assisted with the bit reliability information provided by the soft-bit source decoding [36]. For the MD-ISCD1 scheme with bit interleaving, the source decoder applies two separate MAP detectors and performs turbo cross decoding to exploit the inter-description correlation [22]. Two approaches to symbol-level iterative decoding, denoted by MD-ISCD2 and MD-ISCD3, are presented and investigated. They both applied a symbol interleaver and performed log-MAP symbol decoding of binary convolutional codes based on sectionalized code trellises. Unlike the MD-ISCD1 and MD-ISCD2 which use two MAP detectors with each detector decoding one description, the MD-ISCD3 applies a joint MAP source decoder to improve the estimation of transmitted quantizer indexes by combining reliability information received on different channels. Specifically, the APP to be computed for the MD-ISCD3 is $P(u_t|\tilde{Y}_{I,1}^T, Y_{J,1}^T)$ in (4.14), and $\{P(u_{I,t}|\tilde{Y}_{I,1}^T), P(u_{J,t}|\tilde{Y}_{J,1}^T)\}$ for the other two schemes. The input signals considered here include are first order Gauss-Markov sources described by $v_t = \rho v_{t-1} + w_t$, where w_t is a zero-mean, unit-variance white Gaussian noise, with correlation coefficients of $\rho = 0.8$ and $\rho = 0.95$. As indicated in [37], a value of $\rho = 0.95$ can be found for scale factors determined in the MPEG audio codec for digital audio broadcasting. On the other hand, $\rho = 0.8$ provides a good fit to the long-time-averaged autocorrelation function of 8 kHZ-sampled speech that is bandpass-filtered to the range (300 Hz,3400 Hz) [38]. A total of 3000000 input samples is processed by a scalar M-bit Lloyd-Max quantizer and each quantizer index is mapped to two descriptions, each with R bits per symbol per channel. For each of the two descriptions, the bitstreams were spread by an interleaver of length 300 bits and afterwards they were channel encoded by a rate-1/2 recursive systematic convolutional code with a memory order 2 and generator

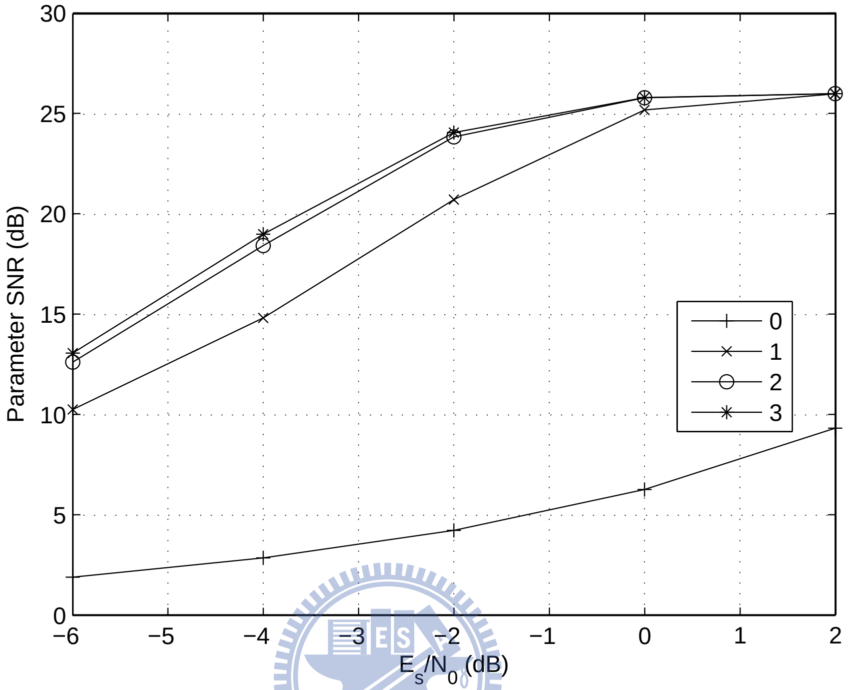

Figure 4.4: MD-ISCD3 performance for Gauss-Markov sources with $\rho = 0.95$ and $(M, R) = (5, 3)$.

polynomial $\mathbf{G}(D) = (1, (1 + D^2)/(1 + D + D^2))$.

A preliminary experiment was first performed to examine the step-wise quality gains due to the turbo-like evaluation of channel-code and source-code redundancies. The variation of parameter signal-to-noise ratio (SNR) as a function of the channel SNR E_s/N_0 for MD-ISCD3 simulation of Gauss-Markov sources with $\rho = 0.95$ and $(M, R) = (5, 3)$ is shown in Figure 4.4. The results indicate that a turbo-like refinement of the extrinsic information from both constituent decoders makes substantial quality improvements possible. The full gain in parameter SNR is reached after three iterations. The investigation further showed that the improved performance achievable using MD-ISCD3 is more noticeable for lower channel SNR. To elaborate further, SNR performances of various MD-ISCD schemes were examined for Gauss-Markov sources with $\rho = 0.8$ and $\rho = 0.95$. We provide results for experiments on MDSQ with

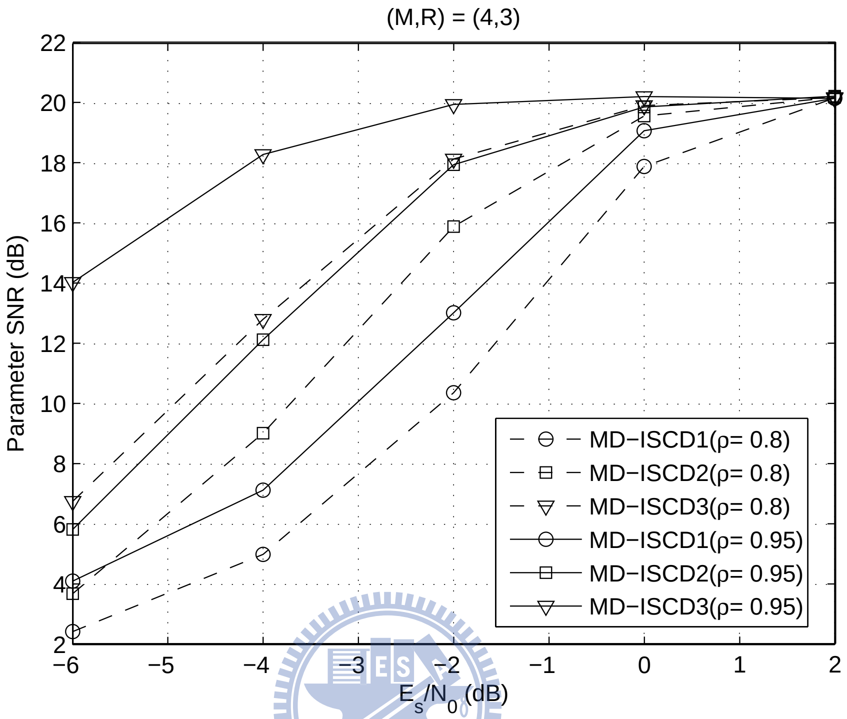

Figure 4.5: SNR performance of different decoders for $(M, R) = (4, 3)$ and Gauss-Markov sources ($\rho = 0.8, 0.95$)

$(M, R) = (4, 3)$ and $(5, 3)$ in Figures. 4.5 and 4.6, respectively. Note that the value of maximal PSNR is 20.22 dB for $M = 4$ and 26.01 dB for $M = 5$ due to Lloyd-Max quantization distortion. Three iterations of the algorithm were performed by each decoder as further iterations did not result in a significant improvement. The results clearly demonstrate the improved performance achievable using symbol decoders MD-ISCD2 and MD-ISCD3 in comparison to that of bit-based MD-ISCD1. Furthermore, the improvement has a tendency to increase for lower channel SNR and for more heavily correlated Gaussian sources. This indicates that the extrinsic information between source and channel decoders is better to be exploited at the symbol level. The investigation further showed that there is a considerable gap between the MD-ISCD2 and MD-ISCD3 schemes. Moreover, the performance gain achievable using MD-ISCD3 increases as more and more diagonals are included in the index assignment. For the case

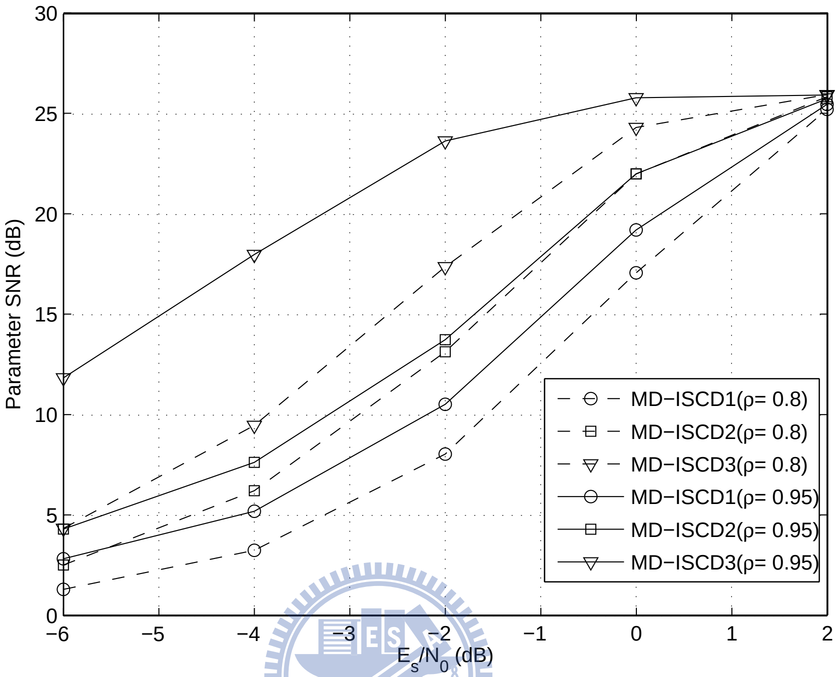

Figure 4.6: SNR performance of different decoders for $(M, R) = (5, 3)$ and Gauss-Markov sources ($\rho = 0.8, 0.95$)

