8-2016

Joint Source-Channel Coding Optimized On End-to-End Distortion for Multimedia Source

Ebrahim Jarvis
ej7414@rit.edu

Joint Source-Channel Coding Optimized On End-to-End Distortion for Multimedia Source

by

Ebrahim Jarvis

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science
in Computer Engineering

Supervised by

Associate Professor Dr. Andres Kwasinski
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2016

Approved by:

Dr. Andres Kwasinski, Associate Professor
Thesis Advisor, Department of Computer Engineering

Dr. Sohail Dianat, Professor
Committee Member, Department of Electrical Engineering

Dr. Shanchieh Jay Yang, Professor
Committee Member, Department of Computer Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Joint Source-Channel Coding Optimized On End-to-End Distortion for Multimedia Source

I, Ebrahim Jarvis, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Ebrahim Jarvis

Date

Dedication

To my family

Acknowledgments

I would like to thank my advisor Dr. Kwasinki for his continuous support and help in my research. I would also like to thank Dr. Dianat and Dr. Yang for their valuable input and feedback.

Abstract

Joint Source-Channel Coding Optimized
On End-to-End Distortion for Multimedia Source

Ebrahim Jarvis

Supervising Professor: Dr. Andres Kwasinski

In order to achieve high efficiency, multimedia source coding usually relies on a use of predictive coding. While more efficient, source coding based on predictive coding has been considered to be more sensitive to errors during communication. With the current volume and importance of multimedia communication, minimizing the overall distortion during communication over an error-prone channel is critical. In addition, for real-time scenarios it is necessary to consider additional constraint such as fix and small delay for a given bit rate. To comply with these requirements, we seek an efficient joint source-channel coding scheme.

In this work, end-to-end distortion is studied for a first order auto regressive synthetic source that represents a general multimedia traffic. This study reveals that predictive coders achieve the same channel-induced distortion performance as memoryless codecs when applying optimal error concealment. We propose a joint source-channel system based on incremental redundancy that satisfies the fixed delay and error-prone channel constraints and combines DPCM as a source encoder and a rate-compatible punctured convolutional (RCPC) error control codec. To calculate the joint source-channel coding rate allocation that minimizes end-to-end distortion, we develop a Markov Decision Process (MDP) approach for delay constrained feedback Hybrid ARQ, and we use a Dynamic Programming (DP) technique. Our simulation results support the improvement in end-to-end distortion

compared to a conventional Forward Error Control (FEC) approach with no feedback.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The ever-growing demand for applications based on multimedia communications requires more efficient use of communication resources. However, multimedia sources, like video, are communication bandwidth intensive. Even though the new technology may support the communication channels with ever-increasing bandwidth, compression of the information source is still play an essential role in communication. This is due to the fact that with the rate of data is growing at higher rate of bandwidth growth that force us to achieve a more efficient bandwidth utilization. For example, in terrestrial TV broadcasting, online gaming, virtual reality, video conferencing, satellite TV and cable TV applications, more high-resolution video channels can be delivered within the allocated bandwidth if each video signal is compressed for transmission. Therefore, the development of an efficient framework to do source compression has attracted considerable research interest.

To address the multimedia source compression problem, numerous techniques have been proposed. Predictive coding technique has shown to be highly efficient in compressing the source data. However, there is widely spread misconception among some researchers, who questions the performance of predictive coding over a noisy channel. They suggest that presence of a transmission error at the decoder would cause the performance of these coders degrades considerably due to error propagation.

To increase throughput in data transmission various transmission schemes have been studied. Numerous studies suggest that feedback-based transmission system such as Hybrid ARQ with incremental redundancy can provide considerable gains in throughput over pure (no feedback) Forward Error Control (FEC) system. This occurs because, Hybrid ARQ system assumes there is a significant probability of correctly receiving data protected with a weak channel code, and hence it is not necessary to add large redundancy in every transmission. Despite these gains application of feedback based schemes to real-time communication has been limited due to the challenges presented by inherent delay related problems regarding transmission of incremental redundancy.

1.2 Contributions In This Thesis

The primary goal of this thesis is to address the source-channel rate allocation problem so as to the end-to-end distortion will be minimized for the communication of a multimedia source. First, we propose a synthetic source that abstracts the fundamental components of video. This is because, working with video is difficult due to complexity of source model. Furthermore, in the proposed source, we assume that the first sample is memoryless and the rest is DPCM coded using the memory between the samples. The precise and novel mathematical expressions have been used to indicate the end-to-end behavior of such system. Furthermore, our simulation results show that the performance of predictive coders in the presence of errors not only will not be catastrophic but also with the proper use of error concealment it leads to having the same performance as the memoryless system. Second, we address the joint source-channel rate problem in the incremental redundancy Hybrid ARQ mechanism. We develop a novel approach based on the notion of codetypes resulting

from modeling our problem with a Markov Decision Process (MDP). A Dynamic Programming algorithm is used to find the optimal source and channel rates. Our simulation results illustrate that our proposed method outperform the same system with no feedback.

Chapter 2

Background

In this chapter, we discuss various technologies and techniques relevant to this work. The chapter is divided into five parts. First, we will briefly introduce previous studies that deals with similar problem. Then we study the basic structure of predictive coding. In Section 2.2.3 we will investigate the statistical properties of the source. Then, we focus our study on differential pulse code modulation (DPCM). We also describe the overall features of DPCM encoder and decoder. In third part of this chapter we will discuss the idea behind variable rate codes. We describe the channel model and decoding mechanism, recognizing that understanding the performance of such system is essential to design joint source-channel codes. In the fourth part of this chapter, we extend the study of RCPC codes, and look into joint source-channel coding. In the last part of this chapter, we discuss the use of incremental redundancy bits in the joint source-channel coding system.

2.1 Previous Work

Of the works with similar objective as ours, the work most closely related to this thesis is that by Chou et al. [2], which addresses the problem of streaming packetized media over a lossy packet network by rate-distortion optimization. Assuming that each data unit of the video stream is independent of one another, the authors showed that the problem of rate-distortion optimized streaming can be reduced to the problem of solving the cost-error

optimization for a single unit data. Furthermore, the authors derived a practical algorithm to address a variety of transmission scenarios such as sender driven or receiver driven transmission, and streaming over the best effort network. This algorithm used a Markov decision process framework, where the cost-error model differentiate different transmission scenarios from each other. The expected rate-distortion optimization was done by using a general iterative algorithm to form a convex hull that can be solved by Lagrange multiplier optimization technique. Furthermore, authors assume a fix short delay (up to several seconds) in the beginning of the data transmission that is independent of the length of the presentation. In addition, their rate optimization system is primarily based on the usage of buffer and playback on the receiver side, where incremental redundancy can be more suitable for a more constrained delay and buffer space. By considering the distortion that is suffered by each single data unit independent of each other and developing an error-cost model based on each frame, concluding the optimal rate-distortion will be approximation of the optimal scenario (scenario that studies the affect of error on current frame to the future frame).

The work [3], studies the performance of DPCM predictive coding method under an error prone channel. This research shows that DPCM works reasonably well to compress the signal. However, the performance degrades considerably when errors occur during transmission. The authors concluded that this is due to error propagation at the decoder, which originates from the DPCM mechanism that relies on previous samples at decoder. The authors suggest that this leads to a catastrophic performance with the even single error during transmission. Furthermore, this study showed that a low-rate feedback channel would improve the overall performance.

The work of Kwasinski et al. [4], presented a joint-source channel coding scheme based on incremental redundancy and feedback (Hybrid ARQ). A real-time source is transmitted

under strict delay constraints. The proposed scheme works with a constant bit rate and delay (one frame delay). In order to add incremental redundancy to the frame when its coded, the source is encoded with a lower bit rate. The underlying design technique uses a Markov chain by introducing various code types, where each code type describes the different compositions of the frames. In the work, there are three possible outcomes to a transmission. The authors assumed three successively transmitted frames, referred in time as previous, current and next frames, and the corresponding source blocks as previous, current and next source blocks. The feedback-based transmission scheme is described as, when the previous block is received with errors, the current block will contain the codeword for the current frame and incremental redundancy for the previous one and will lower the channel code rate when possible (the maximum number of different code rates not reached). If the current block was received without errors after channel decoding, then the current frame will consist entirely of the current frame's bits without incremental redundancy. The result of the work [4], showed distortion value decrease compare to no feedback transmission and hence, 28% increase in the simultaneous calls that can be supported on the CDMA network.

In [5], Zhihai He et al. shed more light on the source and channel distortion modeling. The authors developed a rate-distortion model for DCT-based video by considering the macroblock intrarefreshing rate. Using this R-D model, the authors estimated the distortion given the bit rate and intrarefreshing rate. Furthermore, in this work the behavior of channel error on video coding and transmission is described through closed form expressions. However, the accuracy of their distortion estimation is largely based on the size of source input samples.

2.2 Predictive Coding

A data compression system could be translated into any scheme that operates on source data to eliminate possible redundancies so that only those values needed to reproduction are preserved. There are two main approaches to address redundancy issues namely, time and frequency domain techniques. In the first category, compression is performed through prediction of source samples by exploiting the fact that source samples are serially correlated. Therefore, predictive coding system codes the difference between the source sample and the predicted value instead of coding the source. Common time domain coders include differential pulse code modulation (DPCM) and delta modulation (DM). In frequency domain methods, first, the source spectrum is divided into several frequency bands through subband filtering. Then a procedure called frequency decomposition is applied. This is followed by assigning quantizers to the subbands. If we increase the number of the subbands, we will get less correlation between the subband. Wavelet, subband, and transform coders such as traditional JPEG, MPEG- 1, MPEG-2, H.261, H.263, and H.265 are standard for frequency-based domain coding. Scalar quantization of each source sample independently is a suboptimal method, when there is time correlation between samples. The correlation in time between samples may appear as two samples that are close in time show little change in value. Speech and audio sources both show time correlation between samples because the physical mechanism by which they are generated create a signal that changes smoothly over time. The same can be expressed about image and video sources for time and spatial correlations. When two neighbor samples are correlated, one can use the quantized information from one sample and reduce the amount of data needed to code the other one.

One of the methods that exploit the correlation between samples (memory) is predictive coding. This type of coding is shown in the Figure 2.1, where $\hat{x}[n]$ is subtracted from

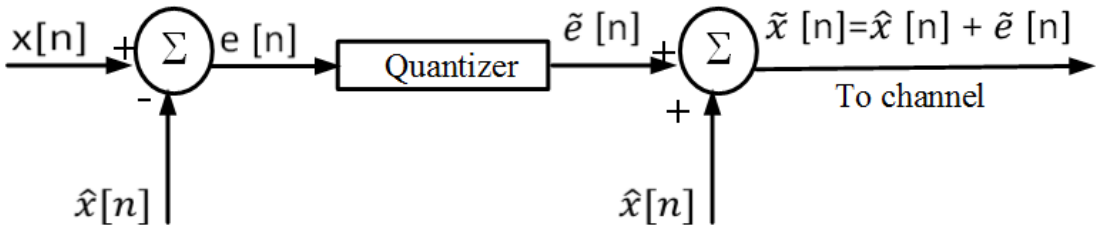

Figure 2.1: predictive source encoder

the source sample $x[n]$ before quantization. The same quantity is added to the quantized difference $\tilde{e}[n]$ to generate the reconstructed output $\tilde{x}[n]$. An important property of predictive coding is that the reconstruction error equals the quantization error. The following equations prove this fact:

$$\begin{aligned} e[n] &= x[n] - \hat{x}[n] \\ \tilde{x}[n] &= \tilde{e}[n] + \hat{x}[n] \\ x[n] - \tilde{x}[n] &= e[n] + \hat{x}[n] - (\tilde{e}[n] + \hat{x}[n]) \\ &= e[n] - \tilde{e}[n]. \end{aligned} \tag{2.1}$$

The best choice of predicted value as noted $\hat{x}[n]$ in the figure 2.1, to produce a difference $e[n]$, can be yield using two main methods namely, maximum a priori probability (MAP) and linear minimum squared estimate (LMMSE). MAP obtains $\hat{x}[n]$ from a group of neighboring samples. Let us denote the adjacent samples by $g[n]$. The MAP estimate $\hat{x}[n]$ of $x[n]$ is given by,

$$\hat{x}[n] = arg\, max_x\, p(\frac{x}{g[n]})$$

The optimization solution calculated by MAP is often considered as too slow due to the complexity and number of operations so that one may take into account a suboptimal settlement to this problem. In the case of adjacent samples in the source sequence showing high dependency (strong correlation), the simplest solution is the previous sample $x[n-1]$. Note that under this condition the difference $e[n] = x[n] - x[n-1]$ is most likely to be small. The main issue with MAP is that to predict $x[n]$ the solution $x[n-1]$ does not take into consideration the correlation between samples.

As mentioned to estimate $x[n]$ we can use the linear minimum squared estimate (LMMSE). This method uses a linear combination of samples in the neighbor group $g[n]$. The following section will review the design of the optimal LMMSE for an autoregressive source as will be considered in this thesis.

2.2.1 Optimal Linear Prediction for Autoregressive Sources

From [6], let $...., x[-1], x[0], x[1], x[2], .., x[n]$ be a zero mean, stationary source sequence. The linear prediction of $x[n]$ from the past L samples can be implemented as,

$$\hat{x}[n] = \sum_{k=1}^{L} a_k \tilde{x}[n-k], \tag{2.2}$$

where $a_1, a_2, ..., a_L$ are the prediction filter coefficients. The solution that minimizes the MSE $E\big[\hat{x}[n] - x[n]\big]^2$ for the vector of prediction coefficients $\mathbf{a}^* = (\mathbf{a_1}^*, \mathbf{a_2}^*, \ldots, \mathbf{a_L}^*)$, comes from the orthogonality principle [7] that the error $x[n] - \hat{x}[n]$ should be orthogonal to the past sample values,

$$E\big[(x[n] - \hat{x}[n])x[n-k]\big] = 0, \quad k = 1, 2, .., L. \tag{2.3}$$

The orthogonality principle generates the Yule-Walker equation [6],

$$\phi_x{}^{(L)}\mathbf{a}^* = \phi_x, \tag{2.4}$$

where in 2.4, $\phi_x(m) = E[x[n]x[n-m]]$ is the autocorrelation function of the source with different lag m, $m = 1, 2, ..., L$; and $\phi_x = (\phi_x(1), \phi_x(2), ..., \phi_x(L))^T$. Then for the L-th order autocorrelation matrix of the source we have,

$$\phi_x{}^{(L)} = \{\phi(|k-m|)\}_{k,m=0}^{L-1}. \tag{2.5}$$

We can express 2.4 in an expanded form of the Yule-Walker that shows the individual elements as

$$\begin{bmatrix} \phi_x(0) & \phi_x(1) & \dots & \phi_x(L-1) \\ \phi_x(1) & \phi_x(0) & \dots & \phi_x(L-2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_x(L-1) & \phi_x(L-2) & \dots & \phi_x(0) \end{bmatrix} \times \begin{bmatrix} a_1^* \\ a_2^* \\ \vdots \\ a_L^* \end{bmatrix} = \begin{bmatrix} \phi_x(1) \\ \phi_x(2) \\ \vdots \\ \phi_x(L) \end{bmatrix}$$

The MMSE can be described as,

$$\begin{aligned} e_{min}^2 &= min\ E[|x[n] - \hat{x}[n]|^2] \\ &= \phi_x(0) - \phi_x^T a^0 \\ &= \phi_x(0) - \phi_x^T(\phi_x^{(L)})^{-1}\phi_x \end{aligned} \tag{2.6}$$

Note that the e_{min}^2 is always less than $\phi_x(0)$ since the autocorrelation matrix ϕ_L is always non-negative definite. This condition will assure that the inverse $(\phi_x^{(L)})^{-1}$ is also definite positive then $\phi_x^T(\phi_x^{(L)})^{-1}\phi_x > 0$ for any vector ϕ_x. Therefore, the minimum error variance is less than the source variance.

2.2.2 Source Key Properties

To study the source statistical properties, we are assuming the input source $x[n]$ follows a first autoregressive, AR(1), source model, we can express the source model as,

$$x[n] = \rho x[n-1] + z[n], \tag{2.7}$$

where the innovation process $z[n]$ forms a sequence of independent and identically distributed random-valued samples following a Gaussian distribution with zero mean and variance σ_z^2. The source probability distribution follows a first-order Markov (Markov-1) process, since the current random variable $x[n]$ only depends on the previous sample $x[n-1]$ and the innovation process $z[n]$ and it does not depend on other past samples.