of $(M, R) = (4, 3)$ and $\rho = 0.95$, the MD-ISCD3 yields about 1.97 dB improvement at $E_b/N_0 = -2$ dB relative to the MD-ISCD2. For the case of $(M, R) = (5, 3)$, the parameter SNR can further be improved by up to 9.91 dB. The difference between them is due to the fact that MD-ISCD2 only accounts for the information received on a single description through the knowledge of symbol APP $P(u_{D,t}|\tilde{Y}^T_{D,1})$. On the other hand, the MD-ISCD3 uses in its APP computation the total channel outputs $\{\tilde{Y}^T_{I,1}, \tilde{Y}^T_{J,1})$ and makes the final decision by incorporating the inter-description correlation as a result of the MDSQ.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this study, we addressed two specific problems in multiple description coding: iterative source-channel decoding and joint playout and FEC control. The Goal of the first problem was to develop an alternative to bit-level multiple description iterative soure-channel decoding(MD-ISCD) scheme . For the bit-level MD-ISCD scheme using the classical BCJR algorithm, the source decoder applies two separate MAP detectors and performs turbo cross decoding to exploit the inter-description correlation[22]. To enhance the error-robustness performance, we investigated the use of a symbol-level MD-ISCD scheme, which avoids the problem of converting between bit-level and symbol-level extrinsic information and therefore allows us to exchange between the source and channel decoders the whole symbol extrinsic information. The first step toward realization is to use sectionalized code trellises rather than bit-level trellises as the bases for soft-output channel decoding of binary convolutional codes. A log-MAP symbol decoding scheme is proposed to decode binary convolutional codes and is shown to be superior to the bit-level BCJR algorithm. Performance is further improved by using a joint MAP source decoder that processes reliability information received on different channels and combines them with inter-description correlation to provide a better estimate of the transmitted quantizer index. The major advantage of the proposed symbol-based MD-ISCD is to allow us to exploit the source residual redundancy as well as the inter-description correlation to the fullest extent. Experimental results indicate that the combined use of a symbol-based channel decoder and a joint MAP source decoder allows the proposed MD-ISCD scheme to achieve high robustness against channel noises.

The second problem we addressed is the perceptually motivated optimization criterion used for joint playout and FEC control in a two-channel transmission system. The commonly used E-model [18] may only suit single-path transmission with two conceivable playout scenarios; i.e., total loss vs. no-loss of packets. A third scenario, partial loss, however, would rise with MD transmission. To better predict the multi-stream voice quality, we used the MD-G.729 scheme described in [4] to generate two descriptions and developed an objective method for prediction of impairment factor $I_{e,k}$ regression model using PESQ algorithm[25]. Based on the new prediction model, we proposed the use of minimum overall impairment as a criterion for perceptual-based multi-stream playout buffer algorithm. As a further step toward perceptual optimization, the error concealing capabilities of MD can be improved by including an forward error control(FEC) mechanism. Traditionally, the study of FEC for loss recovery and playout buffer adaptation for jitter compensation have proceeded independently. By taking advantage of the linking between FEC and playout buffer, we proposed a perceptually motivated optimization criterion and a practically feasible new algorithm for multi-stream voice transmission. We start by considering the perceived voice quality as a function of playout scenario, the packet erasure rate and the end-to-end delay. Adaptive joint playout buffer and FEC adjustment is then formulated as an optimization problem leading to the minimum overall impairment. Experimental results show that the proposed multi-stream voice transmission scheme can achieve a better delay-loss tradeoff and thereby improves the perceived speech quality.

5.2 Future Work

The proposed solutions to MD-ISCD scheme can be regarded as the first step in multi-channel transmission of fixed-length encoded source signals. A number of crucial issues must be further addressed with the goal toward multi-channel transmission of variable-length codec(VLC) encoded source signals. The directions of future work will focus on the generalization of the proposed MD scheme to VLC. This section briefly outlines some directions of future work.

With the rapid development of wireless and multimedia communications, reliable transmission of multimedia signals over band-limited noisy channels are becoming more and more widespread. Source coder is used to extract characteristic parameters of the source signals and a channel coder plays a role in the error protection. Traditionally, the study of source coder for data compression and channel coder for error concealment have developed independently. VLCs are widely employed in state-of-the-art compression standards of audio, image and video. The advantage of VLC is that the source can be encoded efficiently with the average maximal compression rate but is very sensitive to channel noise. A conventional VLC decoder uses the prefix property of VLC to decode the received bit sequence ,which is usually processed hard decisions have been made. A better alternative [39][40] proposed the use of modified Viterbit algorithm based on bit-level VLC trellis to decode received bit sequence. All the possible received bit sequences can be described in a VLC trellis, since a VLC encoder can be viewed as a finite- state machine. Recently, some joint source-channel coding schemes for VLC have gained much interests since they are more advantageous than the traditional separation of source coder and channel coder. In attempts to improve the error robustness, a number of VLCs were proposed by adding extra redundancy including variable length error correcting codes[41][42] and reversible variable length codes (RVLC)[43]. Soft decoding of these VLCs can be decoded by exploiting source a priori information as well as error correcting redundancy.

5.2.1 MD-VLC Design

Researches[4][44][45] proposed some MD schemes to further improve the bandwidth efficiency. In [4], two proposed MD speech coders generate multiple description with fixed length codes. A MDSQ based communication for VLC scheme can be found in [44][45]. A typical example is the MD-VLC encoder which consists of a SQ followed by an index assignment and VLC. The source signals are quantized by scalar quantizer and each index is then mapped to VLC-encoded multiple descriptions. Traditionally, the VLC is designed for the SD transmission scheme. In [43], a new approch was proposed to construct the RVLC which satisfies both the prefix and suffix conditions under the constraint that free distance is not less than 2. The first direction in future work is to design a MD-VLC construction algorithm based on a criterion which takes free distance and path diversity into account in order to increase bandwidth efficiency as well as the error-correcting capability.

5.2.2 Iterative Decoding of Convolutionally-Encoded MD-VLC

The next issue to be addressed is the iterative source-channel decoding approach for MD-VLC encoded signals to further increase error robustness. Previous works toward iterative source-channel decoding of VLC encoded for SD transmission can be found in [46], but their works are expected to show two limitations. Firstly, their iterative decoding algorithm may be only applies for SD transmission, without taking into account the inter-description redundancy coming from the MD-VLC system. Secondly, the major part of decoding process runs on bit-level, but the source decoder itself is realized on symbol-level. For the propose of applicability, the conversion between bit and symbol probability destroys the bit-correlations within a symbol, thus reducing the effectiveness of iterative decoding. While we have developed a symbol-level BCJR algorithm, the concept of sectionalized trellis diagram can only be applied to fixed-length encoded symbols. In the case of VLC-related state transitions, different paths entering a state may consist of different combinations of the bit-vectors.

Recognizing this, the second direction in future work is to develop a symbol-by-symbol iterative source-channel decoding for MD-VLC encoded signals. In order to realize this approach, we first need to construct a symbol-level VLC trellis representation extended from [47] which will be used by channel decoding. Based on this symbol-level VLC trellis, the next step is to derive a variable-length symbol-level BCJR algorithm. In the following sections, an example showing the symbol-level VLC trellis for SD transmission system will be given. We will give a detail derivation of variable-length symbol-level BCJR algorithm. With this, the joint MAP VLC source decoder can be developed by exploiting the reliability information received on different channels and combining them with inter-description and intra-description correlation to provide a better estimation.

5.3 SISO Source Decoding of Variable-Length Codes

The transmission of continuous-valued, autocorrelated source samples is considered. Figure 5.1 shows our model of a transmission system, where a sequence of T source samples is given by $\mathbf{v} = [v_1, v_2, ..., v_T]$. The transmitter consists of a variable-length source encoder and a convolutional channel encoder separated by an interleaver. Suppose at time t, the input sample v_t is quantized by the symbol u_t with M bits. The quantizer's reproduction level corresponding to the symbol $u_t = \lambda$ is denoted by c_λ ,where $\lambda \in \mathcal{I}$ from the finite alphabet $\mathcal{I} = \{0, 1, ..., 2^M - 1\}$. We can generally assume that there is a certain amount of residual redundancy remaining in the symbol sequence due to delay and complexity constraints for the quantization stage. The scalar quantizer is followed by a VLC encoder, which maps a fixed-length symbol $u_t = \lambda$ to

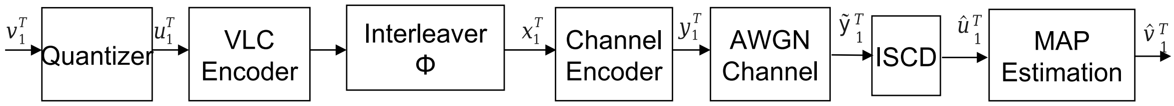

Figure 5.1: Model of the transmission system.

a variable-length bitvector $c(\lambda) = (u_t(1), u_t(2), ..., u_t(l(c(\lambda))))$ of length $l(c(\lambda))$ using the VLC code C. The output of the VLC encoder is given by the binary sequence $\mathbf{b} = [b(1), \ldots, b(n), \ldots, b(N)]$ with a total bit length N, where $b(n)$ represents a single bit at bit instant n. Following the work of [40], we considered the binary sequence $\mathbf{b}$ as one particular codeword of a code B whose codewords are all possible combinations of VLC codewords with total length N. A block of T symbols, written as $u_1^T = \{u_1, \ldots, u_t, \ldots, u_T\}$, are interleaved by a symbol interleaver Φ. The interleaved symbol sequences is denoted by $x_1^T = \{x_1, \ldots, x_t, \ldots, x_T\}$, where each symbol $x_t = \Phi(u_t)$ is associated with a bitvector $(x_t(1), x_t(2), ..., x_t(l(c(\lambda)))$. Afterwards the interleaved bitstream is then encoded by a rate-1/2 systematic convolutional channel encoder whose output is denoted by $y_1^T = \{y_1, \ldots, y_t, \ldots, y_T\} = \{(x_1, z_1), \ldots, (x_t, z_t), \ldots, (x_T, z_T)\}$, where $x_t(l)$ and $z_t(l)$ represent the systematic and parity symbol, respectively. For all simulations, binary phase shift keying (BPSK) is used as modulation scheme and an AWGN channel is assumed for transmission. The transmission of a symbol sequence y_1^T (to a bit sequences b_1^N) over an AWGN channel leads to a symbol sequence $\tilde{y}_1^T$ (to a soft-bit sequence $\tilde{b}_1^{2N}$) at the channel output.

1896

For the concatenation of VLC and channel coding, the turbo-like evaluation of residual source redundancy and of artificial channel-code redundancy makes step-wise quality gains possible by iterative decoding. The ISCD consists of two constituent decoders with soft-inputs and soft-outputs (SISO). Goal of the channel decoder is to process the received code sequence $\tilde{y}_1^T = \{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_T\}$ and combines them with the source *a priori* information to compute the extrinsic information. The source decoder computes an extrinsic value which, after interleaving, can be exploited as additional *a priori* knowledge by the channel decoder in the next iteration. Exchanging extrinsic information between two constituent decoders is iteratively repeated until the reliability gain becomes insignificant. After the last iteration, the *a posteriori* information resulting from the source decoder is used as *a priori* knowledge in a MAP VLC sequence estimation [46], followed by a dequantization leads to an estimate $\hat{v}_t$ of the transmitted source sequence v_t.

The determination rules of extrinsic information of SBSD has been derived in [20], but a slight modification is proposed which allows a delay of T samples in the decoding process. We have chosen the length T in compliance with the defined size of an interleaving block. If on the basis of a first-order Markov model time-correlation between consecutive symbols shall be utilized, then the entire history of received codewords $\tilde{u}_1^t$ and possibly additionally given future codewords $\tilde{u}_{t+1}^T$ have to be considered as well. To advance with this, we derive a forward-backward recursive algorithm that shows how the past and future received codewords can be transformed into extrinsic information utilizable in the iterative decoding process. The basic strategy is to jointly exploit the channel information, the source a priori information as well as the extrinsic information resulting from the SISO channel decoding. We consider a sequence of T source symbols, each of which is encoded by a VLC with alphabet size 2^M. All the possible bit sequences with symbol-length T and bit-length N can be represented in a VLC trellis digram. An example for VLC trellis representation is shown in Figure 5.2 for $T = 4$ and $N = 10$, where N_t represents the set of all possible positions g_t at time instant t. Note that, the state transition is from g_{t-1} to $g_t = l(c(\lambda)) + g_{t-1}$, given an input $u_t = \lambda$. Taking trellis states g_t and g_{t-1} into consideration, APP for each of possibly transmitted symbol $u_t = \lambda$ given the received sequence $\tilde{y}_1^T = \{\tilde{u}_1^T, \tilde{z}_1^T\}$, is given by

$$P_{SD}(u_t = \lambda|\tilde{y}_1^T) = \textstyle\sum_{g_t} \sum_{g_{t-1}} P_{SD}(g_{t-1}, u_t = \lambda, g_t|\tilde{y}_1^T) \tag{5.1}$$

where $P_{SD}(g_{t-1}, u_t = \lambda, g_t|\tilde{y}_1^T)$ can be further decomposed by using the Bayes theorem

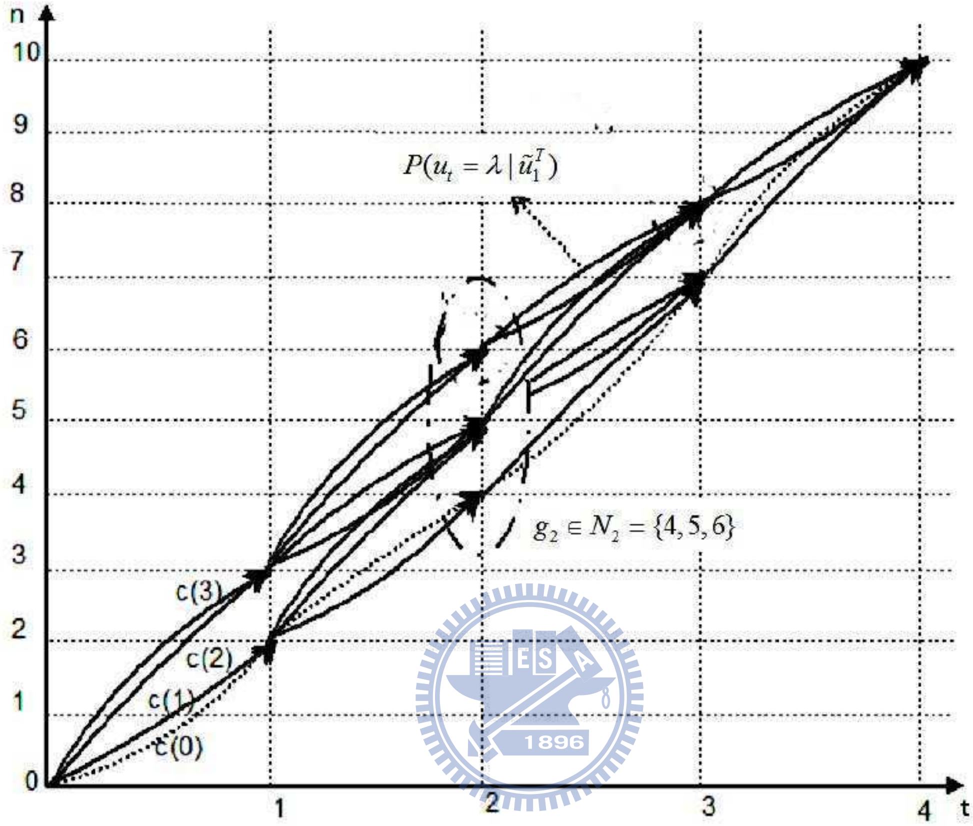
n
10
9
8
7
6
5
4
3
2
1
0
1
2
3
4
t
$P(u_t = \lambda \mid \tilde{u}_1^T)$
$g_2 \in N_2 = \{4,5,6\}$
c(3)
c(2)
c(1)
c(0)

Figure 5.2: VLC trellis representation for $T = 4$, $N = 10$ and $C = \{c(0) = 11, c(1) = 00, c(2) = 101, c(3) = 010\}$.

as

$$P_{SD}(g_{t-1}, u_t = \lambda, g_t|\tilde{y}_1^T) \; = \alpha_t^u(\lambda)\beta_t^u(\lambda) \cdot \frac{P(\tilde{z}_1^T|g_{t-1},u_t=\lambda,g_t,\tilde{u}_1^T)}{P(\tilde{y}_1^T)} \tag{5.2}$$

where $\alpha_t^u(\lambda) = P(g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^t)$ and $\beta_t^u(\lambda) = P(\tilde{u}_{t+1}^T|g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^t)$. Using the Markov property of the symbols and the memoryless assumption of the channel, the forward and backward recursions of the algorithm can be expressed as

$$
\begin{aligned}
\alpha_t^u(\lambda) &= \sum_{g_{t-2}} \sum_{q:l(q)=g_{t-1}-g_{t-2}} P(g_{t-2}, u_{t-1}=q, g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t-2}} \sum_{q:l(q)=g_{t-1}-g_{t-2}} P(u_t=\lambda, g_t, \tilde{u}_t | g_{t-2}, u_{t-1}=q, g_{t-1}, \tilde{u}_1^{t-1}) \\
&\cdot P(g_{t-2}, u_{t-1}=q, g_{t-1}, \tilde{u}_1^{t-1}) \\
&= \sum_{g_{t-2}} \sum_{q:l(q)=g_{t-1}-g_{t-2}} \gamma_{\lambda,q}^u(u_t, g_t, g_{t-2}) \alpha_{t-1}^u(q)
\end{aligned}
\tag{5.3}
$$

and

$$
\begin{aligned}
\beta_t^u(\lambda) &= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} P(\tilde{u}_{t+1}^T, g_{t+1}, u_{t+1}=q | g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} P(\tilde{u}_{t+2}^T, \tilde{u}_{t+1}, g_{t+1}, u_{t+1}=q | g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} P(\tilde{u}_{t+2}^T | \tilde{u}_{t+1}, g_{t+1}, u_{t+1}=q, g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&\cdot P(\tilde{u}_{t+1}, g_{t+1}, u_{t+1}=q | g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} P(\tilde{u}_{t+2}^T | \tilde{u}_1^{t+1}, g_{t+1}, u_{t+1}=q, g_{t-1}, u_t=\lambda, g_t) \\
&\cdot P(\tilde{u}_{t+1}, g_{t+1}, u_{t+1}=q | g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} P(\tilde{u}_{t+2}^T | \tilde{u}_1^{t+1}, g_{t+1}, u_{t+1}=q, g_{t-1}) \\
&\cdot P(\tilde{u}_{t+1}, g_{t+1}, u_{t+1}=q | g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^t) \\
&= \sum_{g_{t+1}} \sum_{q:l(q)=g_{t+1}-g_t} \gamma_{q,\lambda}^u(u_{t+1}, g_{t+1}, g_{t-1}) \beta_{t+1}^u(q)
\end{aligned}
\tag{5.4}
$$

and in (5.3)

$$
\begin{aligned}
\gamma_{\lambda,q}^u(u_t, g_t, g_{t-2}) &= P(u_t=\lambda, g_t, \tilde{u}_t | g_{t-2}, u_{t-1}=q, g_{t-1}, \tilde{u}_1^{t-1}) \\
&= P(\tilde{u}_t | u_t=\lambda, g_t, u_{t-1}=q, g_{t-1}, g_{t-2}, \tilde{u}_1^{t-1}) \\
&\cdot P(u_t=\lambda, g_t | u_{t-1}=q, g_{t-1}, g_{t-2}, \tilde{u}_1^{t-1}) \\
&= P(\tilde{u}_t | u_t=\lambda, g_t) \\
&\cdot P(u_t=\lambda, g_t | u_{t-1}=q, g_{t-1}, g_{t-2})
\end{aligned}
\tag{5.5}
$$

where

$$\begin{aligned} &P(u_t=\lambda, g_t|u_{t-1}=q, g_{t-1}, g_{t-2}) \\ &= \tfrac{1}{C(q,g_{t-1})}\{ \begin{array}{ll} P(u_t=\lambda|u_{t-1}=q), & \text{for } l(c(\lambda))=g_t-g_{t-1} \\ 0, & \text{otherwise} \end{array} \end{aligned} \tag{5.6}$$

with the normalization factor

$$C(q, g_{t-1}) = \sum_{g_t} \sum_{\lambda\in I: l(c(\lambda)=g_t-g_{t-1}} P(u_t=\lambda|u_{t-1}=q). \tag{5.7}$$