The source is encoded by operating on blocks of M samples $\{x[0], x[1], \ldots, x[M-1]\}$. We assume the first sample ($x[0]$) is encoded using a memoryless source encoder and the rest of samples are encoded using predictive coding encoder (DPCM). This mechanism models the main components of the video encoder. In the video encoder the first frame in a block of frames is encoded independent from other frames and without considering the time-correlation (as I frame), and the rest of the frames are encoded by exploiting the temporal correlation with the previous frames (as P frames). Following a simplified model for motion compensated video coding, the first sample is encoded at a rate R_0 bits per sample using a scalar memoryless encoder and all subsequent $M-1$ samples are encoded using a DPCM encoder with a rate of R_n bits per sample for sample $x[n]$. Note that through recurrent use of (2.7) any source sample can be written in terms of the innovation process

samples only:

$$\begin{aligned} x[n] &= \rho x[n-1] + z[n] \\ &= \rho(\rho x[n-2] + z[n-1]) + z[n] \\ &= \rho^2 x[n-2] + \rho z[n-1] + z[n] \\ &= \sum_{j=0}^{n} \rho^j z[n-j]. \end{aligned} \tag{2.8}$$

Since $x[n] = \sum_{j=0}^{n} \rho^j z[n-j]$, a linear combination of Gaussian random variables, $x[n]$ is also Gaussian. This is, $x[n] \sim \mathcal{N}(0, \sigma_x^2)$. From this result, we can now calculate the expected value for $x[n]$,

$$E[x[n]] = E\Big[\sum_{j=0}^{n} \rho^j z[n-j]\Big] = \sum_{j=0}^{n} \rho^j E[z[n-j]] = 0 \tag{2.9}$$

The autocorrelation of $x[n]$ can be derived also from (2.8) by considering samples that occur in the very distant future after the source was turned on in the distant past, this is,

samples $x[n] = \sum_{j=0}^{\infty} \rho^j z[n-j]$. In this case, the autocorrelation is

$$\begin{aligned}
\phi(k) &= E[x[n]x[n-k]] \\
&= E\Big[\Big(\sum_{i=0}^{\infty} \rho^i z[n-i]\Big)\Big(\sum_{j=0}^{\infty} \rho^j z[n-k-j]\Big)\Big] \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \rho^{i+j} E[z[n-i]z[n-k-j]] \\
&\overset{(\#)}{=} \sum_{j=0}^{\infty} \rho^{k+2j}\sigma_z^2 \\
&= \rho^k \sigma_z^2 \sum_{j=0}^{\infty} \rho^{2j} \\
&= \sigma_z^2 \frac{\rho^k}{1-\rho^2},
\end{aligned}$$

where equality (#) follows because the innovation process samples $z[n]$ are uncorrelated,

$$E\Big[z[n-i]z[n-k-j]\Big] = \begin{cases} \sigma_z^2, & \text{if } i = k+j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

Now, since by definition $\sigma_x^2 = \phi(0)$, it follows that

$$\sigma_x^2 = \frac{\sigma_z^2}{1-\rho^2}. \tag{2.11}$$

Therefore, the covariance of the source can be expressed as,

$$\phi(k) = \sigma_x^2 \rho^k, \quad k \geq 0. \tag{2.12}$$

Using this result and that the autocorrelation function is even, it follows that

$$\phi(k) = \sigma_x^2 \rho^{|k|}, \quad k \in Z. \tag{2.13}$$

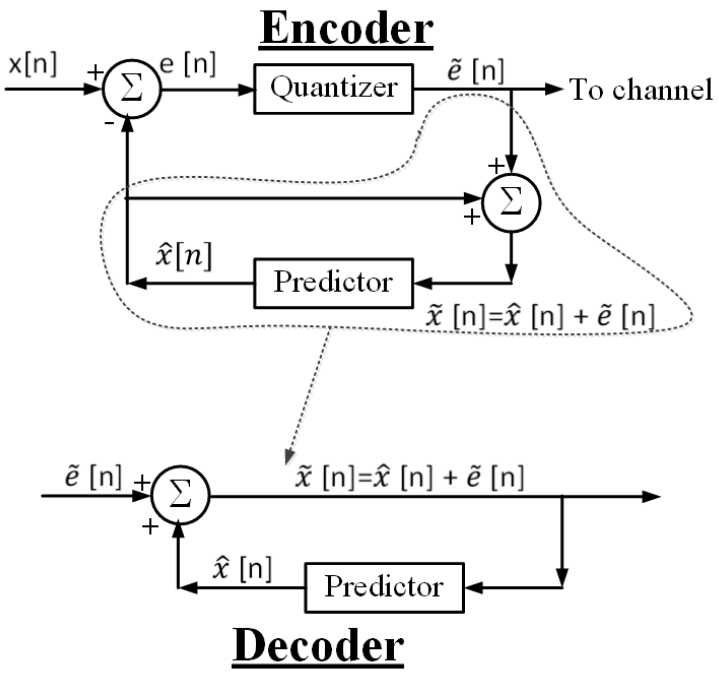

Figure 2.2: DPCM encoder and decoder

Finally, we can conclude that $x[n] \sim \mathcal{N}\big(0, \sigma_z^2/(1-\rho^2)\big), \quad \rho \geq 0.$

2.2.3 DPCM System Description and Statistical Overview

The principle of differential pulse code modulation (DPCM) is based on ideas from predictive coding. DPCM was first introduced in [8] to improve the efficiency of communication systems by taking advantage of the correlation between the source samples. DPCM is a compression method which is even today widely used. Where other compression methods may offer better compression ratio, DPCM offers simplicity results in low cost implementation, which is highly efficient especially in high speed compression.

Figure 2.2, shows the architecture of the DPCM encoder and decoder. It can be seen that the DPCM main components are comprised of two main components: predictor and a quantizer. In DPCM the prediction residual $e[n]$, is the difference between the current

source $x[n]$ and a predicted value $\hat{x}[n]$, then it quantized to $\tilde{e}[n]$, which is then transmitted to the destination. The predicted value $\hat{x}[n]$ is obtained from a linear combination of the past L values of the source $\tilde{x}[n-1], \ldots, \tilde{x}[n-L]$, which are derived from $\tilde{e}[n]$ accordingly. As a result, the current reconstruction of $\tilde{x}[n]$ is the sum of the predicted value $\hat{x}[n]$ and the quantized prediction residual $\tilde{e}[n]$. Note that the reconstruction error can be yield the same way as discussed in predictive coders in (2.1).

As Figure 2.2 illustrates, DPCM uses a linear prediction filter with the input-output relation given by,

$$\hat{x}[n] = \sum_{k=1}^{L} a_k \tilde{x}[n-k], \tag{2.14}$$

for a predictor of order L with coefficients a_k. In the particular case of a first order autoregressive, AR(1), source, $L = 1$ and we have $\hat{x}[n] = a_1 \tilde{x}[n-1]$. From (2.7), it is well known ([6]) that to minimize the mean square prediction error the coefficient a should be, $a = \rho$, making the optimum linear predictor be $\hat{x}[n] = \rho \tilde{x}[n-1]$.

The main challenge for DPCM comes from the fact that the prediction is made from the past values which are not available at the destination because their reconstructed values contains error. If the predictor works by storing the past M values in the memory and the destination tries to reconstruct the samples, the reconstruction error will accumulate. Therefore the DPCM predictor at encoder side uses the source reconstructed values which are the same as those at the destination.

2.2.4 DPCM Coding Error and Gain

The prediction residuals ($e[n]$) are a scalar source signal, with values quantized independently of each other, as shown in Figure 2.2. In DPCM linear prediction is obtained similar to LMMSE mechanism. As shown in (2.6) coding error for LMMSE system is the mean

squared value of the prediction residual and is closely related to the variance of the source. However, the coding error for DPCM is not the same value as the e_{min}^2 in (2.6), since in DPCM the prediction method is done differently. In (2.6), the prediction takes place on the actual past values while in DPCM it takes place on previously reconstructed values.

If we assume that the transmission is done at high rates while the quantization error is small then we can approximate that the predictor coefficients are calculated using the source covariances $E\big[x[n]x[n-l]\big], \quad l = 0, 1, \ldots, L$. Also $\hat{x}[n]$, is approximately derived from the same past values. Using this approximation the coding error can be approximates as [6],

$$E\Big[\hat{e}[n]^2\Big] \approx \phi_x(0) - \phi_x^T(\phi_x^{(L)})^{-1}\phi_x. \tag{2.15}$$

The gain in DPCM coding over PCM can be defined as,

$$G_{DPCM/PCM} = \frac{D_{PCM}}{D_{DPCM}}. \tag{2.16}$$

Which is derived from the ratio of the their corresponding coding errors. Since they both have the same unit variance distortion versus rate function the ratio will be the ratio of their source variance which is,

$$G_{DPCM/PCM} = \frac{\phi_x(0)}{\phi_x(0) - \phi_x^T(\phi_x^{(L)})^{-1}\phi_x}. \tag{2.17}$$

Note that the gain will be more than one in the case of the source with memory.

2.3 Rate-Compatible Channel Codes

Rate-compatible channel codes offer flexibility in adapting the channel code rate when it is used for joint-source channel coding applications. This is because in these applications the channel code rates need to be modifiable as the source rate changes. In this channel

codes, the members are derived from the lowest rate code which is called the "mother code". The higher rate codes in the rate-compatible family can be encoded using puncture tables that operates on the mother code. For each RCPC code, there is a puncturing table to determine the bits that are being removed from the output of mother code encoder. The puncturing table consists of rows which related to each output encoder and columns that show the order of bits. The zero element in the table means that bit from the mother code to be punctured and one indicates that the mother code in that position not to be punctured. Note that since the number of columns are limited, the puncturing operation is repeated periodically. The length of this repetition is called the puncturing period for RCPC codes. The rate compatible restriction over puncturing table ensures that all code bits of high rate codes are used by lower rate codes. If we express that the mother code is extended by puncturing a low rate $\frac{1}{N}$ with period P and a as the punctured matrix and l as the matrix sequence number, latter restriction can be expressed as,

$$\text{if} \quad a_{ij}(l_0) = 1 \quad \text{then} \quad a_{ij}(l) = 1 \quad \text{for all} \quad l \geq l_0 \geq 1, \tag{2.18}$$

or equivalently,

$$\text{if} \quad a_{ij}(l_0) = 0 \quad \text{then} \quad a_{ij}(l) = 0 \quad \text{for all } l \leq l_0 \leq (N-1)(P-1). \tag{2.19}$$

Rate-compatible channel codes can be constructed from any linear code, but it is more efficient and straightforward when based on the use of convolutional codes in which case they are called rate-compatible convolutional codes (RCPC).

Figure 2.3 shows an example of generating various coding rates using the same convolutional code. The figure shows three codes are derived with coding rates decreasing from bottom to top. The mother code in this example has a rate of $1/2$. All RCPC codes that are derived from this mother code will have a coding rate higher than $1/2$. This is because with

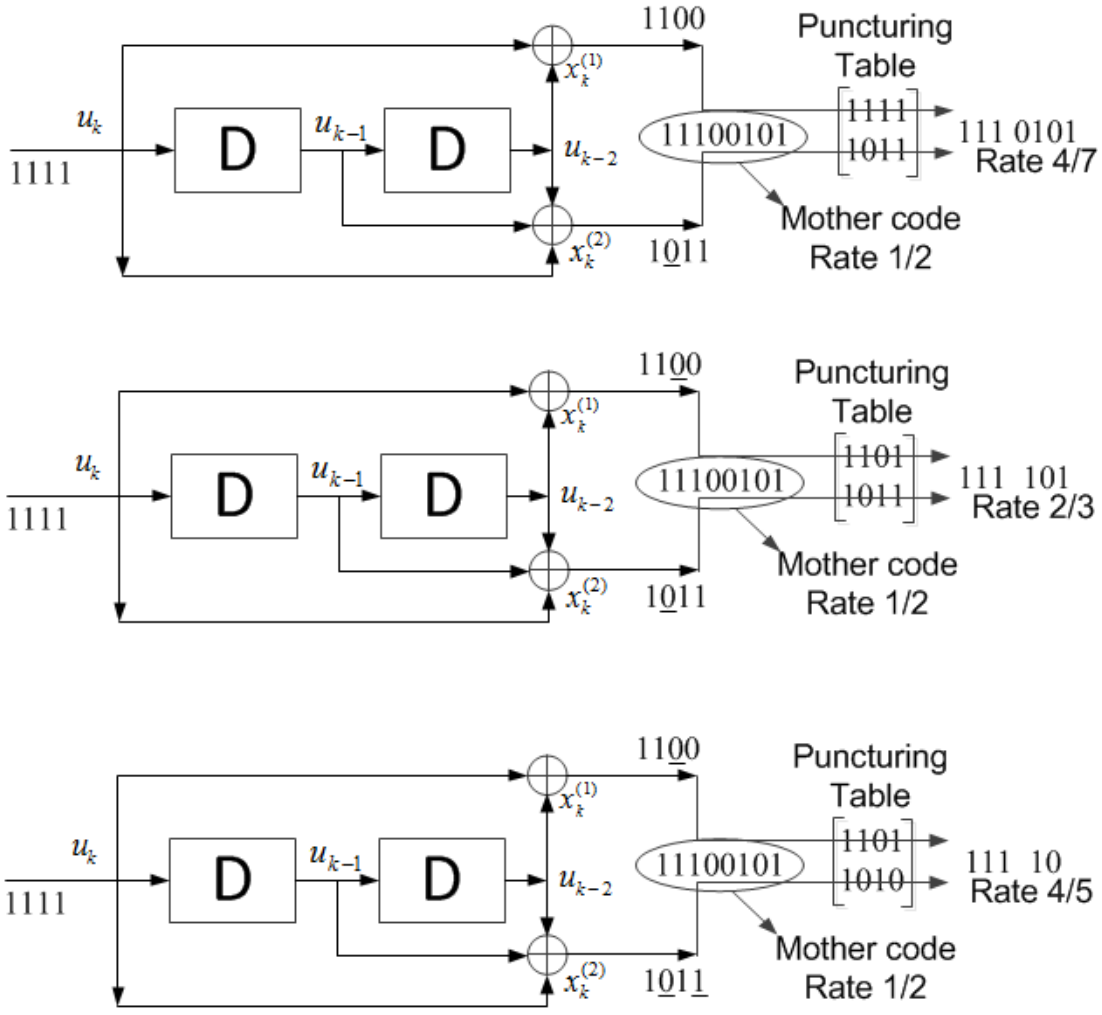

Figure 2.3: Rate-Compatible Punctured Convolutional Channel Code example.

the zero puncturing we get the rate which is equal to the rate of the mother code and any positive number of puncturing results in higher coding rate compare to the mother code. The top code has the largest rate in the example of Figure 2.3 at $4/7$, and is derived from puncturing one bit out of four bits. The punctured bit corresponds to an entry of zero in puncturing table corresponding to the output of the bottom line of the encoder. The deleted bit is shown as underlined. For the example of the top code in Figure 2.3, the puncturing table is,

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

where it can be seen that the punctured bits are the second, sixth, tenth, etc. The puncturing period here is four.

The decoding of RCPC codes can be done the same way as the other punctured convolutional codes. As a matter of fact, the common way of implementing the decoder is to use Viterbi decoder. As it is discussed in [9], the soft decision Viterbi algorithm can be utilized for decoding all the RCPC codes with various puncturing rules.

2.3.1 Performance of RCPC Codes

The performance of RCPC codes can be calculated in the same way as other convolutional codes. Therefore, it can be measured based on the probability of making a decoding error. The applied puncturing rules will affect the strength of the code. Higher rate codes are weaker regarding correcting errors. This point is shown in Figure 2.4, where the BER is plotted as a function of channel SNR for different RCPC codes from the same family.

The analysis of the performance of RCPC codes depends on the channel model and decoding mechanism, since they affect the probability of making a decoding error. It was assumed that the transmission was done over an AWGN channel and decoding with soft

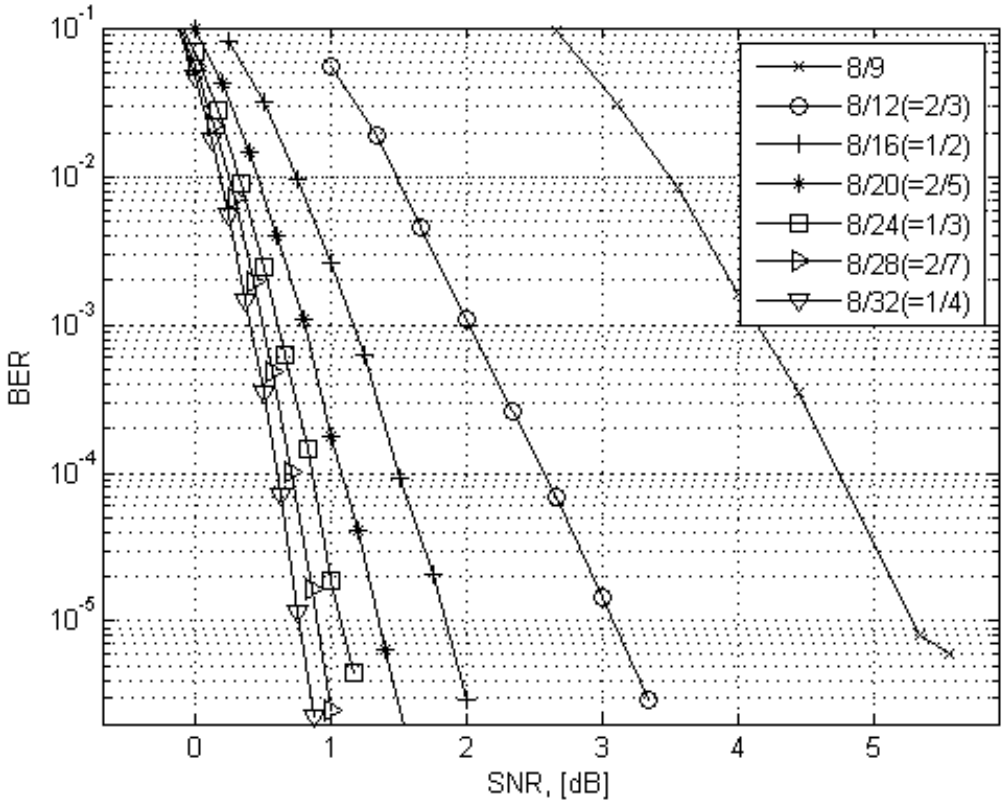

Figure 2.4: Performance of different codes from the same family of RCPC codes with mother code rate 1/4, memory 8 and puncturing period 8 (reproduced from [1]).

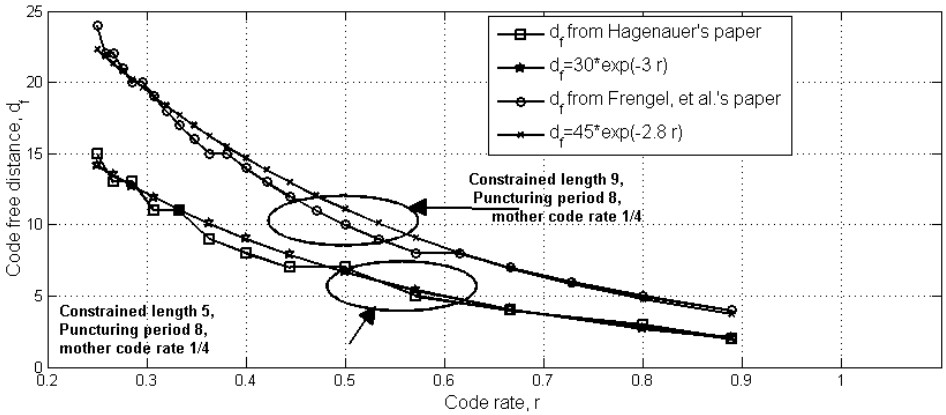

Figure 2.5: Free distance as a function of channel coding rate r for two families of RCPC codes (reproduced from [1]).

metric Viterbi decoder. Assuming, that the input to the encoder is the all zero sequence. The probability of making a decoding error is calculated as the probability of not choosing the right path (all zero in this case) in the corresponding trellis diagram of the decoder. Note that the input can be a stream of bits with not a fixed length necessarily. Therefore, the performance is measured as the probability that the path which separates and rejoins all the zero path for the first time has a path metric larger than the all zero path. This problem can be related to calculating the probability that the Hamming distance between one path and the zero path is d. According to [1], this probability can be expressed as,

$$P_e(d|\gamma_b) = \frac{1}{2}\,\text{erfc}\sqrt{dr\gamma_b}, \tag{2.20}$$

where r is the channel code rate, $\gamma_b = E_b/N_0$ is the received SNR per bit and erfc (γ) is the complementary error function defined as,

$$\text{erfc}(\gamma) = 2/\pi \int_{\gamma}^{\infty} e^{-u^2} du. \tag{2.21}$$

Note that there could be more than one path with distance d. Denoting the minimum

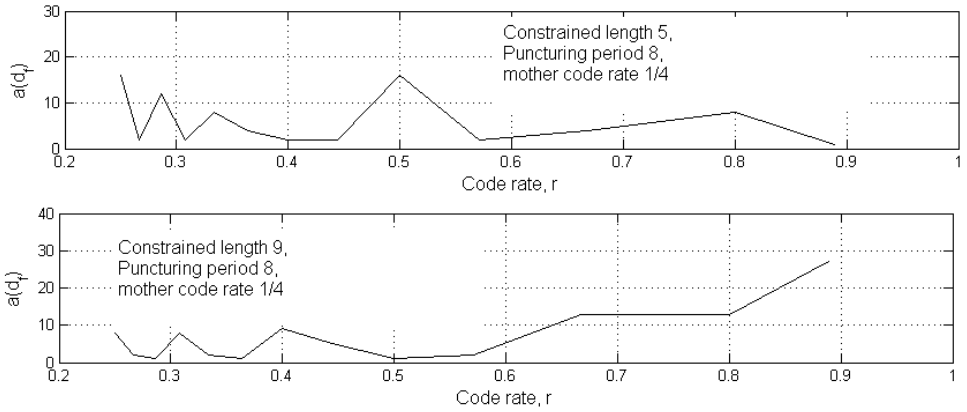

Figure 2.6: Number of error events with Hamming weight d_f as a function of channel coding rate r for two families of RCPC codes (reproduced from [1]).

distance as d_f, the upper bound probability of decoding error can be written as,

$$P(\gamma_b) \leq \sum_{d=d_f}^{\infty} a(d) P_e(d|\gamma_b), \tag{2.22}$$

where $a(d)$ is the number of paths with Hamming distance d from the all zero path. Note that in RCPC codes both d and $a(d)$ depends on the channel code rate. Figure 2.5 shows this relation in more detail where the free distance decrease with increasing in channel code rate. This observation can be expressed as,

$$d_f = ke^{-cr}, \tag{2.23}$$

where k and c are constants. Figure 2.6 illustrates the fact that the behavior of the function $a(d)$ can not be approximated as a simple function of channel coding rate (r).

Finally, as we shall see in section 2.4, rate compatible channel codes, by providing a simple mechanism to adapt channel coding rate while using the same decoder, they are very useful when implement joint-source channel coding techniques. Specifically, in the case of the channel with feedback, it provides an implementation of unequal error protection. In

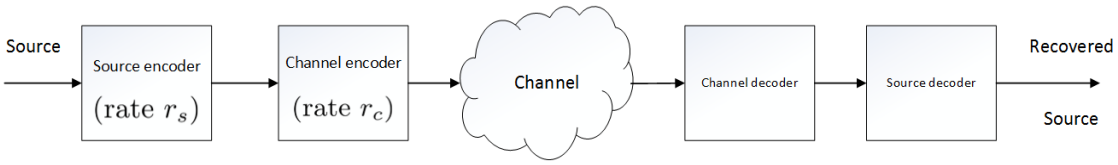

Figure 2.7: Block diagram of a joint source and channel coding scheme

other words, we tend to transmit the source bits with the highest channel rate, where the SNR is high. Using this method in conjunction with the technique that is called "Incremental Redundancy" results in efficient multimedia communication. Incremental redundancy transmission scheme heavily relies on the use of flexible channel codes. Incremental redundancy technique will be discussed in more detail in section 2.5.

2.4 Joint Source-Channel Coding

Shannon's source-channel separation theorem states that, in point-to-point communication systems, source coding (compression) and channel coding (error protection) can be performed separately and sequentially, while maintaining optimality. In many practical applications, the conditions of the Shannon's separation theorem can not be used. This is due to the optmality condition remains true only in the case of asymptotically long block lengths of data. Thus, significant interest has developed in various schemes of joint source-channel coding.

The most shared and simple way to design a joint source-channel coding system is by implementing each source and channel separately while configuring them to work jointly together. In this approach, called concatenated joint source and channel coding we connect separate source and channel coding blocks serially but their operating parameters are jointly optimized. As figure 2.7 shows, while, the source and channel encoding blocks are designed separately they are set to work jointly with each other by choosing the operating

parameters as a single coding unit. Then for this purpose, the source and channel codes are chosen such that they can be easily modified. Two of the common mechanisms employed are embedded source codecs and rate-compatible punctured channel codecs.

The primary goal in designing a concatenated joint source-channel is to assign a proper rate to the channel and source encoder. The resource allocation for joint source and the channel coding is done for a given channel signal to noise ratio (SNR). In addition, it is assumed that the total number of bits transmitted through the channel is fixed W bits.

The main goal regarding the joint allocation of the source and channel coding rate is to minimize the end-to-end distortion. The end-to-end distortion is the combination of the source coding distortion and the channel-induced distortion. Source coding distortion is related to the quantization and compression processes, and channel-induced distortion is caused by channel errors that occur during transmission and cannot be corrected by the channel coding process. As such, the end-to-end distortion can be mathematically written as,

$$D(r_s, r_c, \gamma) = D_F P_r(\gamma) + D_S(r_s)\Big(1 - P_r(\gamma)\Big), \tag{2.24}$$

where r_s is the source encoding rate measured in bits/sample, r_c is the channel coding rate, γ is the channel SNR, D_F is the total distortion when an error occurs during transmission and cannot be corrected by the channel code, $P_r(\gamma)$ is the probability of having post-decoding channel errors and D_S is the source encoding distortion.

Note that in (2.24), the possibility of not having an error during transmission is $1 - P_r(\gamma)$, in which case the end-to-end distortion is equal to the source encoding distortion $D_S(r_s)$ only. When an error happens in the channel during transmission and the channel decoder is unable to fix it part or all of the source encoded data needs to be discarded and the

lost information needs to be estimated or concealed by an error concealment mechanism. In this situation, the distortion introduced is the sum of the source distortion, and channel-induced distortion denoted as D_F.

The distortion D_F value is usually observed to be much larger than D_S, and it is related to various variables, including the level of compression, the algorithm used for error concealment and the source statistics. In some cases the value of D_F can be derived from statistical considerations. In the absence of any extra information, it is known that the optimal error concealment consists of replacing the lost source samples with their expected value. In the case of a memoryless Gaussian source with zero mean, the error concealed value is $\hat{s} = E[s] = 0$ and D_F is,

$$D_F = E\Big[(s - \hat{s})^2\Big] = 1. \tag{2.25}$$

The quality of communication perceived by the end user depends on the reconstructed samples at the receiver side, which also rely on the overall end-to-end distortion. To receive a better quality, then, the goal is to minimize the end-to-end distortion by appropriate choice of source coding rate (r_s) and the channel coding rate r_c. Figure 2.8 illustrates the typical plot of the end-to-end distortion as a function of channel SNR. It can be observed in Figure 2.8, that when the channel SNR is high, the probability of post channel decoding error $(P_r(\gamma))$ is very low, close to zero $P_r(\gamma) \approx 0$, and the value of $D(r_s, r_c, \gamma)$, using (2.24) can be expressed as,

$$D(r_s, r_c, high\,\gamma) \approx D_S(r_s). \tag{2.26}$$

At low SNR the probability of post channel decoding error $(P_r(\gamma))$ is close to one, so the

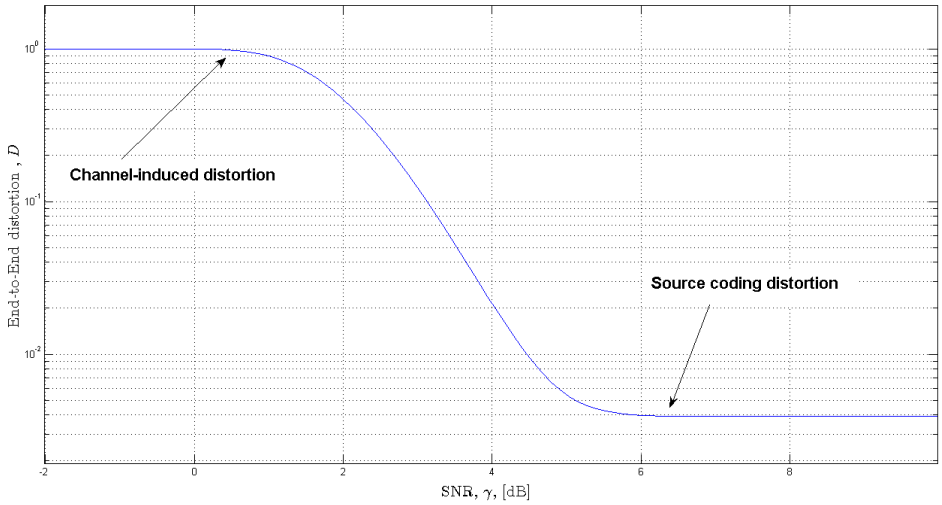

Figure 2.8: A typical end-to-end distortion curve as a function of SNR

value of the end-to-end distortion will be,

$$D(r_s, r_c, low\,\gamma) \approx D_F. \tag{2.27}$$

It can be seen that between the high and low SNR regions there is a transition, whereby decreasing the SNR value the end-to-end distortion value will increase from $D_S(r_s)$ to D_F. Since the total transmitted bits are fixed then by decreasing the SNR, we have the situation where the number of uncorrected errors increases. Figure 2.9 and 2.10, illustrate the typical results when plotting the end-to-end distortion as a function of channel SNR. These figures provide insight into the basic trade offs associated with joint bit rate allocation for concatenated JSCC. The mechanism linking source and channel coding rate implies that choosing a smaller channel coding rate will result in adding more redundancy and having more channel error protection, but it will also dictate using a smaller source coding rate.

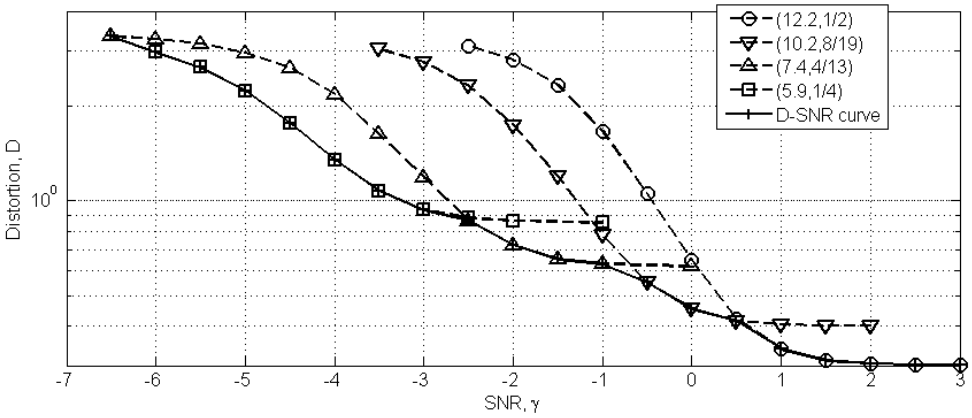

Figure 2.9: The single-mode D-SNR curves and the resulting D-SNR curve for a speech source(reproduced from [1]).

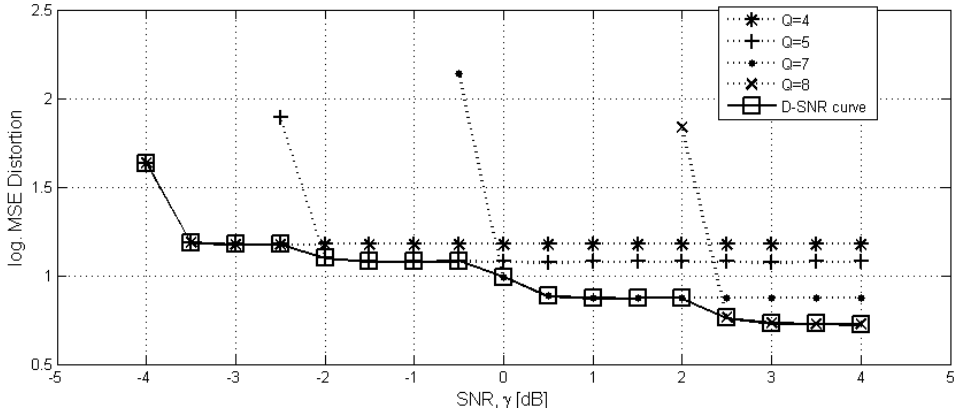

Figure 2.10: The single-mode D-SNR curves and the resulting D-SNR curve for a video source(reproduced from [1]).

The source and channel coding rates in concatenated joint source-channel coding are jointly chosen for a given SNR value. In a practical scenario, the possible choices for coding rates are selected from a finite number of discrete choices. Let $r_s = \{r_{s1}, r_{s2}, \dots\}$ and $r_c = \{r_{c1}, r_{c2}, \dots\}$, and $\Omega = \{\Omega_i\}$, be the finite set that defines possible source encoding rates, possible channel encoding rates and all possible working combination of source code rate and channel code rate as operating mode, respectively. Since the total number of bits are fixed (W), the source and channel rates only are one independent variable, since they are relates as,

$$W \geq N r_s / r_c, \tag{2.28}$$

where N is the number of source sample at the input of the source encoder during one encoding period. The inequality (2.28), provides us with numerous choices for the value of r_s given a fixed r_c, but only the largest possible value is reasonable, because given a fixed SNR, the value of $P_r(\gamma)$ is also fixed, so the largest possible r_s will minimize the $D_S(r_s)$ and end-to-end distortion. Therefore, the choice of r_s given r_c is the one where (2.28) is closest to an equality. Consequently, the operating mode can be identified through only the choice of r_c or the choice of r_s. In practice since we are working with discrete numbers, this condition may not exactly meet, and some padding bits may be added. If we define the operating mode by the source coding rate we have, $\Omega = \{\Omega_i\} = \{r_{si}\}$, then the equation (2.24), will change notation accordingly to the following,

$$d_{\Omega_i} = D_F P_{\Omega_i}(\gamma) + D_S(\Omega_i)\Big(1 - P_{\Omega_i}(\gamma)\Big), \tag{2.29}$$

where $P_{\Omega_i}(\gamma)$ is the frame error probability for the given operating mode (Ω_i), $D_S(\Omega_i)$ is the source codec D-R function, and D_F is the distortion when an error happened during transmission of the data frame and error concealment is needed at the source decoder. The

end-to-end formula (2.29) is called the single mode D-SNR curve, since for each mode we have a different D-SNR curve.

The goal of concatenated joint source-channel coding design is to find the operating mode that minimizes the end-to-end distortion. Mathematically, this goal can be defined as,

$$\min_{\Omega} \left\{ D_F P_{\Omega_i}(\gamma) + D_S(\Omega_i)\Big(1 - P_{\Omega_i}(\gamma)\Big) \right\},$$
$$\text{such that} \quad W \geq N r_s / r_c.$$

The optimization problem for a range of channel SNRs forms an overall minimum distortion function that is called the *"D-SNR curve"*. This optimization problem is discussed in more detail in chapter 4.

2.5 Delay-Constrained JSCC using Incremental Redundancy with Feedback

Studying delay-constrained joint source-channel coding is motivated by the ever increasing demand over sending real-time multimedia sources with constrained delay. As it is described in [10], delay-constrained video coding applications requires a total delay, which consists of buffer delay and transmission delay, to be smaller than the interval between two frames. This results in using Forward Error Correction (FEC) as the primary method for error control.

Figure 2.11 illustrates the block diagram of the communication system to implement delay-constrained JSCC using incremental redundancy. The source encoder works on a fixed block of source samples and can operate at various rates. For each particular rate, a

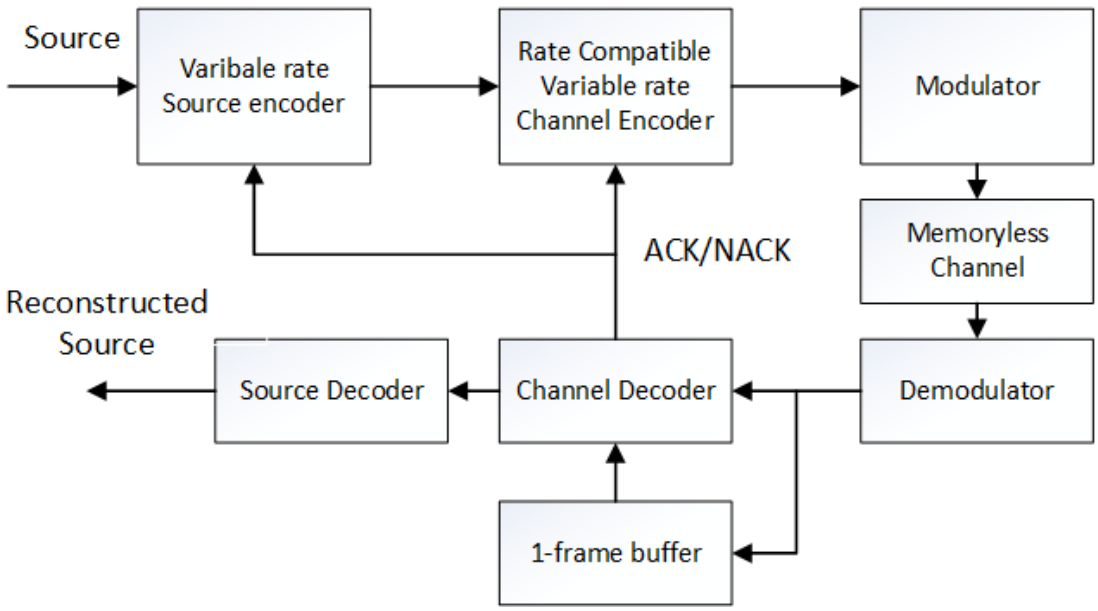

Figure 2.11: Block diagram of the communication system to implement delay-constrained JSCC using incremental redundancy

fixed number of bits are generated to encode the source samples. These bits form source blocks. In the next step codewords are generated from source blocks using a concatenation of an outer error detection code and an inner error correction code selected from a rate compatible family of codes.

Since the transmission is synchronous over a wireless channel, it can be assumed fixed-size frames per source block sample. Incremental redundancy is used with feedback based transmission mechanism. The transmission system uses ACK/NACK feedback to determine what codewords should be included in a frame. A frame may contain codewords and partial codewords for one or two source blocks depending on the feedback.

To see the operation of the system in Figure 2.11, let's consider three frames in three consecutive time-frames regarded as previous, current and next frames with corresponding source blocks following the same sequence in time. If the feedback received at sender was ACK for the previous frame, then the current frame will entirely consist of bits related

to codewords of the current source block. At the receiver, the current frame will be decoded to retrieve the current source block. The channel coding operation determines the success or failure of this transmission and will send one bit ACK/NACK through a feedback channel to notify the sender. If the sender receives an ACK for the current frame, it would continue sending the next frame with the same operation as for the current frame. If a NACK is received, which indicates the failure during transmission of a current frame, the channel encoder would choose lower rate channel code for the current source block (can be achieved through rate compatibility of channel codes by adding extra parity bits). These extra parity bits, referred to as *Incremental Redundancy bits*, are sent as a part of the next frame combined with the codeword for the next source block. Because the frame size is fixed, the source and channel coding rate in the next frame will be chosen so as to accommodate the incremental redundancy for the current source. On the receiver side, the decoder will decode the next codeword and also will try to decode the current codeword to the best of its capability by combining the previously received source block and the received incremental redundancy. Note that upon receiving the incremental redundancy the currently decoded source block is accepted without further attempts at using incremental redundancy. Thus, the maximum delay, in this case, will be one frame to decode a source block which is in compliance with the condition mentioned in [10]. Note that, in this mechanism the essential property is to reduce the source encoding rate in the next frame, to allow for the transmission of incremental redundancy bits to help error recovery of the current source. As a result, the decoded next source block may suffer higher average distortion than the current source. However, it will help to decode the current source and reduce its channel-induced distortion.