Assuming an AWGN channel with zero mean and variance $\sigma_n^2 = N_0/2E_s$, the conditional probability density function (pdf) of $\tilde{u}_t$ can be formulated as

$$\begin{aligned} P(\tilde{u}_t|u_t=\lambda, g_t) &= \textstyle\prod_{m=1}^{l(c(\lambda))} p(\tilde{b}(g_{t-1}+m)|u_t(m))) \\ &= (\tfrac{1}{\sqrt{2\pi}\sigma_n})^M \cdot e^{-\frac{E_s}{N_0}\sum_{m=1}^{l(c(\lambda))}(\tilde{b}(g_{t-1}+m)-u_t(m)))^2} \end{aligned} \tag{5.8}$$

An iterative process using the SISO source decoder as a constituent decoder is realizable, if the APPs $P_{SD}(g_{t-1}, u_t = \lambda, g_t|\tilde{y}_1^T)$ can be separated into four additive terms: the *a priori* probability $P(u_t = \lambda, g_t)$ in terms of the a priori probability $P(u_t = \lambda)$, the channel-related probability $P_c(u_t = \lambda, g_t) = P(\tilde{u}_t|u_t = \lambda, g_t)$, and two extrinsic terms resulting from source and channel decoding. In order to determine each of the four terms, we can substitute (5.5) into (5.3) and rewrite (5.2) as follows:

$$\begin{aligned} P_{SD}(g_{t-1}, u_t=\lambda, g_t|\tilde{y}_1^T) &= P(u_t=\lambda, g_t)\cdot P_c(u_t=\lambda, g_t) \\ &\cdot P_{SD}^{[ext]}(u_t=\lambda, g_t)\cdot P_{CD}^{[ext]}(u_t=\lambda, g_t) \end{aligned} \tag{5.9}$$

where

$$P_{CD}^{[ext]}(u_t=\lambda, g_t) = P(\tilde{z}_1^T|g_{t-1}, u_t=\lambda, g_t, \tilde{u}_1^T) \tag{5.10}$$

and

$$P_{SD}^{[ext]}(u_t=\lambda,g_t)=\beta_t^u(\lambda)\cdot\sum_{g_{t-2}}\sum_{q}P(u_t=\lambda,g_t|u_{t-1}=q,g_{t-1},g_{t-2})\alpha_{t-1}^u(q).\tag{5.11}$$

A detail derivation of (5.10) is presented in the next section. Within the iterations the precision of APP estimation can be enhanced by multiplying $P(u_t=\lambda,g_t|u_{t-1}=q,g_{t-1},g_{t-2})$ in (5.5) by $P_{CD}^{[ext]}(u_t=\lambda,g_t)$ from the channel decoder. The interleaved extrinsic probability can be computed according to

$$P_{SD}^{[ext]}(x_t=\lambda,g_t)=\frac{P_{SD}(g_{t-1},x_t=\lambda,g_t|\tilde{y}_1^T)}{P(x_t=\lambda,g_t)P_c(x_t=\lambda,g_t)P_{CD}^{[ext]}(x_t=\lambda,g_t)}\tag{5.12}$$

and used as new *a priori* information in the next channel decoding round. Notice that the term $P_{SD}(g_{t-1},x_t=\lambda,g_t|\tilde{u}_1^T)$ represents the APP for each interleaved symbol and can be computed by

$$P_{SD}(g_{t-1},x_t=\lambda,g_t|\tilde{y}_1^T)=C'\cdot P_{SD}(x_t=\lambda|\tilde{y}_1^T)\tag{5.13}$$

with the normalization factor

$$C'=\sum_{q}\sum_{\{g_t,g_{t-1}:g_t-g_{t-1}=l(q)\}}P_{SD}(x_t=q|\tilde{y}_1^T).$$

where $P_{SD}(x_t=q|\tilde{y}_1^T)=\Phi(P_{SD}(u_t=q|\tilde{y}_1^T))$.

5.4 SISO Channel Decoding of Convolutionally-Encoded VLC

For the transmission scheme with channel coding, a soft-output channel decoder can be used to provide both decoded bits and their reliability information for further processing to improve the system performance. The commonly used BCJR algorithm is a trellis-based MAP decoding algorithm for both linear block and convolutional codes.

The derivation presented in [23] led to a forward-backward recursive computation on the basis of a bit-level trellis diagram, which has two branches leaving each state and every branch represents a single symbol-bit. Proper sectionalization of a bit-level code trellis may result in useful trellis structural properties [31][32] and allows us to devise SISO channel decoding algorithms which incorporate parameter-oriented extrinsic information from the source decoder. To proceed with this, we propose a modified BCJR algorithm which parses the received code-bit sequence into blocks and computes the APP for each parameter symbol on a symbol-by-symbol basis. Unlike classical BCJR algorithm that decodes one bit at a time, our scheme proceeds with decoding the parameter symbols as nonbinary symbols that are matched to the number of bits in an symbol. By parsing the code-bit sequence into $l(c(x_t))$-bit symbols, we are in essence merging $l(c(x_t))$ stages of the original bit-level code trellis into one. In previous work[31][32], the sectionalized trellis originally proposed for fixed length codevector, is not suit for symbol-by-symbol channel decoding of convolutionally-encoded VLC. In the case of variable-length branch, different paths entering a state have used up a different number of bits from the received sequence and can therefore be extended differently [48]. For this reason, we extend 1-dimensional state s_t to 2-dimensional state As an example, we illustrate in Figure 5.3 three stages of the bit-level trellis diagram of a rate 1/2 convolutional encoder with generator polynomial $(7, 5)_8$. The solid lines and dashed lines correspond to the input bits of 0 and 1, respectively. $\sigma_t = (s_t, g_t)$. Figure 5.3 also shows the sectionalized trellis diagram when two stages of the original bit-level trellis are merged together. In general, there are 2^M branches leaving and entering each state in a $l(c(\lambda))$-stage merged trellis diagram. Having defined the trellis structure as such, there will be one symbol APP corresponding to each branch which represents a particular parameter symbol $x_t = \lambda$. For convenience, we say that the sectionalized trellis diagram forms a finite-state machine defined by its state transition function $F_\sigma(x_t, \sigma_{t-1})$ and output function $F_p(x_t, \sigma_{t-1})$. Viewing from this perspective, the code-bit combination associated with the branch from state σ_{t-1} to state $\sigma_t = F_\sigma(x_t, \sigma_{t-1})$ can be written as $y_t = (x_t, z_t)$, where $z_t = F_p(x_t, \sigma_t)$ is the variable-length codevector

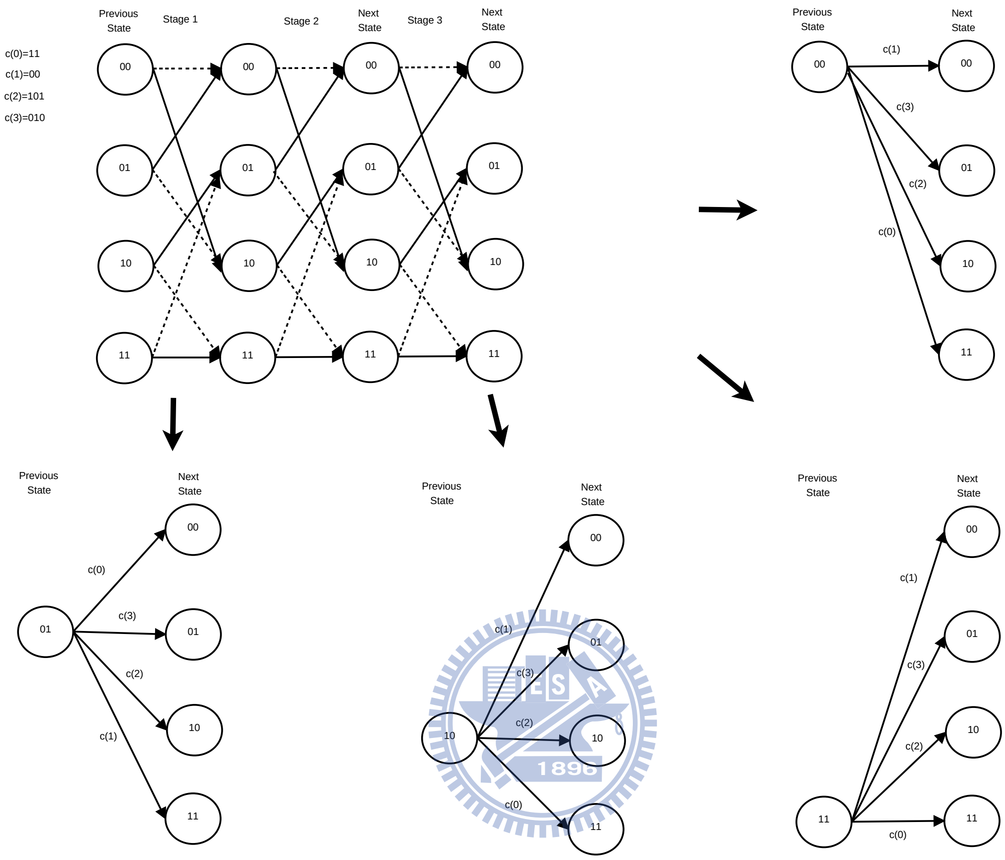

Figure 5.3: Bit-level and merged trellis for $C = \{c(0) = 11, c(1) = 00, c(2) = 101, c(3) = 010\}$.

output given state $\sigma_t = (s_t, g_t) \in \mathcal{S} \times N_t$. $\mathcal{S}$ represents the set of 2^ν possible states for the convolutional encoder with memory order ν. An example for 3-dimension trellis digram is shown in Figure 5.4 for $N = 5$, $T = 2$ and $\nu = 2$.

We next applied this new sectionalized code trellis to compute the APP of a systematic symbol $x_t = \lambda$ given the received code sequence $\tilde{Y}_1^T = \{\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_T\}$ in which $\tilde{y}_t = (\tilde{x}_t, \tilde{z}_t)$. Taking the trellis state σ_t into consideration, we rewrite the APP as follows:

$$P_{CD}(x_t = \lambda | \tilde{y}_1^T) = C \sum_{\sigma_t} P(x_t = \lambda, \sigma_t, \tilde{y}_1^T) = C \sum_{\sigma_t} \alpha_t^x(\lambda, \sigma_t) \beta_t^x(\lambda, \sigma_t), \tag{5.14}$$

where $\alpha_t^x(\lambda, \sigma_t) = P(x_t = \lambda, \sigma_t, \tilde{y}_1^t)$, $\beta_t^x(\lambda, \sigma_t) = P(\tilde{y}_{t+1}^T | x_t = \lambda, \sigma_t, \tilde{y}_1^t)$, and $C =$

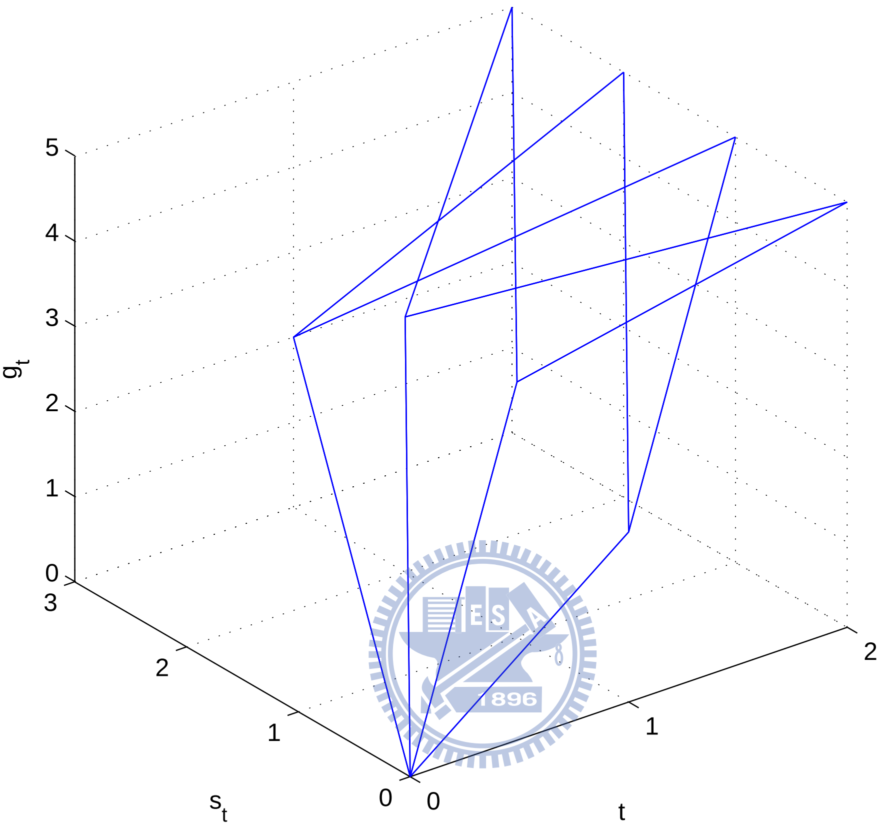

Figure 5.4: Three-dimension Trellis digram for $\nu = 2$, $N = 5$ and $T = 2$.

$1/P(\tilde{y}_1^T)$ is a normalizing factor. For the recursive implementation, the forward and backward recursions are to compute the following metrics:

$$\begin{aligned}
\alpha_t^x(\lambda, \sigma_t) &= \sum_{\sigma_{t-1}} \sum_{q} P(x_t = \lambda, \sigma_t, x_{t-1} = q, \sigma_{t-1}, \tilde{y}_t, \tilde{y}_1^{t-1}) \\
&= \sum_{\sigma_{t-1}} \sum_{q} \alpha_{t-1}^x(q, \sigma_{t-1}) \gamma_{\lambda,q}^x(\tilde{y}_t, \sigma_t, \sigma_{t-1}) \qquad (5.15)
\end{aligned}$$

$$\begin{aligned}
\beta_t^x(\lambda, \sigma_t) &= \sum_{\sigma_{t+1}} \sum_{q} P(x_{t+1} = q, \sigma_{t+1}, \tilde{y}_{t+1}, \tilde{y}_{t+2}^T | x_t = \lambda, \sigma_t, \tilde{y}_1^t) \\
&= \sum_{\sigma_{t+1}} \sum_{q} \beta_{t+1}^x(q, \sigma_{t+1}) \gamma_{q,\lambda}^x(\tilde{y}_{t+1}, \sigma_{t+1}, \sigma_t) \qquad (5.16)
\end{aligned}$$

and in (5.15)

$$\begin{aligned}\gamma^{x}_{\lambda,q}(\tilde{y}_t,\sigma_t,\sigma_{t-1}) &= P(x_t=\lambda,\sigma_t,\tilde{y}_t|x_{t-1}=q,\sigma_{t-1}) \\ &= P(s_t|x_t=\lambda,g_t,\tilde{y}_t,x_{t-1}=q,\sigma_{t-1}) \\ &\cdot P(\tilde{y}_t|x_t=\lambda,g_t,x_{t-1}=q,\sigma_{t-1})P(x_t=\lambda,g_t|x_{t-1}=q,\sigma_{t-1}) \\ &= P(s_t|x_t=\lambda,s_{t-1})P(\tilde{y}_t|x_t=\lambda,g_t,\sigma_{t-1}) \\ &\cdot P(x_t=\lambda,g_t|x_{t-1}=q,\sigma_{t-1}).\end{aligned} \tag{5.17}$$

Having a proper representation of the branch metric $\gamma_{\lambda,q}(\tilde{y}_t,\sigma_t,\sigma_{t-1})$ is the critical step in applying symbol decoding to error mitigation and one that conditions all subsequent steps of the implementation. As a practical manner, several additional factors must be considered to take advantage of the symbol-level trellis structure and AWGN channel assumption. First, making use of the merged variable length code trellis, the value of $P(s_t|x_{t-1}=q,s_{t-1})$ is either one or zero depending on whether symbol q is associated with transition from state σ_{t-1} to state $\sigma_t=F_\sigma(x_t=\lambda,\sigma_{t-1})$. For AWGN channels, the second term in (5.17) is reduced to

$$\begin{aligned}P(\tilde{y}_t|x_t=\lambda,g_t,\sigma_{t-1}) &= P(\tilde{x}_t|x_t=\lambda,\tilde{z}_t,g_t,\sigma_{t-1})P(\tilde{z}_t|x_t=\lambda,\sigma_{t-1},g_t) \\ &= P(\tilde{x}_t|x_t=\lambda,g_t)P(\tilde{z}_t|z_t=F_p(x_t=\lambda,\sigma_{t-1}),g_t)\end{aligned} \tag{5.18}$$

where the conditional pdfs for the received systematic and parity symbols can be computed analogous to (5.6). The third term in (5.17) is reduced to $P(x_t=\lambda,g_t)$ under the assumption that x_t is uncorrelated with x_{t-1}, which is indeed the case as x_t is the interleaved version of parameter symbols. Within the iterations the a priori information $P(x_t=\lambda,g_t)$ can be improved by additional a priori information which is provided by the SISO source decoder in terms of its extrinsic probability $P^{[ext]}_{SD}(x_t=\lambda,g_t)$.

The decoder's next step is to compute the deinterleaved extrinsic information for each symbol. We first compute APPs for deinterleaved symbol $u_t=\lambda$ which takes VLC trellis states into consideration as follows:

$$P_{CD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) \; = C' \cdot P_{CD}(u_t = \lambda | \tilde{y}_1^T) \tag{5.19}$$

where the normalization factor

$$C' = \sum_q \sum_{(g_t, g_{t-1}) \in \{(g_t, g_{t-1}) : g_t - g_{t-1} = l(q)\}} P_{CD}(u_t = q | \tilde{y}_1^T)$$

and $P_{CD}(u_t = q|\tilde{y}_1^T) = \Phi^{-1}(P_{CD}(x_t = q|\tilde{y}_1^T))$.