More detail description of the mathematical model of incremental redundancy scheme

is discussed in chapter 4.

2.6 Summary

In this chapter, we have summarized the elements of predictive coding system and focused on DPCM as an instance of such a system. Then we described the source fundamental properties. Also, we considered the performance of DPCM through end-to-end distortion that is the combination of source encoding distortion and channel-induced distortion. To address the channel-induced distortion we studied the rate-compatible channel codes. Then, to optimize the overall end-to-end distortion we used RCPC codes in conjunction with joint-source channel coding. Finally, we highlight the importance of the incremental redundancy when designing a system based on the delay-constrained joint source-channel coding. In the next chapter, we will focus on DPCM system distortion characteristics.

Chapter 3

Distortion Analysis

3.1 Problem Setup

Because multimedia sources usually exhibit some form of correlation between samples, the source codecs are frequently built around a DPCM-type codec, where the correlation of source samples is exploited through the use of predictive coding. However, it has become a broadly spread notion that transmission errors significantly degrade the performance of DPCM-type codecs (e.g. [3]). Yet, in an often overlooked comment in [11], this notion is noted as a misconception. Nevertheless, this misconception persisted after [11], maybe because this is only a passing comment within a section that is focused on an analysis that does not provide the full insights into the mechanism at play. In fact, the comment in [11] made no attempt at presenting an unequivocal case to dispel the misconception by not going any further than a single comment and not conducting any study about when transmission errors significantly degrade the performance of DPCM-type codecs and when this is not the case. In this chapter we present an analysis that leads to a closed-form expressions to characterize the main elements that determine the end-to-end distortion of multimedia codecs, allowing for an in-depth study of the effects of transmission errors. To achieve this, we extract and model the core operational elements in multimedia source codecs by focusing on a simpler DPCM-based source coding scheme. The insight from the presented analysis will show that both DPCM-type codecs and memoryless codecs

experience the same distortion due to transmission errors, even when DPCM-type codecs suffer from error propagation while memoryless codecs do not. Furthermore, the analysis will show that the reasons for this result are the DPCM coding gain and the better error concealment performance that are achieved in a DPCM system by exploiting the source memory and by using the optimal predictor in the DPCM coding loop.

The rest of this chapter is organized as follows, section 3.2, describes the problem and the setup used to study the problem. Section 3.3 derives the equations related to source encoding distortion. Subsection 3.4 leads to formulate the end-to-end distortion followed by discussion and numerical results.

3.2 System Setup

In order to mimic the main configuration of multimedia codecs, we study the end-to-end distortion of a communication system that transmits a block of M samples using a memoryless source encoder for the first sample and a DPCM (predictive coding) encoder for the rest of samples. This setup models the main elements in the encoding of video, where the fist frame in a block-of-frames is encoded without exploiting the source sample time-correlation (making it an I-frame), and the encoding of the rest of the frames can use the temporal correlation with previous frames (as P frames). The considered DPCM encoder and decoder are shown in Fig. 2.2, where x, $\hat{x}$ and $\tilde{x}$ are the source, predicted and decoded samples respectively. We will assume a synthetic source that is designed to instantiate the most fundamental characteristics of a video-like source. As such, we assume a source that follows a first order autoregressive - AR(1) - process: $x[n] = \rho x[n-1] + z[n]$, where the innovation process $z[n]$ forms a sequence of independent and identically distributed random-valued samples following a Gaussian distribution with zero mean and variance σ_z^2.

Accordingly, A DPCM uses a linear prediction filter with the input-output relation given by,

$$\hat{x}[n] = \sum_{k=1}^{L} a_k \tilde{x}[n-k],$$

For a predictor of order L with coefficients a_k. In the particular case of a first-order autoregressive, AR(1), source, $L = 1$, and we have,

$$\hat{x}[n] = a_1 \tilde{x}[n-1]. \tag{3.1}$$

From ([6]), it is well known that to minimize the mean square prediction error, $a = \rho$.

Note that the source statistical properties show the same behavior as formulated in subsection 2.2.3.

3.2.1 Properties of DPCM Codecs

Note that in a DPCM encoder the signal $e[n]$ is the input to a scalar, memoryless quantizer, while $\tilde{e}[n]$ is the corresponding output. In the case of scalar, memoryless quantizers it was shown in [7] that the input and output from the scalar, memoryless quantizer exhibits the property that,

$$E\big[e[n]\tilde{e}[n]\big] = E\big[\tilde{e}[n]^2\big]. \tag{3.2}$$

Also, since for a good predictive codec the prediction errors at different times are uncorrelated, quantized or not, $E\big[(e[n]\tilde{e}[n-k]\big] = 0$ and $E\big[\tilde{e}[n]\tilde{e}[n-k]\big]$ and, thus,

$$E[(e[n] - \tilde{e}[n])\tilde{e}[n-k]] = 0, \quad \forall\, k > 0 \tag{3.3}$$

3.3 Source Encoding Distortion

In the designed system to transmit a block of M samples, we assume the first sample is encoded using memoryless source encoder and the rest of samples are DPCM encoded. Therefore, to understand the distortion due to source encoding, we need to differentiate between the memoryless encoded source and the DPCM encoded source. Let D_S denote the source encoding distortion for ideal, memory-less encoding. For a source encoding rate of R bits per source samples, from [12], we have,

$$D_S(R) = \sigma_x^2 2^{-2R} = \frac{\sigma_z^2 2^{-2R}}{1-\rho^2},$$

when in the second equation we have used (2.11).

For scalar, memoryless quantization under the high rate approximation, the source encoding distortion is $D_q \approx \frac{\Delta^2}{12}$,[6], where Δ is the average width of the quantization interval. This approximation is valid for uniform and non-uniform quantizers with source sample with symmetric probability density function (pdf). Assuming the dynamic range of the quantizer to be in the range $x \in [-4\sigma_x, 4\sigma_x]$, there are k-2 decision intervals in this range where $k = 2^R$. Then, we have,

$$\Delta = \frac{8\sigma_x}{k-2} \approx \frac{8\sigma_x}{k}$$

$$D_q = \frac{16}{3}\frac{\sigma_x^2}{k^2} = \frac{16}{3}\sigma_x^2 2^{-2R} = \frac{16}{3}D_S. \tag{3.4}$$

Next, let $D_m[n]$ be the source encoding distortion at time n when using DPCM encoding. Using the property of DPCM codec in (2.1), the source encoding distortion can be expressed as,

$$D_m[n] = E[(x[n] - \tilde{x}[n])^2] = E[(e[n] - \tilde{e}[n])^2]. \tag{3.5}$$

Making the approximation that the predictor coefficients are formed from actual past values and the prediction from reconstructed and actual values are about the same, the source encoder predictor error (σ_e^2) yields,

$$\begin{aligned}
\sigma_e^2 &= E\big[(x[n] - \hat{x}[n])^2\big] \\
&= E\big[(x[n] - \hat{x}[n])(x[n] - \hat{x}[n])\big] \\
&= E\big[\big(x[n] - \hat{x}[n]\big)x[n]\big] - E\big[\big(x[n] - \hat{x}[n]\big)\hat{x}[n]\big] \\
&\overset{(\dagger)}{=} E\big[\big(x[n] - \hat{x}[n]\big)x[n]\big] \\
&\overset{(\ddagger)}{=} \sigma_x^2 - \sum_{k=1}^{L} a_k E\big[x[n]x[n-k]\big] \\
&= \sigma_x^2 - \sum_{k=1}^{L} a_k \phi(k),
\end{aligned} \tag{3.6}$$

where equality $(\ddagger)$ follows because the predictor can be approximated as

$$\sum_{k=1}^{L} a_k \tilde{x}[n-k] \approx \sum_{k=1}^{L} a_k x[n-k]. \tag{3.7}$$

This clearly shows that in DPCM prediction is done based on reconstructed past values. Also, equality $(\dagger)$ follows because,

$$
\begin{aligned}
E\big[\big(x[n]-\hat{x}[n]\big)\hat{x}[n]\big] &= E\Big[\big(x[n]-\hat{x}[n]\big)\Big(\sum_{k=1}^{L} a_k\tilde{x}[n-k]\Big)\Big] \\
&\overset{\text{from (3.7)}}{\approx} E\Big[\big(x[n]-\hat{x}[n]\big)\Big(\sum_{k=1}^{L} a_k x[n-k]\Big)\Big] \\
&= \sum_{k=1}^{L} a_k E\big[\big(x[n]-\hat{x}[n]\big)x[n-k]\big] \\
&\overset{(\spadesuit)}{=} 0.
\end{aligned}
$$

Here, equality $(\spadesuit)$ uses the orthogonality principle for the MMSE predictor as showed in (3.3) (input samples are orthogonal to filter prediction error). Note that when $L = 1$ (AR(1) process), from (3.6),

$$
\begin{aligned}
\sigma_e^2 &= \sigma_x^2 - a\phi(1) \\
&= \sigma_x^2 - a\rho\sigma_x^2 \\
&= \sigma_x^2(1-a\rho). \qquad (3.8)
\end{aligned}
$$

From inspecting figure 2.2, it can be seen that the input to the scalar, memoryless quantizer is the signal $e[n]$. At the same time, (3.5) indicates that the distortion of the DPCM codec equals the quantization mean squared error experienced by the signal $e[n]$. Assuming high rate approximation and that the pdf of $e[n]$ is symmetric, D_m is the distortion for the signal $e[n]$ when it is being quantized by a DPCM encoder. Using (3.4),

$$
D_m = \frac{16}{3}\sigma_e^2 2^{-2R}. \qquad (3.9)
$$

Then, for an AR(1) source, the source encoding distortion (D_m), from (3.9) and (3.8) can be expressed as,

$$\begin{aligned} D_m &= \frac{16}{3}(1-a\rho)\sigma_x^2 2^{-2R} \\ &= (1-a\rho)D_q. \end{aligned} \tag{3.10}$$

The coefficient $1-a\rho$ is the source coding gain of DPCM source coding over the memoryless source coding. Not that when the source shows high correlation this gain increases.

3.4 End-to-End Distortion

Calculation of the end-to-end distortion involves the computation of the source encoder distortion (discussed above) and the channel-induced distortion. One factor that affects the magnitude of the channel-induced distortion is the error concealment operation that is being applied at the source decoder. In this section, we assume that the optimal error concealment is being applied. Also, in the next section, we will study the effect of having suboptimal error concealment. In this work, we assume that an error in the transmission of the first sample encoded using a scalar memoryless quantizer is concealed by replacing it with its expected value. For the $M-1$ subsequent samples, which are DPCM-encoded, a transmission error is concealed by estimating the lost sample using the reconstructed value of the previous sample (*i.e.* replacing the corrupted value by its predicted value). Since, the studied system uses both memoryless and DPCM coding, the analysis of the channel-induced distortion requires considering different cases. We first consider cases results in unsuccessful transmission of the first sample. As indicated, when the sample received at time 0 has uncorrectable errors (which will be denoted as $\tilde{x}_0[0]$), it is concealed as

$$\tilde{x}_0[0] = E[x[0]] = 0. \tag{3.11}$$

Let the average end-to-end distortion at time 0, when there is an error in the transmission of the sample at time 0 be denoted as $D_{ee,0}[0]$. This distortion is,

$$\begin{aligned} D_{ee,0}[0] &= E[(x[0]-\tilde{x}_0[0])^2] \\ &= E[(x[0]-\tilde{x}[0]+\tilde{x}[0]-\tilde{x}_0[0])^2] \\ &= E[(x[0]-\tilde{x}[0])^2]+E[(\tilde{x}[0]-\tilde{x}_0[0])^2] \\ &\quad +2E[(x[0]-\tilde{x}[0])(\tilde{x}[0]-\tilde{x}_0[0])] \\ &\overset{(✠)}{=} D_q + E[\tilde{x}[0]^2], \end{aligned} \tag{3.12}$$

where (✠) follows from the following reasons:

- $2E[(x[0]-\tilde{x}[0])(\tilde{x}[0]-\tilde{x}_0[0])] = 0$ as was shown in [7] under the assumption of optimal quantization and sample reconstruction.
- By definition $E[(x[0]-\tilde{x}[0])^2] = D_q$.
- Because of error concealment, $\tilde{x}_0[0] = E[x[0]] = 0$, $E[(\tilde{x}[0]-\tilde{x}_0[0])^2] = E[\tilde{x}[0]^2]$.

Next, to derive $E[\tilde{x}[0]^2]$ in (3.12), since $D_q = E[(x[0]-\tilde{x}[0])^2] = E[x[0]^2] + E[\tilde{x}[0]^2] - 2E[x[0]\tilde{x}[0]]$, we have that

$$E[\tilde{x}[0]^2] = D_q - E[x[0]^2] + 2E[x[0]\tilde{x}[0]]. \tag{3.13}$$

From (3.2), now with input and output samples $x[0]$ and $\tilde{x}[0]$, respectively, it follows that $E[x[0]\tilde{x}[0]] = E[\tilde{x}[0]^2]$ which when replaced in (3.13) results in,

$$
\begin{aligned}
E[\tilde{x}[0]^2] &= D_q - \sigma_x^2 + 2E[\tilde{x}[0]^2], \\
\Rightarrow E[\tilde{x}[0]^2] &= \sigma_x^2 - D_q \triangleq D_0[0]. \qquad (3.14)
\end{aligned}
$$

Combining (3.12) and (3.14), it yields $D_{ee,0}[0] = D_q + D_0[0]$, which provides meaning to $D_0[0]$ as the distortion at time $n = 0$ that is added to the source encoding distortion due to a transmission error at time $n = 0$. Also, this result implies that

$$D_{ee,0}[0] = D_q + \sigma_x^2 - D_q = \sigma_x^2. \qquad (3.15)$$

Next, we look into different scenarios for DPCM-AR(1) coding. To obtain a similar derivation for the end-to-end distortion denotes as $D_{ee}[1]$, we should recognize that transmission error can happen at time 0, at time 1 or both. First we consider the situation, when a transmission error happens at time 0. In this case the output of the DPCM decoder at time $n = 1$ is (with error concealment as in (3.11)),

$$\tilde{x}_0[1] = a\tilde{x}_0[0] + \tilde{e}[1] = \tilde{e}[1]. \qquad (3.16)$$

The average end-to-end distortion in this case is,

$$\begin{aligned} D_{ee,0}[1] &= E[(x[1]-\tilde{x}_0[1])^2] \\ &= E[(x[1]-\tilde{x}[1])^2] + E[(\tilde{x}[1]-\tilde{x}_0[1])^2] \\ &\quad -2E[(x[1]-\tilde{x}[1])(\tilde{x}[1]-\tilde{x}_0[1])] \\ &\overset{(\flat)}{=} D_m[1] + E[(a\tilde{x}[0]+\tilde{e}[1]-\tilde{e}[1])^2] \\ &= D_m[1] + E[(a\tilde{x}[0])^2] \\ &\overset{\text{from (3.14)}}{=} D_m[1] + a^2(\sigma_x^2 - D_q) \\ &= D_m[1] + a^2 D_0[0], \end{aligned} \tag{3.17}$$

where $(\flat)$ follows follows from the following reasons:

- By definition, $E[(x[1]-\tilde{x}[1])^2] = D_m[1]$.
- Because of (3.16) and $\tilde{x}[1] = a\tilde{x}[0] + \tilde{e}[1]$ (recall that $\tilde{x}[1]$ is the output with no error and no error propagation), $E[(\tilde{x}[1]-\tilde{x}_0[1])^2] = E[(a\tilde{x}[0]+\tilde{e}[1]-\tilde{e}[1])^2]$,
- The fact that,

$$\begin{aligned} E[(x[1]-\tilde{x}[1])(\tilde{x}[1]-\tilde{x}_0[1])] &= E[(x[1]-\tilde{x}[1])\tilde{e}[1]] \\ &\overset{\text{from (2.1)}}{=} E[(e[1]-\tilde{e}[1])\tilde{e}[1]] \\ &= E[e[1]\tilde{e}[1]] - E[\tilde{e}[1]^2] \\ &\overset{\text{from (3.2)}}{=} 0. \end{aligned}$$

Note how (3.17) shows that the error introduced at time 0, through the error concealment, propagates to time 1 scaled by the factor a^2. By applying recursion and using the same

procedures, at an arbitrary time k,

$$D_{ee,o}[k] = D_m[k] + a^{2k} D_0[0] = D_m[k] + a^{2k}(\sigma_x^2 - D_q). \tag{3.18}$$

We now consider the case of a transmission error at time $n = 1$ only (when transmitting $\tilde{e}[1]$). In this case, error concealment takes the form $\tilde{x}_1[1] = a\tilde{x}[0]$ and the average end-to-end distortion at time $n = 1$ is,

$$\begin{aligned}
D_{ee,1}[1] &= E[(x[1] - \tilde{x}_1[1])^2] \\
&= E[(x[1] - \tilde{x}[1])^2] + E[(\tilde{x}[1] - \tilde{x}_1[1])^2] - 2E[(x[1] - \tilde{x}[1])(\tilde{x}[1] - \tilde{x}_1[1])] \\
&= D_m[1] + E[(\tilde{x}[1] - \tilde{x}_1[1])^2] - 2E[(x[1] - \tilde{x}[1])(\tilde{x}[1] - \tilde{x}_1[1])] \\
&\overset{(\natural)}{=} D_m[1] + E[(a\tilde{x}[0] + \tilde{e}[1] - a\tilde{x}[0])^2] \\
&= D_m[1] + E[\tilde{e}[1]^2] \\
&= E[(x[1] - \tilde{x}[1])^2] + E[\tilde{e}[1]^2] \\
&= E[(e[1] - \tilde{e}[1])^2] + E[\tilde{e}[1]^2] \\
&= E[e[1]^2] - 2E[e[1]\tilde{e}[1]] + E[\tilde{e}[1]^2] + E[\tilde{e}[1]^2] \\
&\overset{\text{from (3.2)}}{=} E[e[1]^2] + 2E[\tilde{e}[1]^2] - 2E[\tilde{e}[1]^2] \\
&= E[e[1]^2] \\
&= \sigma_e^2 \\
&\overset{\text{from (3.8)}}{=} (1 - a\rho)\sigma_x^2,
\end{aligned}$$

where equality ($\natural$) follows from $\tilde{x}[1] = a\tilde{x}[0]+\tilde{e}[1]$, $\tilde{x}_1[1] = a\tilde{x}[0]$ and $E[(x[1]-\tilde{x}[1])(\tilde{x}[1]-\tilde{x}_1[1])] = 0$ due to

$$\begin{aligned}
E[(x[1]-\tilde{x}[1])(\tilde{x}[1]-\tilde{x}_1[1])] &= E[(x[1]-\tilde{x}[1])(\tilde{x}[1]-a\tilde{x}[0])] \\
&= E[(x[1]-\tilde{x}[1])\tilde{x}[1]] - aE[(x[1]-\tilde{x}[1])\tilde{x}[0]] \\
&= E[(x[1]-\tilde{x}[1])(a\tilde{x}[0]+\tilde{e}[1])] - aE[(x[1]-\tilde{x}[1])\tilde{x}[0]] \\
&= aE[(x[1]-\tilde{x}[1])\tilde{x}[0]] - aE[(x[1]-\tilde{x}[1])\tilde{x}[0]] \\
&\quad +E[(x[1]-\tilde{x}[1])\tilde{e}[1]] \\
&= E[(x[1]-\tilde{x}[1])\tilde{e}[1]] \\
&\overset{\text{from (2.1)}}{=} E[(e[1]-\tilde{e}[1])\tilde{e}[1]] \\
&= E[e[1]\tilde{e}[1]] - E[\tilde{e}[1]^2] \\
&\overset{\text{from (3.2)}}{=} 0.
\end{aligned}$$

The average distortion at time $n = 1$ due to a transmission error at time $n = 1$ can be written as,

$$D_1[1] = E[\tilde{e}[1]^2] = (1-a\rho)\sigma_x^2 - D_m[1],$$

which follows from $D_{ee,1}[1] = D_m[1] + D_1[1]$.