An iterative process using the SISO channel decoder as a constituent decoder is realizable, if the reliability of the APPs $P_{CD}(g_{t-1}, u_t = \lambda, g_t|\tilde{y}_1^T)$ can be separated into three terms according to Bayes theorem: the *a priori* probability $P_a(u_t = \lambda, g_t)$, the channel-related probability $P_c(u_t = \lambda, g_t) = P(\tilde{u}_t|u_t = \lambda, g_t)$, and an extrinsic term resulting from $P_{CD}^{[ext]}(u_t = \lambda, g_t)$. The equation (5.19) leads to

$$\begin{aligned}
P_{CD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) \; &= P(u_t = \lambda, g_{t-1}, g_t, \tilde{u}_1^T, \tilde{z}_1^T)/P(\tilde{y}_1^T) \\
&= P(u_t = \lambda, g_{t-1}, g_t, \tilde{u}_1^T) \cdot \tfrac{P(\tilde{z}_1^T | u_t = \lambda, g_{t-1}, g_t, \tilde{u}_1^T)}{P(\tilde{y}_1^T)} \\
&= P(\tilde{u}_t | u_t = \lambda, g_{t-1}, g_t, \tilde{u}_1^{t-1}, \tilde{u}_{t+1}^T) \cdot P(x_t = \lambda, g_t, g_{t-1}) \\
&\cdot P(\tilde{u}_1^{t-1}, \tilde{u}_{t+1}^T) \cdot P(\tilde{z}_1^T | u_t = \lambda, g_{t-1}, g_t, \tilde{U}_1^T)/P(\tilde{y}_1^T) \\
&= C \cdot P_c(\tilde{u}_t | u_t = \lambda, g_t) \cdot P(u_t = \lambda, g_t) \\
&\cdot P_{CD}^{[ext]}(u_t = \lambda, g_t)
\end{aligned} \tag{5.20}$$

where C is a normalization factor and

$$P_{CD}^{[ext]}(u_t = \lambda, g_t) \; = P(\tilde{z}_1^T | u_t = \lambda, g_{t-1}, g_t, \tilde{u}_1^T). \tag{5.21}$$

With this, the deinterleaved extrinsic probability $P_{CD}^{[ext]}(u_t = \lambda, g_t)$ can be calulated by

$$P_{CD}^{[ext]}(u_t = \lambda, g_t) \; = \frac{P_{CD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)}{P_c(u_t = \lambda, g_t) P_{SD}^{[ext]}(u_t = \lambda, g_t) P(u_t = \lambda, g_t)} \tag{5.22}$$

and used as new *a priori* information for the source decoder. Similarly, the interleaved extrinsic probability $P_{CD}^{[ext]}(x_t = \lambda, g_t)$ can be calulated by

$$P_{CD}^{[ext]}(x_t = \lambda, g_t) \; = \frac{P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)}{P_c(x_t = \lambda, g_t) P_{SD}^{[ext]}(x_t = \lambda, g_t) P(x_t = \lambda, g_t)}. \tag{5.23}$$

Finally, we summarize the proposed symbol-based ISCD of convolutionally-encoded VLC as follows:

1. Initialization:
   Set the extrinsic information of source decoding to $P_{SD}^{[ext]}(x_t, g_t) = 1$. Set the iteration counter to $n = 0$ and define an exit condition n_{max}.

2. Read series of received sequences $\tilde{y}_1^T$ and map all received systematic symbols $\tilde{x}_t$ to channel-related probabilities $P_c(x_t, g_t)$.

3. Perform MAP channel decoding on each symbol to compute the APP $P_{CD}(x_t = \lambda|\tilde{y}_1^T)$ leaded by substituting (5.15) and (5.16) into (5.14). Then the symbol APP for each interleaved symbol which takes VLC trellis states, $P(x_t = \lambda, g_t, g_{t-1}|\tilde{y}_1^T)$, is computed by (5.19). The de-interleaved extrinsic probability and interleaved extrinsic probability $P_{CD}^{[ext]}(u_t, g_t)$, $P_{CD}^{[ext]}(u_t, g_t)$ can be calculated by (5.22) and (5.23), respectively. These extrinsic probabilities can be used as *a priori* information for the source decoder.

4. Perform SISO source decoding of VLC by inserting the de-interleaved extrinsic probability $P_{CD}^{[ext]}(u_t, g_t)$ into (5.5) and then de-interleaved extrinsic probability for each symbol is calculated by (5.11). Compute the symbol *a posteriori* probability $P_{SD}(u_t, g_t, g_{t-1}|\tilde{y}_1^T)$ by (5.9) and then substitute these probabilities into (5.1) to lead APPs $P_{SD}(u_t|\tilde{y}_1^T)$. Interleave these APPs to compute the interleaved symbol APP for each symbol which takes trellis states, $P(x_t = \lambda, g_t, g_{t-1}|\tilde{y}_1^T)$ by (5.13). Then, the interleaved extrinsic probability $P_{SD}^{[ext]}(x_t, g_t)$ is computed by (5.12) and is forwarded to the channel decoder as *a priori* information.

5. Increase the iteration counter $n \leftarrow n + 1$. If the exit condition $n = n_{max}$ is fulfilled, then continue with step 6, otherwise proceed with step 3.

6. Using the symbol APPs obtained from step 4 to calculate the decoder output

signals $\hat{v}_t$ by MAP estimation, where the estimated value are given as

$$\hat{v}_t = c_{\lambda^*}, \lambda^* = \arg\max_{\lambda} P_{SD}(u_t = \lambda | \tilde{y}_1^T) \tag{5.24}$$

Bibliography

[1] J. Rosenberg, L. Qiu, and H. Schulzrinne, "Integrating pakcet FEC into adaptive voice playout buffer algorithms on the internet," in *Proc. IEEE INFOCOM 2000*, vol. 3, Tel Aviv, Israel, Mar. 2000, pp. 1705-1714

[2] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *International Conference on Multimedia and Expo*, New York, USA, August . 2000, vol. 1,pp. 444–447.

[3] Y.J. Liang, E.G. Steinbach, and B. Girod, "Multi-stream voice over IP using packet path diversity," in *Multimedia Signal Processing IEEE Fourth Workshop*, 2001, pp. 555–560.

[4] J. Balam and J. D. Gibson "Multiple descriptions and Path Diversity for Voice Communications Over Wireless Mesh Networks, " *IEEE Transactions on Multimedia*, August 2007.

[5] V. K. Goyal, "Multiple description coding: 'compression meets the network," *IEEE Signal Processing Magazine*, Sep. 2001.

[6] A. Ingle and V. A. Vaishampayan, "DPCM system design for diversity systems with applications to packetized system," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821-834, May 1993.

[7] W. Jiang and A. Ortega,“Multiple description speech coding for robust commonucation over lossy packet networks,” in *Proc. IEEE Int. Conf. Multimedia and Expo.*, 2000, vol. 1, pp.444-447.

[8] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvine, ,“The adaptive multirate wideband speech codec (AMR-WB),” *IEEE Trans. Speech Audio Proces.*, Nov. 2002.

[9] International Telecommunication Union,“Coding of Speech at 8 kbit/s Using Conjugate- Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP),” *ITU-T G.729 Recommendation.*, Nov. 2000.

[10] V. A. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Trans. Speech Audio Process*, vol. 3, no. 1, pp. 48-58, Jan. 1995.

[11] S.B. Moon, J. Kurose, and D. Towsley, “Packet audio playout delay adjustment: Performance bounds and algorithms,” *Multimedia Systems* , vol. 6, no. 1, pp. 17–28, Jan. 1998.

[12] L. Sun and E. Ifeachor, “Voice quality prediction models and their application in VoIP networks,” *IEEE Transactions on Multimedia*, August 2006.

[13] K. Fujimoto, S. Ata, and M. Murata “Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications ,” in *Processings of IEEE Globecom*, Nov 2002.

[14] S. Lin and D.J. Costello, Error Control Coding, Pearson Prentice Hall, New Jersey, 2004.

[15] J. Rosenberg, L. Qiu, and H. Schulzrinne, “Integrating pakcet FEC into adaptive voice playout buffer algorithms on the internet,” in *Proc. IEEE INFOCOM 2000*, vol. 3, Tel Aviv, Israel, Mar. 2000, pp. 1705-1714

[16] C. Boutremans and J. Boudec, “Adaptive Joint Playout Buffer and FEC Adjustemnt for Internet Telephony,” in *Processings of IEEE INFOCOM*, 2003.

[17] Chia-Chen Kuo, Ming-Syan Chen, and Jeng-Chun Chen, “An Adaptive Transmission Scheme for Audio and Video Synchronization based on Real-time Transport Protocol, ” in *IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, August 2001.

[18] International Telecommunication Union, “The E-model, a computational model for use in transmission planning, ” *ITU-T Recommendation G.107*, July 2000.

[19] N. Gortz, “A generalized framework for iterative source-channel decoding,” *Annals of Telecommunications, Special Issue on Turbo Codes,* pp. 435-446, July/August 2001.

[20] M. Adrat, P. Vary, and J. Spittka, “Iterative source-channel decoder using extrinsic information from softbit source decoding,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* vol. 4, pp. 2653-2656, Salt Lake City, Utah, USA, May 2001.

[21] M. Srinivasan, “Iterative decoding of multiple descriptions,” *in Proc. IEEE ICC*, MArch 1999, pp. 3-12.

[22] J. Barros, J. Hagenauer, and N. Gortz, “Turbo cross decoding of multiple descriptions,” *in Proc. IEEE ICC*, vol. 3, 2002, pp. 1398-1402.

[23] L. R. Bahl, Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, Mar. 1974.

[24] R. Cole and J. Rosenbluth, “Voice over IP performance monitoring,” in *Journal on Computer Communication Review*, vol. 31, no. 2, Apr. 2001.

[25] International Telecommunication Union "Perceptual Evaluation of Speech Quality (PESQ), An Objective Method for End-to-end Speech Quality Assessment of Narrow- band Telephone Networks and Speech Codecs" *ITU-T Recommendation P.862*, Feb. 2001.

[26] P.A. Barrent, R.M. Voelcker, and A.V. Lewis, "Speech transmission over digital mobile radio channels", *BT Technology Journal*, vol. 14, no. 1, pp. 45-56, Jan. 1996.

[27] L. Ding and R.A. Goubran, "Assessment of effects of packet loss on speech quality in VoIP", in *Proceeding of IEEE International workshop on haptic, audio and visual environments and their applications*, pp. 49-54, Sep. 2003.

[28] International Telecommunication Union, "Objective measuring apparatus, Appendix 1: Test signals," *ITU-T Recommendation P.50*, Feb. 1998.

[29] E.K.P. Chong and S.H. Zak, *An Introduction to Optimization*, John Wiley & Sons, Inc., 2001.

[30] Chun-Feng Wu, and Wen-Whei Chang "Perceptual Optimization of Playout Buffer in VoIP Appications, " in *Proceedings of Chinacom*, Oct. 2006.

[31] Y. Liu, S. Lin, and M.P.C. Fossorier, "MAP algorithms for decdoing linear block codes based on sectionalied trellis diagrams," *IEEE Trans. Commun.*, vol. 48, pp. 577-587, April 2000.

[32] M. Bingeman and A. K. Khandani, "Symbol-based turbo codes," *IEEE Communications Letters,* vol. 3, pp. 285-287, Oct. 1999.

[33] J. A. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. Commun.*, vol. 42, pp. 1661-1671, 1994.

1896

[34] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” in *Proc. IEEE International Conference on Communication,* vol. 2, pp. 1009-1013, Jun 1995.

[35] M. Srinivasan, “Iterative decoding of multiple descriptions,” *in Proc. IEEE ICC*, March 1999, pp. 3-12.

[36] T. Fingscheidt and P. Vary, “Softbit speech decoding: a new approach to error concealment,” *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, pp. 240-251, March 2001.

[37] N. Gortz, “On the iterative approximation of optimal joint source-channel decoding,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1662-1670, 2001.

[38] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, N.J., 1984

[39] R. Bauer and J. Hagenauer, “On variable length codes for iterative source/channel decoding,” in *Proc. IEEE Data Compression Conference* , March 2001, pp.273-282.

[40] R. Thobaben and J. Kliewer, “Low-complexity iterative joint source-channel decoding for variable-length encoded Markov sources,” *IEEE Transactions on Communications*, vol.53, no.12, pp. 2054- 2064, Dec. 2005

[41] M. A. Bemard and B. D. Sharma, “Some combinatorial results on variable length error correcting codes”, in *Ars Combinatoria*, Vol. 25B, 1988, pp. 181-194

[42] R. Bauer, J. Hagenauer, “Iterative source channel-decoding using Reversible Variable Length Codes,” in *Proc. IEEE Data Compression Conference*, Snowbird, USA, March 2000, pp. 93-102

[43] Y. Takishima, M. Wada and H. Murakami, “Reversible variable length codes,” *IEEE Trans. on Comm.*, vol. COM-43, No. 2/3/4, 1995, pp. 158-162

[44] X. Wang and X. Wu, “Joint Source-Channel Decoding of Multiple Description Quantized and Variable Length Coded Markov Sequences,” in *Proc. IEEE International Conference on Multimedia and Expo,* , July 2006, pp.1429-1432.

[45] T. Guionnet, C. Guillemot, and E. Fabre, “Soft decoding of multiple descriptions,” in *IEEE International Conference on Multimedia, ICME*, vol. 2, Lausanne, Switzerland, August 26-29 2002, pp. 601604.

[46] J. Kliewer and R. Thobaben, “Iterative joint source-channel decoding of variable-length codes using residual source redundancy,” *IEEE Transactions on Wireless Communications*, vol.4, no.3, pp. 919- 929, May 2005

[47] R. Bauer and J. Hagenauer, “Symbol-by-symbol MAP decoding of variable length codes,” in*Proc. 3rd ITG Conf. Source Channel Coding*, Munich, Germany, Jan. 2000, pp. 111116.

[48] K. Sayood, H.H Otu and N. Demir, “Joint source/channel coding for variable length codes,” *IEEE Transactions on Communications*, vol.48, no.5, pp.787-794, May 2000

Appendix A

This section gives the detailed computation of $R^{(l)}(m, n, D_{F,i})$ and $S^{(l)}(m, n, D_{F,i})$ when (1) a Reed-Solomon code (N, K) is used, (2) packets are sent over a Gilbert channel and (3) the FEC delay of packet i is $D_{F,i}$. For $m = 1$, $n \geq 1$, $R^{(l)}(1, n, D_{F,i})$ is the probability that none of the packets are missing in the next $n-1$ packets following the network loss of packet i, and is given by

$$\begin{aligned} R^{(l)}(1, n, D_{F,i}) &= \Pr(W_{i+1}^{i+n-1} = 0^{n-1} | W_i = 1) \\ &= q^{(l)}(1 - p^{(l)})^{n-2} \cdot \textstyle\prod_{h=1}^{n-1}(1 - e_{b,i+h}^{(l)}). \end{aligned} \tag{A.1}$$

For $2 \leq m \leq n$, we compute $R^{(l)}(m, n, D_{F,i})$ conditionally to the event $\{A_j, B_j, C_j, j = 0, 1, \ldots, n-m\}$ on the arriving states of packets:

$$\begin{aligned} A_j &= \{W_i^{i+j+1} = 10^j1\} \\ B_j &= \{W_i^{i+j+1} = 10^j2\} \\ C_j &= \{m-2 \text{ missing packets in } W_{i+j+2}^{i+n-1}\} \end{aligned} \tag{A.2}$$

where 0^j is a shorthand for j successive 0's. For a Gilbert loss model with parameters $p^{(l)}$ and $q^{(l)}$, we have

$$\Pr(A_j) = \begin{cases} (1 - q^{(l)}), j = 0 \\ q^{(l)}(1 - p^{(l)})^{j-1}p^{(l)} \prod_{h=1}^{j}(1 - e_{b,i+h}^{(l)}), j \geq 1 \end{cases} \tag{A.3}$$

$$\Pr(B_j) = q^{(l)}(1 - p^{(l)})^j \textstyle\prod_{h=1}^{j}(1 - e_{b,i+h}^{(l)})e_{b,i+j+1}^{(l)}, j \geq 1 \tag{A.4}$$

$$\begin{aligned}\Pr(C_j|A_j) &= \Pr(m-2 \text{ missing packets in } W_{i+j+2}^{i+n-1}|W_i^{i+j+1} = 10^j1) \\ &= R^{(l)}(m-1, n-j-1, D_{F,i+j+1})\end{aligned} \tag{A.5}$$

$$\begin{aligned}\Pr(C_j|B_j) &= \Pr(m-2 \text{ missing packets in } W_{i+j+2}^{i+n-1}|W_i^{i+j+1} = 10^j2) \\ &= S^{(l)}(m-1, n-j-1, D_{F,i+j+1})\end{aligned} \tag{A.6}$$

From the total probability theorem, $R^{(l)}(m, n, D_{F,i})$ can be computed as follows:

$$\begin{aligned}R^{(l)}(m, n, D_{F,i}) &= \sum_{j=0}^{n-m} \Pr(C_j|A_j)\Pr(A_j) + \Pr(C_j|B_j)\Pr(B_j) \\ &= (1-q^{(l)})R^{(l)}(m-1, n-1, D_{F,i+1}) \\ &+ \sum_{j=1}^{n-m}\{q^{(l)}(1-p^{(l)})^{j-1}\prod_{h=1}^{j}(1-e_{b,i+h}^{(l)}) \cdot \{p^{(l)}R^{(l)}(m-1, n-j-1, D_{F,i+j+1}) \\ &+(1-p^{(l)})e_{b,i+j+1}^{(l)}S^{(l)}(m-1, n-j-1, D_{F,i+j+1})\}\}.\end{aligned} \tag{A.7}$$

Similarly, the probability $\tilde{R}^{(l)}(m, n, D_{F,i})$ can also be computed by recurrence as

$$\tilde{R}^{(l)}(m, n, D_{F,i}) = \tag{A.8}$$
$$\begin{cases} q^{(l)}(1-p^{(l)})^{n-2} \cdot \prod_{h=1}^{n-1}(1-e_{b,i-h}^{(l)}), m = 1, n \geq 1 \\ \\ (1-q^{(l)})R^{(l)}(m-1, n-1, D_{F,i+1}) \\ + \sum_{j=1}^{n-m}\{q^{(l)}(1-p^{(l)})^{j-1}\prod_{h=1}^{j}(1-e_{b,i-h}^{(l)}) \cdot \{p^{(l)}\tilde{R}^{(l)}(m-1, n-j-1, D_{F,i}) \\ +(1-p^{(l)})e_{b,i-j-1}^{(l)}\tilde{S}^{(l)}(m-1, n-j-1, D_{F,i})\}\}, 2 \leq m \leq n \end{cases}$$

Next, we give the detailed computation of $S^{(l)}(m, n, D_{F,i})$. For $m = 1$, $n \geq 1$, $S^{(l)}(1, n, D_{F,i})$ is the probability that none of the packet are missing in the next $n-1$ packets following the late loss of packet i, and is given by

$$\begin{aligned}S^{(l)}(1, n, D_{FEC,i}) &= \Pr(W_{i+1}^{i+n-1} = 0^{n-1}|W_i = 2) \\ &= e_{b,i}^{(l)}(1-p^{(l)})^{n-1} \cdot \prod_{h=1}^{n-1}(1-e_{b,i+h}^{(l)}).\end{aligned} \tag{A.9}$$

For $2 \leq m \leq n$, we compute $S^{(l)}(m, n, D_{F,i})$ conditionally to the event $\{C_j, D_j, E_j, j = 0, 1, \dots, n-m\}$ on the arriving states of packets:

$$\begin{aligned} C_j &= \{m-2 \text{ missing packets in } W_{i+j+2}^{i+n-1}\} \\ D_j &= \{W_i^{i+j+1} = 20^j 1\} \\ E_j &= \{W_i^{i+j+1} = 20^j 2\} \end{aligned} \tag{A.10}$$

For a Gilbert loss model with parameters $p^{(l)}$ and $q^{(l)}$, we have

$$\Pr(D_j) = e_{b,i}^{(l)}(1-p^{(l)})^j p^{(l)} \textstyle\prod_{h=1}^{j}(1-e_{b,i+h}^{(l)}) \tag{A.11}$$

$$\Pr(E_j) = e_{b,i}^{(l)}(1-p^{(l)})^{j+1} \textstyle\prod_{h=1}^{j}(1-e_{b,i+h}^{(l)})e_{b,i+j+1}^{(l)} \tag{A.12}$$

$$\begin{aligned} \Pr(C_j|D_j) &= P(m-2 \text{ missing in } W_{i+j+2}^{i+n-1}|W_i^{i+j+1} = 20^j 1) \\ &= R^{(l)}(m-1, n-j-1, D_{F,i+j+1}) \end{aligned} \tag{A.13}$$

$$\begin{aligned} \Pr(C_j|E_j) &= P(m-2 \text{ missing packets in } W_{i+j+2}^{i+n-1}|W_i^{i+j+1} = 20^j 2) \\ &= S^{(l)}(m-1, n-j-1, D_{F,i+j+1}) \end{aligned} \tag{A.14}$$