We can now calculate the the average end-to-end distortion at a general time j when a transmission error occurs at the same time $n = j$, $D_j[j]$. In this case the error concealment is done similar to other DPCM coded samples as $\tilde{x}_j[j] = a\tilde{x}[j-1]$. Therefore, for the

average end-to-end distortion, we get,

$$\begin{aligned} D_j[j] &= E[(x[j] - \tilde{x}_j[j])^2] \qquad (3.19) \\ &= E[(x[j] - a\tilde{x}[j-1])^2] \\ &= E[(a\tilde{x}[j-1] + \tilde{e}[j] - a\tilde{x}[j-1])^2] \\ &= E[\tilde{e}[j]^2] \\ &\stackrel{\Diamond}{=} \sigma_e^2 - D_m[j] \\ &= (1 - a\rho)\sigma_x^2 - D_m[j], \qquad (3.20) \end{aligned}$$

where equality $\Diamond$ follows because,

$$\begin{aligned} D_m[j] &= E[(x[j] - \tilde{x}[j])^2] \\ &\stackrel{\text{from (2.1)}}{=} E[(e[j] - \tilde{e}[j])^2] \\ &= E[e[j]^2] - 2E[e[j]\tilde{e}[j]] + E[\tilde{e}[j]^2] \\ &= E[\tilde{e}[j]^2] \\ &\stackrel{\text{from (3.2)}}{=} \sigma_e^2 - 2E[\tilde{e}[j]^2] + E[\tilde{e}[j]^2] \\ &= \sigma_e^2 - E[\tilde{e}[j]^2] \\ \Rightarrow E[\tilde{e}[j]^2] &= \sigma_e^2 - D_m[j]. \end{aligned}$$

Using the superposition principle, the average end-to-end distortion at time $n = 1$ when there are transmission errors both at times $n = 0$ and $n = 1$ is,

$$D_{ee,01}[1] = E[(x[1] - \tilde{x}_{01}[1])^2] \tag{3.21}$$

$$= a^2(\sigma_x^2 - D_q) + \sigma_x^2(1 - a\rho) \tag{3.22}$$

Considering now a last specific scenario for a single transmission error, the end-to-end distortion at time $n = 2$ when the error happens at time 1 follows from considering that the reconstruction will use the previous error-concealed sample $\tilde{x}_1[1]$ in the reconstruction of $\tilde{x}_1[2]$, $\tilde{x}_1[2] = a\tilde{x}_1[1] + \tilde{e}[2]$. Therefore,

$$\begin{aligned}
D_{ee,1}[2] &= E[(x[2] - \tilde{x}_1[2])^2] \\
&= E[(x[2] - \tilde{x}[2])^2] \\
&\quad + E[(\tilde{x}[2] - \tilde{x}_1[2])^2] - 2E[(x[2] - \tilde{x}[2])(\tilde{x}[2] - \tilde{x}_1[2])] \\
&= D_m[2] + E[(\tilde{x}[2] - \tilde{x}_1[2])^2] \\
&\quad - 2E[(x[2] - \tilde{x}[2])(\tilde{x}[2] - \tilde{x}_1[2])] \\
&\overset{(\bigstar)}{=} D_m[2] + E[(a\tilde{x}[1] + \tilde{e}[2] - a\tilde{x}_1[1] - \tilde{e}[2])^2] \\
&= D_m[2] + a^2 E[(\tilde{x}[1] - \tilde{x}_1[1])^2] \\
&\overset{\text{from (3.19)}}{=} D_m[2] + a^2 D_1[1] \\
&\overset{\text{from (3.20)}}{=} D_m[2] - a^2 D_m[1] + a^2(1 - a\rho)\sigma_x^2,
\end{aligned}$$

Where ($\bigstar$) follows because, $\tilde{x}[2] = a\tilde{x}[1] + \tilde{e}[2]$, $\tilde{x}_1[2] = a\tilde{x}_1[1] + \tilde{e}[2]$ and $E[(x[2] - \tilde{x}[2])(\tilde{x}[2] - \tilde{x}_1[2])] = 0$ since,

$$\begin{aligned}
E[(x[2] - \tilde{x}[2])(\tilde{x}[2] - \tilde{x}_1[2])] &= E[(x[2] - \tilde{x}[2])(a\tilde{x}[1] + \tilde{e}[2] - a\tilde{x}_1[1] - \tilde{e}[2])] \\
&\overset{\text{from (2.1)}}{=} E[(e[2] - \tilde{e}[2])(a\tilde{x}[1] - a\tilde{x}_1[1])] \\
&= E[(e[2] - \tilde{e}[2])a(a\tilde{x}[0] + \tilde{e}[1] - a\tilde{x}[0])] \\
&= E[(e[2] - \tilde{e}[2])a(\tilde{e}[1])] \\
&= aE[(e[2] - \tilde{e}[2])\tilde{e}[1]] \\
&\overset{\text{from (3.3)}}{=} 0
\end{aligned}$$

To study the general end-to-end distortion case, we denote $D_{ee,j}[k]$ as the overall average distortion at time $k > 0$ when an error happened at time j ($0 < j \leq k$) and with the error concealed as,

$$\tilde{x}_j[j+1] = a\tilde{x}[j] \tag{3.23}$$

The end-to-end distortion is obtained as,

$$
\begin{aligned}
D_{ee,j}[k] &= E[(x[k]-\tilde{x}_j[k])^2] \\
&= E[(x[k]-\tilde{x}_j[k])^2] + E\big[(\tilde{x}[k]-\tilde{x}_j[k])^2\big] - 2E[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] \\
&= D_m[k] + E\big[(\tilde{x}[k]-\tilde{x}_j[k])^2\big] - 2E[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] \\
&\overset{(\blacklozenge)}{=} D_m[k] + a^{2(k-j)}E[\tilde{e}[j]^2] - 2E\big[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] \\
&\overset{(\square)}{=} D_m[k] + a^{2(k-j)}E[\tilde{e}[j]^2] \\
&= D_m[k] + a^{2(k-j)}\big[(1-a\rho)\sigma_x^2 - D_m[j]\big], \qquad (3.24)
\end{aligned}
$$

where $(\blacklozenge)$ follows from applying recursion on $\tilde{x}[k]-\tilde{x}_j[k]$,

$$
\begin{aligned}
\tilde{x}[k]-\tilde{x}_j[k] &= a(\tilde{x}[k-1]-\tilde{x}_j[k-1]) \\
&= a^2(\tilde{x}[k-2]-\tilde{x}_j[k-2]),
\end{aligned}
$$

and continuing the recursion, untill we eventually obtain,

$$
\begin{aligned}
\tilde{x}[k]-\tilde{x}_j[k] &= a^{k-j}(\tilde{x}[k-(k-j)]-\tilde{x}_j[k-(k-j)]) \\
&= a^{k-j}(\tilde{x}[j]-\tilde{x}_j[j]) \\
&= a^{k-j}(a\tilde{x}[j-1]+\tilde{e}[j]-a\tilde{x}[j-1]) \\
&= a^{k-j}\tilde{e}[j] \qquad (3.25)
\end{aligned}
$$

Also, in the derivation of $D_{ee,j}[k]$, equality $(\square)$ is due to $E\big[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] = 0$, because, from (2.1), (3.3) and (3.25), $E\big[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] = E\big[(e[k]-\tilde{e}[k])a^{k-j}\tilde{e}[j]\big] = 0$.

We now proceed to calculate the overall average end-to-end distortion at a time n, that combines all possible events (errors may or may not occur), $D_{ee}[n]$. Let P_n denote the probability of a transmission error at time n. At time $n = 0$ the distortion will equal D_q when there is no transmission error (with probability $1 - P_0$) and $D_{ee,0}[0]$ when there is a transmission error (with probability P_0). Using (3.12) and (3.14) to find D_{ee} at time 0, we get,

$$\begin{aligned} D_{ee}[0] &= D_q(1 - P_0) + D_{ee,0}[0]P_0 \\ &= D_q(1 - P_0) + \sigma_x^2 P_0 \\ &= D_q + P_0(\sigma_x^2 - D_q). \end{aligned} \tag{3.26}$$

Note that this expression shows how we always (with probability one) have a distortion D_q (source encoding distortion) and to this distortion we add a distortion $D_0[0]$ with probability P_0 to account for possible transmission errors.

The overall average end-to-end distortion at time $n = 1$ is calculated by combining the distortion from four possible events: a transmission error happened only at time $n = 0$ (an event with probability $P_0(1 - P_1)$), a transmission error happened only at time $n = 1$ (an event with probability $(1 - P_0)P_1$), a transmission error happened at times $n = 0$ and $n = 1$ (an event with probability P_0P_1), and no transmission error happened (an event with

probability $(1-P_0)(1-P_1)$). Therefore, the overall end-to-end distortion is,

$$\begin{aligned} D_{ee}[1] &= D_m[1](1-P_0)(1-P_1) + D_{ee,0}[1]P_0(1-P_1) \\ &\quad + D_{ee,1}[1](1-P_0)P_1 + D_{ee,01}[1]P_0P_1 \\ &= D_m[1](1-P_0-P_1+P_0P_1) \\ &\quad + D_{ee,0}[1](P_0-P_0P_1) + D_{ee,1}[1](P_1-P_0P_1) + D_{ee,01}[1]P_0P_1. \end{aligned} \tag{3.27}$$

When there are multiple transmission errors the average end-to-end distortion is calculated by applying the superposition principle. Therefore, it is possible to observe the relation $D_{ee,jk}[n] = D_{ee,j}[n] + D_{ee,k}[n] - D_m[n]$, where the source encoding distortion $D_m[n]$ is subtracted to compensate for being accounted for both in $D_{ee,j}[n]$ and in $D_{ee,k}[n]$. Using this relation, replacing $D_{ee,01}[1] = D_{ee,0}[1] + D_{ee,1}[1] - D_m[1]$ in (3.27) and simplifying the resulting expression, yields

$$D_{ee}[1] = D_m[1](1-P_0-P_1) + D_{ee,0}[1]P_0 + D_{ee,1}[1]P_1 \tag{3.28}$$

$$\stackrel{(\blacksquare)}{=} D_m[1](1-P_1) + a^2(\sigma_x^2 - D_q)P_0 \tag{3.29}$$

$$+ (1-a\rho)\sigma_x^2 P_1, \tag{3.30}$$

where equality ($\blacksquare$) follows from replacing $D_{ee,0}[n] = D_m[n] + a^{2n}(\sigma_x^2 - D_q)$ (as per (3.18)) and $D_{ee,k}[n] = D_m[n] + a^{2(n-k)}\big[(1-a\rho)\sigma_x^2) - D_m[k]\big]$ (as per (3.24)) followed by simplifications and collecting common terms. Both expressions (3.28) and (3.29) are insightful in their own different way to characterize how the overall average end-to-end distortion at time $n = 1$ is constructed. Expression (3.28) shows that end-to-end distortion at time $n = 1$ ($D_{ee}[1]$) is closely related to the value of end-to-end distortion propagated

from time $n = 0$ and time $n = 1$. Also, the equation (3.29) shows the value of the end-to-end distortion from time $n = 0$ decreased at time $n = 1$ with the coefficient of a^2.

The derivation of the end-to-end distortion at time $n = 2$ follows the same procedure as for the time $n = 1$ with the only addition that now it will be necessary to consider the end-to-end distortion in the case when there was three transmission errors (at times $n = 0$, 1 and 2). For this case, it will be necessary to use the relation $D_{ee,jkl}[n] = D_{ee,j}[n] + D_{ee,k}[n] + D_{ee,l}[n] - 2D_m[n]$, derived from the application of the superposition principle. Note how now source encoding distortion $D_m[n]$ is subtracted twice to compensate for being accounted for in all $D_{ee,j}[n]$, $D_{ee,k}[n]$ and $D_{ee,l}[n]$ distortions. Consequently,

$$
\begin{aligned}
D_{ee}[2] &= D_m[2](1 - P_0 - P_1 + P_0P_1)(1 - P_2) + D_{ee,0}[2](1 - P_1 - P_2 + P_1P_2) \\
&\quad + D_{ee,1}[2]P_1(1 - P_0 - P_2 + P_0P_2) + D_{ee,2}[2]P_2(1 - P_0 - P_1 + P_0P_1) \\
&\quad + D_{ee,01}[2](P_0P_1 - P_0P_1P_2) + D_{ee,02}[2](P_0P_2 - P_0P_1P_2) \\
&\quad + D_{ee,12}[2](P_1P_2 - P_0P_1P_2) + D_{ee,012}[2]P_0P_1P_2 \\
&= D_m[2](1 - P_0 - P_1 - P_2) + D_{ee,0}[2]P_0 + D_{ee,1}[2]P_1 + D_{ee,2}[2]P_2 \\
&= D_m[2](1 - P_0 - P_1 - P_2) + D_m[2]P_0 + a^4(\sigma_x^2 - D_q)P_0 \\
&\quad + (D_m[2] + a^2[(1 - a\rho)\sigma_x^2 - D_m[1]])P_1 + (1 - a\rho)\sigma_x^2P_2 \\
&= D_m[2](1 - P_2) + a^4(\sigma_x^2 - D_q)P_0 + a^2[(1 - a\rho)\sigma_x^2 \\
&\quad - D_m[1]]P_1 + (1 - a\rho)\sigma_x^2P_2.
\end{aligned}
$$

To summarize the results for overall average end-to-end distortion at times $n = 0$, $n = 1$ and $n = 2$, the results that provide insight into how the distortion is composed are,

$$D_{ee}[0] = D_q(1 - P_0) + D_{ee,0}[0]P_0 \tag{3.31}$$

$$= D_q(1 - P_0) + (\sigma_x^2)P_0, \tag{3.32}$$

$$D_{ee}[1] = D_m[1](1 - P_0 - P_1) + D_{ee,0}[1]P_0 + D_{ee,1}[1](P_1) \tag{3.33}$$

$$= D_m[1](1 - P_1) + a^2(\sigma_x^2 - D_q)P_0 + (1 - a\rho)\sigma_x^2 P_1, \tag{3.34}$$

$$D_{ee}[2] = D_m[2](1 - P_0 - P_1 - P_2) + D_{ee,0}[2]P_0 \tag{3.35}$$

$$+D_{ee,1}[2]P_1 + D_{ee,2}[2]P_2$$

$$= D_m[2](1 - P_2) + a^4(\sigma_x^2 - D_q)P_0 \tag{3.36}$$

$$+a^2[(1 - a\rho)\sigma_x^2 - D_m[1]]P_1 + (1 - a\rho)\sigma_x^2 P_2.$$

By following the derivations for $D_{ee}[0]$, $D_{ee}[1]$ and $D_{ee}[2]$, along with inspecting the results (3.31), (3.33) and (3.35), we can obtain from recursion a general formula for the end-to-end distortion at time n,

$$\boxed{\begin{array}{c} D_{ee}[n] = D_m[n](1 - \sum_{k=0}^{n} P_k) + \sum_{j=0}^{n} D_{ee,j}[n]P_n; \\ \\ D_m[0] = D_q; \qquad \text{for } n = 1, .., M - 1. \end{array}} \tag{3.37}$$

Similarly, by using (3.32), (3.34) and (3.36),

$$D_{ee}[0] = D_q(1-P_0) + \sigma_x^2 P_0 = D_q + P_0(\sigma_x^2 - D_q),$$

$$D_{ee}[n] = D_m[n](1-P_n) + a^{2n}(\sigma_x^2 - D_q)P_0 + (1-a\rho)\sigma_x^2 P_n$$
$$+ \sum_{k=1}^{n-1} \Big(a^{2(n-k)} P_k \Big[(1-a\rho)\sigma_x^2 - D_m[k] \Big] \Big),$$
$$\text{for } \; n = 1, .., M-1.$$

Moving the terms $-D_m[n]P_n$ and $(1-a\rho)\sigma_x^2 P_n$ into the sum yields,

$$\boxed{\begin{aligned} &D_{ee}[0] = D_q(1-P_0) + \sigma_x^2 P_0 = D_q + P_0(\sigma_x^2 - D_q), \qquad (3.38)\\ &D_{ee}[n] = D_m[n] + a^{2n}(\sigma_x^2 - D_q)P_0 \qquad (3.39)\\ &\qquad + \sum_{k=1}^{n} \Big(a^{2(n-k)} P_k \Big[(1-a\rho)\sigma_x^2 - D_m[k] \Big] \Big),\\ &\qquad \text{for } n = 1, .., M-1. \end{aligned}}$$

The average end-to-end distortion over all the M samples is,

$$D_{ee} = \frac{1}{M} \sum_{n=0}^{M-1} D_{ee}[n],$$

which when applied to (3.39) yields,

$$\begin{aligned} D_{ee} \;\; = \;\; & \frac{1}{M} \Big[D_q + \sum_{n=1}^{M-1} D_m[n] + P_0(\sigma_x^2 - D_q) \sum_{n=0}^{M-1} (a^{2n}) \\ & + \sum_{n=1}^{M-1} \sum_{k=1}^{n} \Big(a^{2(n-k)} P_k [(1-a\rho)\sigma_x^2 - D_m[k]] \Big) \Big]. \end{aligned}$$

Applying the series result $\sum_{n=0}^{M-1}(a^{2n}) = (1-a^{2M})/(1-a^2)$ results in,

$$\boxed{\begin{aligned} D_{ee} = \frac{1}{M}\Big[D_q + \sum_{n=1}^{M-1} D_m[n] + \frac{P_0(\sigma_x^2 - D_q)(1-a^{2M})}{1-a^2} \\ + \sum_{n=1}^{M-1}\sum_{k=1}^{n}\Big(a^{2(n-k)}P_k[(1-a\rho)\sigma_x^2 - D_m[k]]\Big)\Big]. \end{aligned}} \tag{3.40}$$

We proceed next to specialize the results for a series of special but common cases. First, let's assume $\boxed{a=\rho}$ (ideal predictor filter and error concealment). In this case, from (3.38), (3.39), and (3.40) we get,

$$\boxed{\begin{aligned} &D_{ee}[0] = D_q(1-P_0) + (\sigma_x^2)P_0 = D_q + P_0(\sigma_x^2 - D_q), \\ &D_{ee}[n] = D_m[n] + \rho^{2n}(\sigma_x^2 - D_q)P_0 \\ &\qquad + \sum_{k=1}^{n}\Big(\rho^{2(n-k)}P_k[(1-\rho^2)\sigma_x^2 - D_m[k]\Big]\Big); \quad n = 1,..,M-1, \\ &D_{ee} = \frac{1}{M}\Big[D_q + \sum_{n=1}^{M-1} D_m[n] + \frac{P_0(\sigma_x^2 - D_q)(1-\rho^{2M})}{1-\rho^2} \\ &\qquad + \sum_{n=1}^{M-1}\sum_{k=1}^{n}\Big(\rho^{2(n-k)}P_k[(1-\rho^2)\sigma_x^2 - D_m[k]]\Big)\Big] \end{aligned}}$$

From this result, assuming also that $\boxed{D_m[n] = D_m, n = 1, 2, .., M-1}$ (same DPCM coding rate at all times),

$$\boxed{\begin{aligned}
&D_{ee}[0] = D_q(1-P_0) + (\sigma_x^2)P_0 = D_q + P_0(\sigma_x^2 - D_q),\\
&D_{ee}[n] = D_m + \rho^{2n}(\sigma_x^2 - D_q)P_0\\
&\qquad + [(1-\rho^2)\sigma_x^2 - D_m]\sum_{k=1}^{n}\Big(\rho^{2(n-k)}P_k\Big),\\
&D_{ee} = \frac{1}{M}\Big[D_q + (M-1)D_m + \frac{P_0(\sigma_x^2 - D_q)(1-\rho^{2M})}{1-\rho^2}\\
&\qquad + \sum_{n=1}^{M-1}\sum_{k=1}^{n}\Big(\rho^{2(n-k)}P_k[(1-\rho^2)\sigma_x^2 - D_m]\Big)\Big].
\end{aligned}}$$

From this result, assuming also that $\boxed{P_n = P_f,\ n = 0, 1, .., M-1}$ (same probability of transmission error for all transmissions) we get,

$$\begin{aligned}
D_{ee}[n] &= D_m + \rho^{2n}(\sigma_x^2 - D_q)P_f\\
&\quad + [(1-\rho^2)\sigma_x^2 - D_m]\rho^{2n}P_f\sum_{k=1}^{n}\rho^{-2k}.\\
&\overset{(\blacksquare)}{=} D_m + \rho^{2n}(\sigma_x^2 - D_q)P_f\\
&\quad + [(1-\rho^2)\sigma_x^2 - D_m]P_f\frac{1-\rho^{2n}}{1-\rho^2}.\\
&= D_m + \rho^{2n}(\sigma_x^2 - D_q)P_f\\
&\quad + (1-\rho^{2n})[\sigma_x^2 - \frac{D_m}{1-\rho^2}]P_f.
\end{aligned}$$

Where equality (■) follows from,

$$\begin{aligned}\sum_{k=1}^{n}\rho^{-2k} &= \sum_{j=0}^{n-1}\rho^{-2(j+1)}\\ &= \rho^{-2}\sum_{j=0}^{n-1}\rho^{-2j}\\ &= \rho^{-2}\frac{1-\rho^{-2n}}{1-\rho^{-2}}\\ &= \frac{\rho^{-2n}-1}{1-\rho^2}\end{aligned} \tag{3.41}$$

Note that $\frac{D_m}{1-\rho^2} = D_q$ if source coding rate is the same for $n = 0$ and $n > 0$.

$$\begin{aligned}D_{ee} &= \frac{1}{M}\Big[D_q + (M-1)D_m + P_f(\sigma_x^2 - D_q)\sum_{n=0}^{M-1}\rho^{2n}\\ &\quad + P_f[\sigma_x^2 - \frac{D_m}{1-\rho^2}](M-1-\sum_{n=1}^{M-1}\rho^{2n})\Big]\\ &\overset{\text{from (3.41)}}{=} \frac{1}{M}\Big[D_q + (M-1)D_m + \frac{P_f(\sigma_x^2 - D_q)\rho^{2M}}{1-\rho^2}\\ &\quad + P_f[\sigma_x^2 - \frac{D_m}{1-\rho^2}](M - \frac{1-\rho^{2M}}{\rho^2})\Big]\end{aligned}$$

Still when $\frac{D_m}{1-\rho^2} = D_q$, we have:

$$\boxed{\begin{aligned}D_{ee} &= \frac{1}{M}\Big[D_q + (M-1)D_m + \frac{P_f(\sigma_x^2 - D_q)\rho^{2M}}{1-\rho^2}\\ &\quad + P_f[\sigma_x^2 - D_q](M - \frac{1-\rho^{2M}}{\rho^2})\Big]\\ &= \frac{D_q + (M-1)D_m}{M} + P_f(\sigma_x^2 - D_q)\end{aligned}} \tag{3.42}$$

3.4.1 Suboptimal Error Concealment Distortion Analysis

In the previous section, to obtain expression for end-to-end distortion we used an optimal error concealment at the source decoder. To study the affect of suboptimal error concealment, we take into account two scenarios where we are using sub-optimal error concealment and for each case, we calculate the end-to-end distortion.

First, we consider the simple situation where we coded the sample with the DPCM encoder and concealed the corrupted sample received at the decoder with the expected value of the sample (*i.e.* $\tilde{x_j}[j] = 0$). We also assume the same error concealment for the memoryless coded sample (first sample). Note that this is a common error concealment method in scenarios which the source samples have little or no correlation. Calculating the end-to-end distortion for this case can be helpful by characterizing the upper bound of the overall distortion. To achieve this, we first need to calculate $D_{ee,j}[k]$,

$$
\begin{aligned}
D_{ee,j}[k] &= E[(x[k] - \tilde{x}_j[k])^2] \\
&= E[(x[k] - \tilde{x}_j[k])^2] + E[(\tilde{x}[k] - \tilde{x}_j[k])^2] - 2E[(x[k] - \tilde{x}[k])(\tilde{x}[k] - \tilde{x}_j[k])] \\
&= D_m[k] + E[(\tilde{x}[k] - \tilde{x}_j[k])^2] - 2E[(x[k] - \tilde{x}[k])(\tilde{x}[k] - \tilde{x}_j[k])] \\
&= D_m[k] + a^{2(k-j)}E[\tilde{e}[j]^2] - 2E[(x[k] - \tilde{x}[k])(\tilde{x}[k] - \tilde{x}_j[k])] \\
&= D_m[k] + a^{2(k-j)}E[(\tilde{x}[j])^2] \\
&\overset{(\bigstar)}{=} D_m[k] + a^{2(k-j)}\Big[a^{2j}(\sigma_x^2 - D_q) + [(1 - a\rho)\sigma_x^2 - D_m[j]] \\
&\quad + \sum_{n=1}^{j-1}[a^2(1 - a\rho)\sigma_x^2 - D_m[n]]\Big]
\end{aligned}
\tag{3.43}
$$

Where equality ($\bigstar$) follows from,

$$\begin{aligned}
E[(\tilde{x}[j])^2] &= E[(a\tilde{x}[j-1]+\tilde{e}[j])^2] \\
&= E[(a(a\tilde{x}[j-2]+\tilde{e}[j-1])+\tilde{e}[j])^2] \\
&\overset{\text{from (3.5)}}{=} E[(a^2\tilde{x}[j-2])^2]+E[(a\tilde{e}[j-1])^2]+E[(\tilde{e}[j])^2] \\
&\overset{\text{Applying recursion}}{=} E[(a^j\tilde{x}[0])^2]+\sum_{n=1}^{j-1}E[(a^{j-n}\tilde{e}[n])^2] \\
&= a^{2j}E[\tilde{x}[0]^2]+\sum_{n=1}^{j-1}a^{2(j-n)}E[\tilde{e}[n]^2] \\
&\overset{(\lozenge)}{=} a^{2j}(\sigma_x^2-D_q)+[(1-a\rho)\sigma_x^2-D_m[j]] \\
&\quad +\sum_{n=1}^{j-1}[a^{2(j-n)}(1-a\rho)\sigma_x^2-D_m[n]]
\end{aligned}$$

Where equality $\lozenge$ follows from replacing the $E[\tilde{x}[0]^2]$ and $E[\tilde{e}[n]^2]$ from equation (3.14) and (3.20), respectively.

By using (3.43) we can obtain a formula for $D_{ee}[n]$,

$$\begin{aligned}
D_{ee}[n] &= D_m[n]+a^{2n}(\sigma_x^2-D_q)P_0 \\
&\quad +\sum_{k=1}^{n}\Big(a^{2(n-k)}P_k\Big[a^{2k}(\sigma_x^2-D_q)+[(1-a\rho)\sigma_x^2-D_m[k]] \\
&\quad +\sum_{j=1}^{k-1}[a^{2(k-j)}(1-a\rho)\sigma_x^2-D_m[j]\ \Big]\Big).
\end{aligned} \tag{3.44}$$

In the second suboptimal scenario we consider a more general case by defining a random variable coefficient denoted as α which is defined as $\alpha \in [0,1]$. α is a coefficient that dictates how optimal we do the error concealment. Therefore, we can achieve the optimal case by assuming $\alpha = 1$ as studied in the previous section, or the least optimal concealment as we observed with $\alpha = 0$. It is important to understand the behavior of end-to-end

distortion with general α values. In this scenario a sample received in error is concealed as,

$$\tilde{x_j}[j] = \alpha a \tilde{x}[j-1]. \tag{3.45}$$

Similarly to the first case we can derive expression for the $D_{ee,j}[k]$,

$$\begin{aligned}
D_{ee,j}[k] &= E[(x[k]-\tilde{x}_j[k])^2] \\
&= E[(x[k]-\tilde{x}[k])^2] + E[(\tilde{x}[k]-\tilde{x}_j[k])^2] \\
&\quad -2E[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])] \\
&\overset{(\clubsuit)}{=} D_m[k] + E[(\tilde{x}[k]-\tilde{x}_j[k])^2] \\
&= D_m[k] + E[(a\tilde{x}[k-1]+\tilde{e}[k]-a\tilde{x}_j[k-1]-\tilde{e}[k])^2] \\
&\overset{\text{Applying recursion}}{=} D_m[k] + a^{2(k-j)}E[(a\tilde{x}[j-1]+\tilde{e}[j]-\alpha a\tilde{x}[j-1])^2] \\
&= D_m[k] + a^{2(k-j)}E[((1-\alpha)a\tilde{x}[j-1]+\tilde{e}[j])^2] \\
&= D_m[k] + a^{2(k-j)}E[((1-\alpha)a(a\tilde{x}[j-2]+\tilde{e}[j-1])+\tilde{e}[j])^2] \\
&= D_m[k] + a^{2(k-j)}\big((1-\alpha)^2a^4E[\tilde{x}[j-2]^2] \\
&\quad +(1-\alpha)^2a^2E[\tilde{e}[j-1]^2]+E[\tilde{e}[j]^2]\big) \\
&\overset{\text{Applying recursion}}{=} D_m[k] + a^{2(k-j)}\big((1-\alpha)^2a^{2j}E[\tilde{x}[0]^2] \\
&\quad +\sum_{n=1}^{j-1}(1-\alpha)^2a^{2(j-1-n)}E[\tilde{e}[n]^2]+E[\tilde{e}[j]^2]\big) \\
&\overset{(\bigstar)}{=} D_m[k] + a^{2(k-j)}\Big[(1-\alpha)^2a^{2j}(\sigma_x^2-D_q) \\
&\quad +[(1-a\rho)\sigma_x^2-D_m[j]] \\
&\quad +\sum_{n=1}^{j-1}[(1-\alpha)^2a^{2(j-1-n)}(1-a\rho)\sigma_x^2-D_m[n]]\Big],
\end{aligned} \tag{3.46}$$

where equation (♣) follows because,

$$\begin{aligned} E[(x[k]-\tilde{x}[k])(\tilde{x}[k]-\tilde{x}_j[k])] &= E\big[(e[k]-\tilde{e}[k])(\tilde{x}[k]-\tilde{x}_j[k])\big] \\ &\overset{(\diamondsuit)}{=} E[(e[k]-\tilde{e}[k])(a\tilde{x}[k-1]+\tilde{e}[k] \\ &\quad -a\tilde{x}_j[k-1]-\tilde{e}[k])] \\ &= E\Big[(e[k]-\tilde{e}[k])a^{(k-j)} \\ &\quad \big((1-\alpha)a\tilde{x}[j-1]+\tilde{e}[j]\big)\Big] \\ &\overset{\blacktriangledown}{=} 0, \end{aligned}$$

Also, equality (♢) follows from,

$$\begin{aligned} \tilde{x}[k]-\tilde{x}_j[k] &= a(\tilde{x}[k-1]-\tilde{x}_j[k-1]) \\ &= a^2(\tilde{x}[k-2]-\tilde{x}_j[k-2]) \\ &\overset{\text{Applying recursion}}{=} a^{k-j}(a\tilde{x}[j-1]+\tilde{e}[j]-\tilde{x}_j[j]) \\ &\overset{\text{from (3.45)}}{=} a^{k-j}(a\tilde{x}[j-1]+\tilde{e}[j]-a\alpha\tilde{x}[j-1]). \end{aligned}$$

Where equality (▼) follows from the fact that prediction error at different times are uncorrelated. This can be shown as,

$$\begin{aligned} E\Big[(e[k]-\tilde{e}[k])\big((1-\alpha)a\tilde{x}[j-1]+\tilde{e}[j]\big)\Big] &= E\big[(e[k]-\tilde{e}[k])(1-\alpha)a\tilde{x}[j-1]\big] \\ &\quad +E\big[(e[k]-\tilde{e}[k])\tilde{e}[j]\big] \\ &\overset{\text{from (3.3)}}{=} E\big[(e[k]-\tilde{e}[k])(1-\alpha)a\tilde{x}[j-1]\big] \\ &= E\big[(e[k]-\tilde{e}[k])(1-\alpha)a^2\tilde{x}[j-2] \\ &\quad +a\tilde{e}[j-1]\big] \\ &= E\big[(e[k]-\tilde{e}[k])(1-\alpha)a^j\tilde{x}[0] \\ &\quad +\sum_{n=1}^{j-1}a^{j-n}\tilde{e}[n]\big] \\ &\overset{(\blacksquare)}{=} 0, \end{aligned}$$

where equality (■) follows according to the orthogonality principle where,

$$E\big[(e[k]-\tilde{e}[k])\tilde{x}[0]\big] \overset{\text{from (3.3)}}{=} 0,$$

and

$$\begin{aligned} E\big[(e[k]-\tilde{e}[k])(\sum_{n=1}^{j-1}\tilde{e}[n])\big] &= E\big[e[k](\sum_{n=1}^{j-1}\tilde{e}[n])\big]-E\big[\tilde{e}[n](\sum_{n=1}^{j-1}\tilde{e}[n])\big]\big] \\ &\overset{(\square)}{=} 0. \end{aligned}$$

Equality (□) is equal to 0 because of the orthogonality principle and n being different from

k. Here we prove that it is the same even if $n = k$. we have,

$$\begin{aligned} E\big[(e[k]-\tilde{e}[k])(\sum_{n=1}^{j-1}\tilde{e}[n])\big] &= E\big[e[k]\tilde{e}[k]\big] - E\big[\tilde{e}[k]\tilde{e}[k]\big] \\ &\overset{\text{from (3.2)}}{=} E\big[\tilde{e}[k]^2\big] - E\big[\tilde{e}[k]^2\big] \\ &= 0. \end{aligned}$$

Note that the equation ($\bigstar$) is derived in a similar manner to (3.44) by using (3.14) and (3.20).

From (3.46), the end-to-end distortion can be derived as,

$$\begin{aligned} D_{ee}[n] &= D_m[n] + a^{2n}(\sigma_x^2 - D_q)P_0 \\ &+ \sum_{k=1}^{n}\Big(a^{2(n-k)}P_k\Big[(1-\alpha)^2a^{2k}(\sigma_x^2 - D_q) + [(1-a\rho)\sigma_x^2 - D_m[k]] \\ &+ \sum_{j=1}^{k-1}[(1-\alpha)^2a^{2(k-1-j)}(1-a\rho)\sigma_x^2 - D_m[j]\Big]\Big). \end{aligned} \tag{3.47}$$

In this section we studied the overall distortion performance of the proposed DPCM system. To obtain closed form expression for overall distortion we consider applying the optimal error concealment. Also we studied the same system under the assumption that error concealment is not necessary optimal. Studying a more general error concealment mechanism provides insight into the affect error concealment method in distortion performance.

3.5 Numerical Results and Discussion

In order to study the expression derived for the end-to-end distortion and error concealment optimality, we conducted numerical simulations. For this, we assumed that the number of samples $M = 30$, source samples power $\sigma_x^2 = 1$, $a = \rho$ and $\rho = 0.85$. Furthermore, we modeled the wireless fading channel as a finite state Markov process. A common model

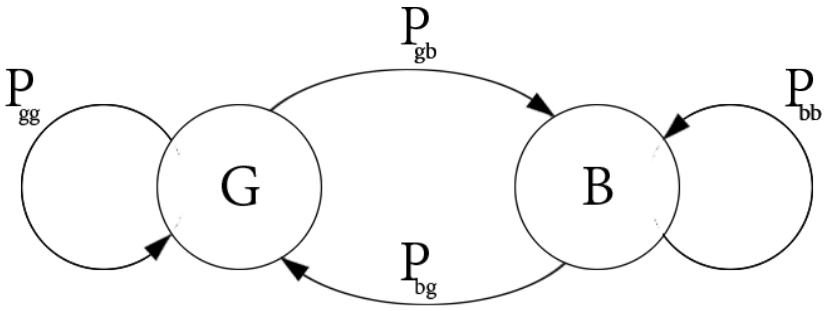

Figure 3.1: Gillbert Elliot channel model

for a wireless channel is the two-state Gilbert-Elliot model [13], as shown in Figure 3.1. As Figure 3.1 shows Gillbert-Elliot channel comprised of two states G (for good channel) and B (for bad or burst error channel). In the simulations, the probabilities for transition between states were chosen as $p_{gb} = 0.09$, $p_{bg} = 0.01$, $e_b = 0.111$, $e_g = 0.001$, for good state to bad state, bad state to good state, error at good state and error at bad state, respectively. We assumed different α values to be 0, 0.25, 0.75 and 1. The end-to-end distortion is computed using a Monte Carlo method through 100000 runs to model the Gilbert-Elliot channel. Figure 3.2 shows the result of the simulation and how different error concealment will affect the overall end-to-end distortion.