From the total probability theorem, $S^{(l)}(m, n, D_{F,i})$ can be computed as follows:

$$\begin{aligned} S^{(l)}(m, n, D_{F,i}) &= \sum_{j=0}^{n-m} \Pr(C_j|D_j)\Pr(D_j) + \Pr(C_j|E_j)\Pr(E_j) \\ &= \sum_{j=0}^{n-m} \{e_{b,i}^{(l)}(1-p^{(l)})^j \prod_{h=1}^{j}(1-e_{b,i+h}^{(l)}) \cdot \{p^{(l)}R^{(l)}(m-1, n-j-1, D_{F,i+j+1}) \\ &+(1-p^{(l)})e_{b,i+j+1}^{(l)}S^{(l)}(m-1, n-j-1, D_{F,i+j+1})\}\}. \end{aligned} \tag{A.15}$$

Similarly, $\tilde{S}^{(l)}(m,n,D_{F,i})$ can be computed by recurrence as

$$\tilde{S}^{(l)}(m,n,D_{F,i}) = \tag{A.16}$$
$$\begin{cases} e_{b,i}^{(l)}(1-p^{(l)})^{n-1}\cdot\prod_{h=1}^{n-1}(1-e_{b,i-h}^{(l)}), m=1, n\geq 1 \\ \\ \displaystyle\sum_{j=0}^{n-m}\{e_{b,i}^{(l)}(1-p^{(l)})^{j}\prod_{h=1}^{j}(1-e_{b,i-h}^{(l)})\cdot\{p^{(l)}\tilde{R}^{(l)}(m-1,n-j-1,D_{F,i-j-1}) \\ +(1-p^{(l)})e_{b,i-j-1}^{(l)}\tilde{S}^{(l)}(m-1,n-j-1,D_{F,i-j-1})\}\}, 2\leq m\leq n \end{cases}$$

Appendix B

In this Appendix B we shall show that the *a priori* LLR in (4.24) is equal to the deinterleaved sequence of extrinsic information provided by the SISO channel decoder, i.e, $L_a(u_{I,t} = l_I) = L_{CD}^{[ext]}(u_{I,t} = l_I)$. The APP of a systematic symbol $x_{D,t} = l_D$, given the received code sequences $\tilde{Y}_{D,1}^{T} = (\tilde{X}_{D,1}^{T}, \tilde{Z}_{D,1}^{T})$, can be decomposed by using the Bayes theorem as

$$
\begin{aligned}
&P(x_{D,t} = l_D | \tilde{Y}_{D,1}^{T}) && \text{(B.1)}\\
&= P(x_{D,t} = l_D, \tilde{X}_{D,1}^{T}) \cdot P(\tilde{Z}_{D,1}^{T} | x_{D,t} = l_D, \tilde{X}_{D,1}^{T}) / P(\tilde{Y}_{D,1}^{T})\\
&= P(\tilde{x}_{D,t} | x_{D,t} = l_D, \tilde{X}_{D,1}^{t-1}, \tilde{X}_{D,t+1}^{T}) \cdot P(x_{D,t} = l_D, \tilde{X}_{D,1}^{t-1}, \tilde{X}_{D,t+1}^{T})\\
&\cdot P(\tilde{Z}_{D,1}^{T} | x_{D,t} = l_D, \tilde{X}_{D,1}^{T}) / P(\tilde{Y}_{D,1}^{T})\\
&= C \cdot P(\tilde{x}_{D,t} | x_{D,t} = l_D) \cdot P(x_{D,t} = l_D) \cdot P(\tilde{Z}_{D,1}^{T} | x_{D,t} = l_D, \tilde{X}_{D,1}^{T})
\end{aligned}
$$

where $C = P(\tilde{X}_{D,1}^{t-1}, \tilde{X}_{D,t+1}^{T}) / P(\tilde{Y}_{D,1}^{T})$. We rewrite (B.1) in log-likelihood algebra as

$$
\begin{aligned}
&L(x_{D,t} = l_D | \tilde{Y}_{D,1}^{T}) && \text{(B.2)}\\
&= L_a(x_{D,t} = l_D) + L_c(x_{D,t} = l_D) + L_{CD}^{[ext]}(x_{D,t} = l_D)
\end{aligned}
$$

with

$$
L_{CD}^{[ext]}(x_{D,t} = l_D) = \log \frac{P(\tilde{Z}_{D,1}^{T} | x_{D,t} = l_D, \tilde{X}_{D,1}^{T})}{P(\tilde{Z}_{D,1}^{T} | x_{D,t} = 0, \tilde{X}_{D,1}^{T})}. \qquad \text{(B.3)}
$$

Since the de-interleaved sequence of $L_{CD}^{[ext]}(x_{D,t}=l_D)$ is used by the source decoder, we have

$$L_{CD}^{[ext]}(u_{D,t}=l_D)=\log\frac{P(\tilde{Z}_{D,1}^{T}|u_{D,t}=l_D,\tilde{U}_{D,1}^{T})}{P(\tilde{Z}_{D,1}^{T}|u_{D,t}=0,\tilde{U}_{D,1}^{T})} \tag{B.4}$$

where $\tilde{U}_{D,1}^{T}=\Phi^{-1}(\tilde{X}_{D,1}^{T})$. Once the LLR $L_{CD}^{[ext]}(u_{D,t}=l_D)$ has been determined, we can compute the probability as follows

$$P(\tilde{Z}_{D,1}^{T}|u_{D,t}=l_D,\tilde{U}_{D,1}^{T})=e^{L_{CD}^{[ext]}(u_{D,t}=l_D)}/\sum_{j=0}^{2^R-1}e^{L_{CD}^{[ext]}(u_{D,t}=j)}. \tag{B.5}$$

個人資料

姓 名：吳俊鋒

性 別：男

出生年月日：民國 68 年 4 月 5 日

籍 貫：台灣省雲林縣

學 歷：國立台灣大學數學系畢業(87 年 9 月~92 年 6 月)
國立交通大學電信研究所碩士班畢業(93 年 9 月~95 年 6 月)
國立交通大學電信研究所博士班(95 年 9 月~100 年 6 月)

論文題目：
中文: 基於多重敘述編碼理論之無線通訊系統的品質管理研究
English: QoS Control for Mult-Stream Voice over Mobile IP Networks

Publication List

Journal Papers

1. Chun-Feng Wu, Wen-Whei Chang and Yuan-Chuan Chiang, *"Perceptual-Based Playout Mechanisms for Multi- Stream Voice over IP Networks"* , IEICE Trans. on Information and Systems, Vol.E94-D,No.5, 2011 年 05 月
2. Chun-Feng Wu and Wen-Whei Chang, *"Iterative Source-Channel Decoding Using Information Symbol- Level Extrinsic"* , IEICE Trans. on Communications, Vol.E93-B, No.12, 2010 年 12 月
3. **(審核中**) Chun-Feng Wu, Wen-Whei Chang and Yuan-Chuan Chiang, *"Joint Playout and FEC Control for Multi-Stream Voice over IP Networks"* , JCIE, 2011
4. **(審核中**) Chun-Feng Wu and Wen-Whei Chang " *Symbol-based iterative decoding of convolutionally encoded multiple descriptions*" IET, 2011

Conference Papers

1. *Yung-Le Chang, Chun-Feng Wu and Wen-Whei Chang, " Joint Playout and FEC Control for Enhancing Perceived Quality of Multi-Stream Voice Communication", 2011 European Signal Processing Conference, 2011 年 8 月*
2. Yung-Le Chang, Chun-Feng Wu and Wen-Whei Chang, *"Joint Playout Buffer and FEC Adjustment for Multi-Stream Voice over IP Networks"* , 2010 International Workshop on Information Communication Technology, Bangkok, Thailand, 2010 年 08 月
3. Yen-Chang Pan, Chun-Feng Wu, and Wen-Whei Chang, *"Iterative Source-Channel Decoding Using Symbol-Level Extrinsic Information"* , International Conference on Information, Communications and Signal Processing, Macau, 2009 年 12 月
4. Chun-Feng Wu, Yung-Le Chang and Wen-Whei Chang, *"PERCEPTUAL-BASED PLAYOUT MECHANISMS FOR MULTI-STREAM VOICE OVER IP NETWORKS"* , European Signal Processing Conference, 2009 年 08 月
5. Kuang-Yi Yen, Chun-Feng Wu and Wen-Whei Chang, *"Iterative decoding of convolutionally encoded multiple descriptions"* , Data Compression Conference, 2009 年 03 月
6. Chun-Feng Wu, Cheng-Lung Lee, and Wen-Whei Chang, *"Perceptual-Based Playout Mechanisms for Multi-Stream Voice over IP"* , Interspeech '2007-Eurospeech, Antwerp, 2007 年 08 月
7. Chun-Feng Wu, and Wen-Whei Chang, *"PERCEPTUAL OPTIMIZATION OF PLAYOUT BUFFER IN VOIP APPLICATIONS"* , First International Conference on Communications and Networking in China (CHINACOM 2006), 2006 年 10 月
8. I-Te Lin, Chun-Feng Wu, Sin-Horng Chen, and Wen-Whei Chang, *"MULTIPLE DESCRIPTION QUANTIZATION FOR RECOGNIZING VOICE OVER PACKET NETWORKS"* , First International Conference on Communications and Networking in China (CHINACOM 2006), 2006 年 10 月

9. Chun-Feng Wu, I-Te Lin, and Wen-Whei Chang, *"ADAPTIVE PLAYOUT SCHEDULING FOR MULTI-STREAM VOICE OVER IP NETWORKS"* , 14th European Signal Processing Conference, 2006 年 09 月

專利

1.(Revised)雙通道語音傳輸系統及播放排程設計模組, 正文科技, 中華民國專利

2.(Revised)Two channel voice transmission system and playout schedule design, U.S.A. Patent

E S A
1896