We also assume that the source coding rate is the same for $n = 0$ and $n > 0$. Figure 3.3 shows more feature for the end-to-end distortion when $\alpha = 1$. Notice that in the case of $\alpha = 1$, the end-to-end distortion almost appears to be constant, because the effect of transmission errors in DPCM, which propagates over time, is no worse than when using a memoryless codec, where there is no error propagation. In other words, the average distortion introduced by transmission errors is the same for the DPCM and the memoryless codec. To see this, recall that due to the assumption of optimal error concealment ($\alpha = 1$), same error probabilities ($P_0 = P_k = P_f$), and same source coding rate for all samples, it

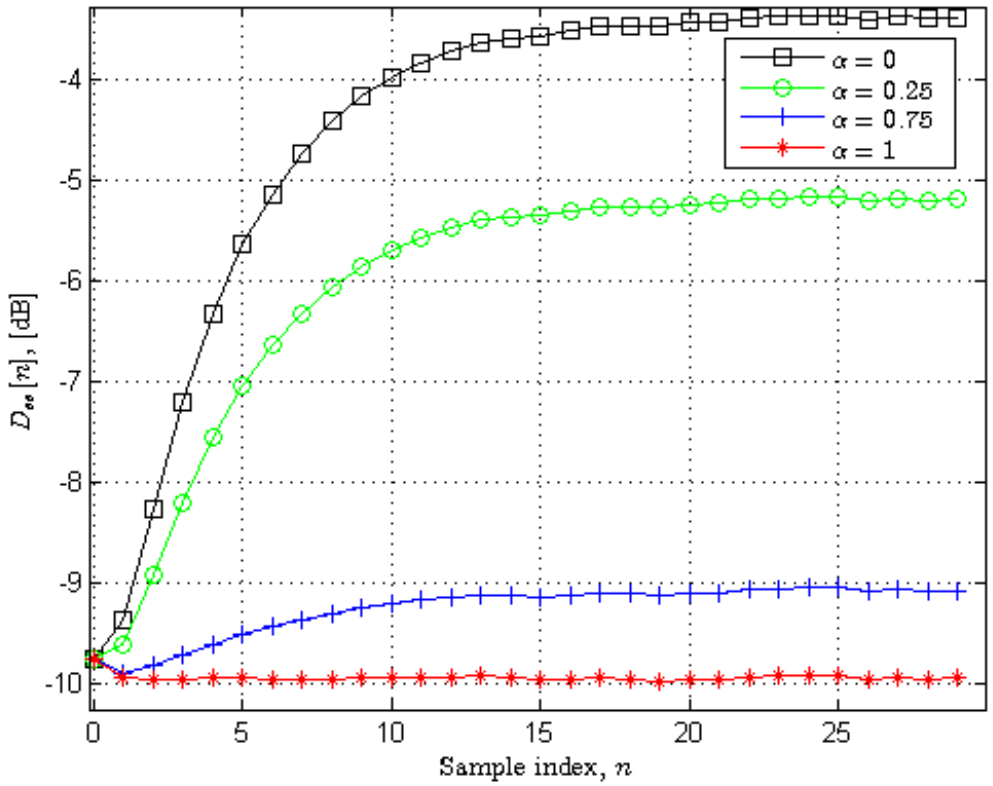

Figure 3.2: End-to-end distortion simulation of proposed work with different α coefficients

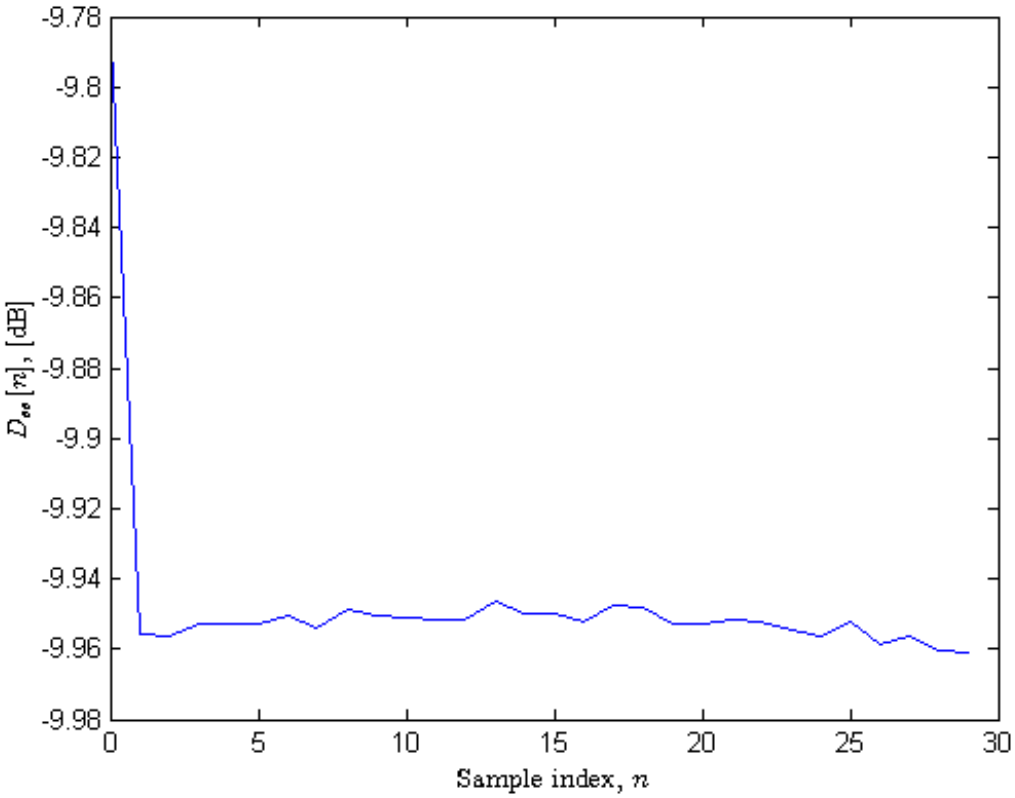

Figure 3.3: End-to-end distortion with $\alpha = 1$ coefficient

holds that $\frac{D_m}{1-\rho^2} = D_q$. Replacing this expression in (3.47) and calculating the sums yields,

$$D_{ee}[n] = D_m + \rho^{2n}(\sigma_x^2 - D_q)P_f + (1-\rho^{2n})\Big[\sigma_x^2 - D_q\Big]P_f,$$

and after simplifying,

$$D_{ee}[n] = D_m + \Big[\sigma_x^2 - D_q\Big]P_f = D_m + D_0[0]. \tag{3.48}$$

In this expression it is possible to recognize the distortion term due to source encoding (D_m) and the distortion term due to channel errors, $(\sigma_x^2 - D_q)P_f = D_0[0]$. Comparing this result with the corresponding to memoryless coding, (3.26), shows the different source encoding distortion (D_q), but the same channel-induced distortion. Therefore, in DPCM the channel induced distortion *is* made from the superposition of the distortions due to the propagation of all previously occurring errors. The distortion is attenuated by a factor ρ^2 each unit of time that it propagates through the predictive filter in the DPCM decoder. Moreover, the channel induced distortion that is introduced in DPCM, $D_j[j] = (1 - a\rho)[\sigma_x^2 - D_q]$ (from (3.46)), is a factor $(1 - \rho^2)$, the DPCM coding gain, smaller than the channel induced distortion introduced with memoryless coding. This is due to the introduced error being already smaller because the error concealment is taking advantage of the memory in the source. Because the error concealment takes advantage of the same prediction mechanism that yields the source coding gain in DPCM, now the channel-induced distortion exhibits the same gain in reducing the error. All in all, the overall distortion is made from the superposition of distortions that are attenuated over time and, when introduced, already smaller than the memoryless coding distortion.

3.6 Conclusion

In this chapter, we have derived a closed-form expression for the end-to-end distortion of a DPCM-type codec system followed by investigating the effect of error concealment method. Furthermore, it was shown that both a DPCM-type codec and memoryless codecs suffer the same distortion due to a transmission error, whereas it was a common misconception that transmission errors would affect the performance of DPCM-type codec significantly more.

Chapter 4

Distortion Optimized Joint Source Channel Coding using Incremental Redundancy

Real-time transmission of multimedia sources such as video streaming, video conferencing and videophone, require adhering to stringent quality of service (QoS) constraints. As it is elaborated in [14] one way to address these constraints is to use an error control mechanism.

There are two basic error control methods that are used in practice: feedback-based error control and forward error correction (FEC). Each has its benefits and drawbacks in regards with error robustness and network traffic load [15], [16].

Of the two mentioned error control methods, feedback-based error control such as Hybrid ARQ works based on the assumption that usually the probability of receiving data protected with a weak channel code is still relatively large. As a result, such systems are not using a large number of redundant bits in every transmission. On the other hand, FEC incurs constant overhead even when there is no error happened during transmission, and hence feedback based systems can provide a significant gain in throughput over no feedback FEC based systems. However, ARQ systems are not widely used in real-time multimedia streaming applications due to delay related issues such as delay accumulation, delay jitter, and complex buffer management.

In this study, we will focus on a real-time wireless communications system with Hybrid ARQ feedback under a delay constrained scenario. This setup results in a gain in

throughput over pure FEC-based techniques by achieving lower end-to-end distortion using feedback while addressing the delay issues. The advantage of using ARQ for real-time applications or a delay constrained application layer has been shown in [15], [18]. We achieve this by designing the system that combines incremental redundancy error control with feedback and joint source-channel coding. As a result, the proposed system transmits fixed size frames that operates with a constant bit rate and fixed low delay, and hence, this system enjoys the synchronous transmission of no feedback transmission schemes.

In a typical conversational communication scenario, devices that are intended to communicate will be connected in conjunction with subsystems either in the same or different network. Because each of this devices will add to the overall delay, it is important to design a system with small delay. In the proposed scheme, the novel contribution is that we design a stringent low delay joint source-channel coding for a source with memory and coded using DPCM. This delay, is equal to one source encoder period. This strict low delay is met continuously by limiting the transmission of incremental redundancy bits to no more than one time, while also changing the source and channel coding rates so as to accommodate the incremental redundancy bits, and to maintain a constant transmit bit rate.

There has been considerable research in employing a joint source-channel coding (JSCC) framework to design an optimal error control for real-time multimedia transmission under delay constraint, e.g. [19], [20], [21] ,[22], [23], [24], [25], [26], and [27]. Typically, JSCC is achieved by the jointly design of the entropy coder and the quantizer for given channel errors, as described in [19]. For multimedia application, JSCC requires accomplishing three tasks: obtaining an optimal bit allocation between source coding and channel coding for given channel loss characteristics; designing the source coding to achieve the target source rate, and developing the channel coding to achieve the required robustness [28], [29]. Joint

source coding and FEC has been extensively studied in [30]. The author proposed a scheme for joint optimization of channel coding rate allocation at different source layers, and the power allocation at different CDMA channels, to minimize the distortion of the received video data. Furthermore, through simulations it was shown that the proposed joint optimal FEC/power allocation scheme offers better performance over the system with optimal FEC and equal power levels.

In [4], Kwasinski et al. studied the delay-constrained real-time transmission of a memoryless source, where a feedback-based incremental redundancy error control scheme (Hybrid ARQ) is used. Unlike retransmission based schemes, the proposed system is designed to work with a constant bit-rate. Furthermore, the authors introduced the notion of code-types that is used to model the transmission mechanism with a Markov chain. Using this model, a novel algorithm was developed to address the optimal source and channel rate allocation. In this study, it is showed that the proposed system captures the benefits of feedback, while it allows the synchronous transmission similar to pure FEC systems. Furthermore, the result suggested that the proposed mechanism obtains higher channel SNR gains over a pure FEC based scheme. In the case of a CDMA network, the higher SNR gain can be translated into an increase in the number of users that can be simultaneously supported at the same level of distortion.

Another work related to this thesis is by Chou et al., which considers a receiver-driven Hybrid FEC/pseudo-ARQ mechanism so as to optimize error control for streaming applications [16]. In this study, feedback is proposed as an operation of joining and soon leaving a multicast group carrying delayed versions of additional parity data. The receiver-driven scheme arises when each receiver estimates its available bandwidth and joins the multicast groups so as to fill the available bandwidth. In this work, the available bandwidth

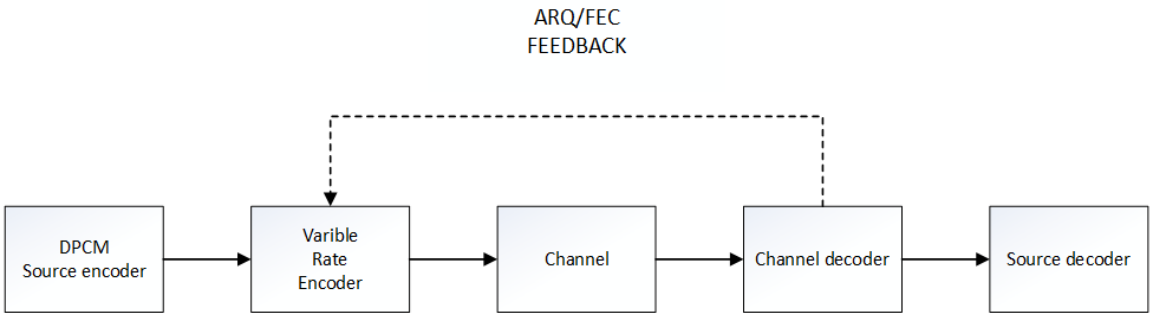

Figure 4.1: Block diagram of DPCM coded with ARQ/FEC transmission scheme

is calculated by measuring packet losses. The error control is also being deployed to use the available transmission rate to minimize the effect of packet loss. Their findings support that, an appropriately designed combination of unequal FEC and receiver-driven soft ARQ, can yield significant gains in performance over pure FEC systems, with and without unequal error protection. Their work was designed primarily for streaming applications, where it is appropriate to have a large initial delay. Given a total bit rate constraint for each of the source blocks layers, this work presented an optimal solution to the problem of optimal bit allocation to source and channel coding for all possible transmission of additional parity. However, this work provides a suboptimal solution to the problem of maintaining a constant transmit bit rate.

4.1 System Description

The main components of the proposed system are shown in Figure 4.1. The source and its source encoder and decoder are the same as described in the previous chapter. We assume that the DPCM source encoder input is a real-time source. In order to work with a real-time source, the source encoder is required to operate at several different coding rates. There are two ways of implementing this by using either variable rate or embedded

codecs. A variable-rate source encoder generates one-bit stream for each of the possible encoding rates. Only one of these bit streams will be chosen and transmitted based on the rate assignment. On the other hand, embedded encoder generates only one-bit stream, and the rate adaptation process works by cutting bits from the end of the only generated one-bit stream. Therefore, embedded encoder process is simpler than variable rate encoder. In our model, the variable rate encoder is used because of its better performance compared to an embedded encoder. The DPCM source encoder output is divided into source blocks. The length of these source blocks may change based on the different encoding rates. Then in the next step the redundancy bits (channel coding bits) are added to protect the source blocks from the error-prone channel. We want to alter the code rate or in another word, the number of redundant bits, hence the correction power of the code during transmission of an information frame can be modified proportionate to the source and channel needs. For practical purposes, we would like to operate on one encoder and one decoder instead of switching between a set of encoders and decoders. This can be achieved by using punctured codes. In a rate-compatible punctured convolutional codes (RCPC), codes can be arranged in descending order of rate, where every high rate code is achieved by puncturing a lower rate code. In other words, we can switch from a weak code (high rate) code to a stronger code (low rate) by appending some extra parity check symbols to a codeword of a weak code. These extra parity check bits which are called ”incremental redundancy” are the ones that are punctured from the weaker code and available in the stronger one.

It is important to use the feedback channel efficiently so as to maximize the end-to-end quality of the multimedia source, for a given transmission rate. It is well known in Information Theory that the capacity of a binary symmetric channel does not increase with the feedback. Nevertheless, it has been shown in many practical systems, that proper use of

feedback such as Hybrid Automatic ReQuest (ARQ)/Forward Error Correction (FEC) will lead to a better bandwidth utilization over no feedback systems such as pure FEC.

There are more complicated ways of utilizing the feedback channel other than the only use of single bit ACK/NACK feedback. There are many such protocols such as a likelihood ratio feedback, complete information feedback, and channel state feedback. There are pros and cons associated with each method, for instance, transmission of the floating point numbers for the likelihood ratio feedback is more demanding than only sending one symbol to compare to ACK/NACK protocol. Also, the complete information feedback may require larger bandwidth in reverse than in the forward direction. Besides, the possibility of channel errors in the feedback channel needs to be addressed sufficiently. Using only one bit to demonstrate the two values of feedbacks have the advantage that it is simple to generate, requires low bandwidth for the reverse transmit direction and can be protected easily with error correcting codes. On the other hand, this simplicity has the disadvantage of suboptimality since the feedback information only answers the question of whether the source was received with an error or not.

We assume that transmission over the wireless channel occurs in fixed-size frames. For each frame, one source block is generated. Then, at the receiver side each of these source blocks are decoded to retrieve the frame. If the receiver can successfully recover the transmitted information bits from the perhaps corrupted received channel symbols, the decoding process will be successful, and one bit (ACK) will be sent through a feedback channel. Otherwise, an error detection code will recognize the presence of errors and as a result, the decoding process will fail, and the one bit ”NACK” will be transmitted to inform the sender that additional transmission of the same information bits may be required. The ACK/NACK generation is usually implemented through an error detection mechanism like

cyclic redundancy check (CRC). Additionally, we assume the feedback channel to be error free and instantaneous.

We use the following protocol to configure each frame for transmission based on the feedback that is received from the previous frame. Assume two consecutive frames referred in time as the current and the next frames and their corresponding source blocks as the current and the next source blocks. If an ACK is received for the current frame, then the next frame will only be comprise entirely of the bits of the codeword of the current source block. On the other hand, if a NACK is received, the encoder, if it is possible, chooses a lower rate channel code for the current source and will send the corresponding incremental redundancy bits as part of the next source block. Recall that because of the rate compatibility property of the channel codes using (2.18) and (2.19), the lower rate can be accomplished by inserting some extra bits to the next frame. These bits are the redundancy bits for the lower rate channel code which previously were not sent. Note that since these incremental redundancy bits are sent as part of the next frame the constraint of a fixed frame size forces a change in the source coding rate. Upon receiving the next frame, if it contains incremental redundancy for the current frame, in addition to decoding the next frame to retrieve the next source block the receiver channel decoder will combine the incremental redundancy to attempt a second decoding try of the current frame to the best of its ability. Note that by using this procedure we constrain the delay incurred by incremental redundancy transmission to a maximum of one extra frame transmission. These delay boundaries are required for real-time communication, specifically. Because of the restriction of sending only one set of incremental redundancy per source block, if transmission fails after the last attempt error concealment is applied and there is no further transmission of feedback. All the frames are bound to use this protocol so depending on the immediately previous frame each of the

frames may comprise of either a codeword or a codeword plus incremental redundancy.

The benefit of the proposed system is that it changes the source coding rate so as to accommodate the transmission of incremental redundancy bits. As a result, unlike a typical retransmission based system, the delays will not accumulate, and the system satisfies the synchronized nature of real-time applications. Furthermore, the synchronous operation eliminates unnecessary complexities related to unknown buffer size and delay jitter at the receiver side. In our case, the memory size of the buffer will be one frame.

4.2 System Design

4.2.1 Mathematical Setup

As mentioned, a frame is either composed of one codeword or one codeword and incremental redundancy that is sent for the previous source block codeword. We introduce the notion of codetype to describe the different frame compositions. We consider the codetype of a codeword to be "1" if there is no incremental redundancy being sent and its bits occupy the entire frame. Also, the codetype of a codeword is "m" if the frame contains incremental redundancy bits for the codetype "$m-1$". We also consider there is a maximum number of N different codetypes. Let m_n denote the codetype of codeword used while sending the n^{th} frame, and F_n is the feedback received after the n^{th} frame. Then the various states of the codetype can be described as,

$$m_n = \begin{cases} 1 & \text{if } F_{n-1} = ACK, \\ m_{n-1}+1 & \text{if } F_{n-1} = NACK \text{ and } m_{n-1} < N, \\ N & \text{if } F_{n-1} = NACK \text{ and } m_{n-1} = N. \end{cases} \tag{4.1}$$

To explain the different type of information that constitutes the frame with a codeword of codetype m, we recall that there are source bits, channel codeword bits, and incremental redundancy bits. Let $r_s(m)$ denote the number of source bits, $r_c(m,0)$ denote the number of channel code parity bits associated with the current source bits for the first attempt, and $r_c(m,1)$ denote the number of incremental redundancy bits. When a frame with a codeword of codetype m, is received the channel decoder tries to decode the $r_s(m)$ source bits, from the received word of length $r_s(m)+r_c(m,0)$ in the first attempt and the word of the length $r_s(m)+r_c(m,0)+r_c(m,1)$ for the second attempt. Note that notwithstanding that our model description is based on a systematic channel code, this is not actually necessary for our design.

The performance of channel codes can be measured by the Frame Error Rate (FER). In this work, we denote the minimum Frame Error Rate (probability of uncorrectable post-decoding error) as $FER(r_s,l)$ for a channel code from the code family which encodes a minimum of r_s bits into a maximum of l bits. Also, let $p(m,0)$ denote the probability of a frame error rate after the first decoding attempt for codetype m, then the probability of a NACK feedback is,

$$p(m,0) \stackrel{\text{def}}{=} FER(r_s(m), r_s(m)+r_c(m,0)). \tag{4.2}$$

In addition to $p(m,0)$, we define $p(m,1)$ as the probability of a residual frame error after the second attempt, *i.e.* after sending the incremental redundancy, as

$$p(m,1) \stackrel{\text{def}}{=} FER(r_s(m), r_s(m)+r_c(m,0)+r_c(m,1)). \tag{4.3}$$

Note that the value of $p(m,0)$ and $p(m,1)$ are associated with the performance of channel

coding system. In our case, since we are using RCPC codes these probabilities can be derived using (2.23).

Consider a first autoregressive, AR(1), source model similar to the previous chapter. We assume the frame encoded with r_s source bits through DPCM source encoder, r_{c0} parity bits by channel encoder (RCPC code) for the first transmission attempt and r_{c1} parity bits in the incremental redundancy transmission. Then the end-to-end distortion can be expressed as

$$D_{ee}(r_s, r_{c0}, r_{c1}) \stackrel{\text{def}}{=} \big(1 - FER(r_s, r_s + r_{c0} + r_{c1})\big) D_m(r_s) + FER(r_s, r_s + r_{c0} + r_{c1}) D_c(r_s, r_{c0} + r_{c1}). \tag{4.4}$$

Note that here $D_m(r_s)$ is the distortion suffered due to the source encoder when using the source rate of r_s and can be derived through (3.10). Also, $D_c(r_s, r_{c0} + r_{c1})$ is the channel-induced distortion once the frame error happens. The $D_m(r_s)$ is only a function of the source rate r_s, while D_c depends on r_s , $r_{c0} + r_{c1}$, error concealment mechanism as described in DPCM and channel SNR. Since we are using the DPCM encoder to encode the source data, the overall distortion also will follow the similar terms computed previously. To describe this, we assume the (3.48), under the same conditions that led to that approximation, for the distortion using (4.4) we have,

$$D_{ee}(r_s, r_{c0}, r_{c1}) \stackrel{\text{def}}{=} \big(1 - FER(r_s, r_s + r_{c0} + r_{c1})\big) D_m + FER(r_s, r_s + r_{c0} + r_{c1})(\sigma_x^2 - D_q). \tag{4.5}$$

This is similar to the end-to-end distortion that is derived for DPCM in (3.48), with the major difference that the probability of error in this scheme depends on the properties of channel code and feedback based transmission system. Note that the probability of receiving NACK after first transmission attempt does not directly appear in our expression

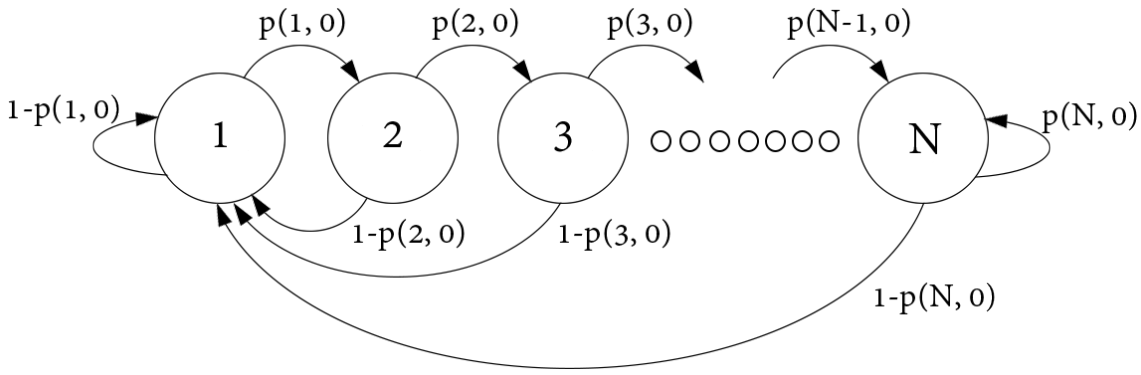

Figure 4.2: Markov chain of codetypes

for the average distortion.

Considering the equation (4.5) for the evolution of codetypes, it can be seen that each codetype only depends on the previous codetype. We can use Markov chain to model this evolution of codetypes for transmission over the memoryless channel. The relation between codetypes can be shown in Figure 4.2. Note that each state is associated with only one codetype. Also, each transition is conditioned by the feedback that is transmitted from receiver to transmitter. The transmission of codetype m^{th} may take place only if the feedback after sending the $(m-1)^{th}$ was a NACK. Using this model we can obtain the stationary distributions for each of the codetype denoted as π following,

$$\pi_1 = \left(1 + \sum_{j=1}^{N-2}\left(\prod_{i=1}^{j} p(i,0)\right) + \frac{\left(\prod_{i=1}^{N-1} p(i,0)\right)}{1-p(N,0)}\right)^{-1} \tag{4.6}$$

$$\pi_m = \pi_1 \prod_{i=1}^{m-1} p(i,0) \quad \text{for } m = 2, \ldots N-1 \tag{4.7}$$

$$\pi_N = \frac{\pi_1}{1-p(N,0)} \prod_{i=1}^{N-1} p(i,0). \tag{4.8}$$

Using the equation (4.5) we can write the expression to calculate the end-to-end distortion

associated with each codetype $D_{ee}(m)$ as,

$$D_{ee}(m) = (1 - p(m,1))D_m + p(m,1)(\sigma_x^2 - D_q). \tag{4.9}$$

We can describe the average distortion for the entire transmission session as the product of the probability of each state and the distortion occurs in that state as follows,

$$D_{ee} = \sum_{m=1}^{N} D_{ee}(m)\pi_m. \tag{4.10}$$

Now, if we assume that the value of probability of post-decoding error is insignificant, then the average distortion of packets not suffering from post-decoding channel induced errors is given by,

$$D_{ee} = \frac{\sum_{m=1}^{N} \pi_m(1 - p(m,1))D_m(r_s(m))}{\sum_{m=1}^{N} \pi_m(1 - p(m,1))}. \tag{4.11}$$

4.2.2 Dynamic Programming Solution

According to the fixed length frame constraint that we have mentioned above, we now introduce a dynamic programming based technique that can find the optimal solution to assign source coding, channel coding, and incremental redundancy in a 2-step transmission framework (*i.e.* sending incremental redundancy rates at most once for each source block).

The given optimal solution will be computed such that the average end-to-end distortion in (4.10) be minimized.

4.2.3 Distortion as a Ratio of Costs

To obtain the optimal solution, first, we introduce the notion of distortion as a ratio of costs. Similar to the notation we used in the chapter 2.4, let Ω denote the set of all possible valid code assignments. Each element of this set can be specified by the three associated values of the source block size, some channel code parity bits in the first transmission attempt, and some incremental redundancy bits for each of the N codetypes. Since we have a maximum number of N codetypes then Ω is a finite set. Ω describes the combination of available operating modes in both the source and channel encoders restricted by the fixed frame size condition. Let some random valid code assignment be denoted by $\mu \in \Omega$. Also, let us denote $p_\mu(i,0)$ and $p_\mu(i,1)$ as the probability of post-decoding errors after the first and second transmission attempt while sending the i^{th} code used in the assignment, respectively. Note that, since we are limited to send only up to one incremental redundancy then the $p_\mu(i,1)$ is best described as the residual FER. Note that this residual error is closely related to the error concealment mechanism that used by the source decoder. We assume the same error concealment as explained in previous chapter (*i.e.* using the previous reconstructed source sample). In the same manner, let $r_{s\mu}(i)$, $r_{c\mu}(i,0)$ and $r_{c\mu}(i,1)$ denote the number of source bits, the number of channel bits in the first attempt and number of channel bits in the second attempt (*i.e.* adding incremental redundancy bits to the first attempt channel bits), of codetype i of assignment μ, respectively. Also, let $D_{ee}\mu(i)$ defined as in (4.9) denote the distortion that is suffered due to sending the codetype i of the assignment μ. Then using (4.10) we can write D_{ee} as

$$\frac{D_{ee}\mu(1) + \sum_{j=1}^{N-2}\left(\prod_{i=1}^{j} p_\mu(i,0)\right) D_{ee}\mu(j) + \frac{\left(\prod_{i=1}^{N-1} p_\mu(i,0)\right)}{1-p_\mu(N,0)} D_{ee}\mu(N)}{1 + \sum_{j=1}^{N-2}\left(\prod_{i=1}^{j} p_\mu(i,0)\right) + \frac{\left(\prod_{i=1}^{N-1} p_\mu(i,0)\right)}{1-p_\mu(N,0)}} \tag{4.12}$$

This expression is the ratio of two cost function based on the Markov chain stationary distribution. The numerator and denominator are the average distortion value and an average number of steps starting at codetype 1 before returning to codetype 1.

4.2.4 Optimization Problem

To solve the optimization problem described for the distortion as ratio of cost we consider the expression used in (4.12). Let $\delta(\mu)$ and $R(\mu)$ denote the numerator term and denominator term of D_{ee} in (4.12) when code assignment μ is used. Therefor, for the codetype μ the optimization problem can be expressed as

$$\min_{\mu\in\Omega} \frac{\delta(\mu)}{R(\mu)}. \tag{4.13}$$

Note that there are multiple constraints as mentioned earlier which all code assignments $\mu \in \Omega$ must satisfy. The total number of bits denoted as W. Hence the limitations for the given codetype μ can be stated as,

$$r_{s\mu}(m) + r_{c\mu}(m,0) + r_{c\mu}(m-1,1) \leq W \tag{4.14}$$

$$r_{s\mu}(1) + r_{c\mu}(1,0) \leq W \tag{4.15}$$

$$r_{s\mu}(N) + r_{c\mu}(N,0) + r_{c\mu}(N,1) \leq W \tag{4.16}$$

4.2.5 Minimum as a Zero of the Lagrangian

To solve the end-to-end distortion optimization problem (4.12) through constraints expressed in (4.14), (4.15) and (4.16), we used Lagrangian optimization. Using Lagrangian optimization for rate control under multiple rate constraints was previously studied in depth in [31]. This problem emerges in many practical situations where the goal is to minimize the total end-to-end distortion by appropriately assigning various rates to the source blocks and channel rates without transcending a prespecified rate budget. To address the problem described in (4.13), let λ be a real non-negative number. Define for a code assignment μ, the Lagrangian $\mathcal{L}_\mu(\lambda) \stackrel{def}{=} \delta(\mu) - \lambda R(\mu)$ and consider the optimization problem,

$$\min_{\mu\in\Omega} \mathcal{L}_\mu(\lambda) = \min_{\mu\in\Omega} \delta(\mu) - \lambda R(\mu). \tag{4.17}$$

The problem that remains is to find out the appropriate multiplier λ such that no constraint is violated. $\mu^*(\lambda)$ denote the assignment achieving the minimum in the problem (4.17). If we assume that for some multiplier λ^* the optimization problem $\min_{\mu\in\Omega} \mathcal{L}_\mu(\lambda) = 0$ then, $\lambda^* = \min_{\mu\in\Omega} \frac{\delta(\mu)}{R(\mu)}$ and $\mu^*(\lambda^*)$ is the solution to optimization problem. This is due to the fact that for all $\mu \in \Omega$ we have $\frac{\delta(\mu)}{R(\mu)} \geq \lambda^*$, as a result $\mathcal{L}_\mu(\lambda^*) = \delta(\mu) - \lambda^* R(\mu) \geq 0$ and minimum is zero when $\mu = \mu^*$.

Accordingly, using (4.12) and (4.17) the optimization problem can be written as,

$$\begin{aligned}\delta(\mu) - \lambda R(\mu) &= [D_{ee}\mu(1) - \lambda] \\ &+ \sum_{j=1}^{N-2} \left(\prod_{i=1}^{j} p_\mu(i,0) \right) [D_{ee}\mu(j) - \lambda] \\ &+ \frac{\left(\prod_{i=1}^{N-1} p_\mu(i,0) \right)}{1 - p_\mu(N,0)} [D_{ee}\mu(N) - \lambda]\end{aligned} \tag{4.18}$$

4.2.6 Optimization by Dynamic Programming

The optimization problem discussed above can be translated into a discrete optimization problem. Since the number of codetypes are restricted to be a maximum N codetype. Therefore, the operating mode set is also a finite set. Hence, the optimization problem can be mapped into a finite horizon Markov Decision Process (MDP) problem. Now we solve the problem in (4.17) for a fixed λ by Dynamic Programming. Consider (4.19) we denote $J_\mu(N,\lambda)$ and $J_\mu(m,\lambda)$ to be the cost functions for the specific λ and codetype N and m respectively. This can be defined as

$$\begin{aligned}J_\mu(N,\lambda) &= \frac{(D_{ee}\mu(N) - \lambda)}{1 - p_\mu(N,0)} \text{ and} \\ J_\mu(m,\lambda) &= (D_{ee}\mu(m) - \lambda) + p_\mu(m,0) J_\mu(m+1,\lambda),\end{aligned} \tag{4.19}$$

$$\text{for } m = 1, 2, \ldots\ N-1,$$

then $\delta(\mu) - \lambda R(\mu) = J_\mu(1, \lambda)$. Note that

1. $J_\mu(m, \lambda)$ depends only on the assignments of the codes for codetypes $m, m+1, \ldots N$ and not on the earlier codewords. This will suffice the principle of optimality. The tail policy states starting from the end policy and do the optimization without considering the past occurrences and continuing the process until reaching the first state, then the whole system is optimized. According to the principle of optimality, the tail policy is optimal for the tail subproblem. Alternatively, optimized solution prescribes that optimal solution for the current state is accomplished regardless of what occurred in the past.

2. On the other hand, the constraints in equation (4.14) relates codetype m to codetype $m-1$ only through the parameter $r_{c\mu}(m-1, 1)$, as $r_{s\mu}(m) + r_{c\mu}(m, 0) = W - r_{c\mu}(m-1, 1)$.

Therefore, the following algorithm can be formulated. Let $r_{smin} + r_{cmin} < l \leq W$ where $r_{smin} + r_{cmin}$ is the smallest possible length of a codeword in the available channel code family. Then define

$$J_N^*(l, \lambda) = \min_{r_s, r_{c0}} \frac{(D_{ee}(r_s, r_{c0}, r_{c1}) - \lambda)}{1 - FER(r_s, l)}$$

$$\text{subject to } r_s + r_{c0} = l. \tag{4.20}$$

D_{ee} is the end-to-end distortion when using DPCM. $J_N^*(l, \lambda)$ denotes the smallest value of $J_\mu(N, \lambda)$ satisfying the constraint on the final frame error rate.

Similarly, for $m = 1, 2, \ldots N-1$, define recursively

$$J_m^*(l,\lambda) = \min_{r_s, r_{c0}, r_{c1}} (D_{ee}(r_s, r_{c0}, r_{c1}) - \lambda) + FER(r_s, l) J_{m+1}^*(W - r_{c1}, \lambda)$$

$$\text{subject to } r_s + r_{c0} = l. \tag{4.21}$$

Note that this is the Bellman equation for optimality. The value $J_1^*(W,\lambda)$ is the value of the minimum in problem (4.17) and the minimizing parameters in the above optimization give the code for m^{th} codetype in $\mu^*(\lambda)$. The reason for that is because we recursively solve the optimization problem for the tail subproblem. This process will extend until we reach the end of the recursion process which is the first codetype.

4.2.7 Policy Iteration algorithm for Optimal λ

Algorithm 1: Policy Iteration DP

1. Set $\lambda_0 = a$ (arbitrary constant).
2. Obtain $\mu^1 = \mu^*(\lambda_0)$ by equations 4.20 and 4.21.
3. $k = 1$.
4. Set $\lambda_k = \frac{\delta(\mu^k)}{R(\mu^k)}$.
5. Set $\mu^{k+1} = \mu^*(\lambda_k)$ by solving equations 4.20 and 4.21.
6. If $\mu^{k+1} \neq \mu^k$ Set $k = k+1$ and go to Step 4. Else, stop. The converged assignment is the optimal assignment and the final λ is the optimal λ.

The above dynamic programming algorithm (4.21) can obtain for any λ. the optimal code assignment from Ω. But we are interested in the specific value of λ which yields $\mathcal{L}_{\mu^*(\lambda)}(\lambda) = 0$. The optimal value of λ can be obtained by policy iteration algorithm [32]. The policy iteration algorithm obtains, at each step, the policy for each of the codetypes

starting at N^{th} codetype and update the value of λ. This algorithm guarantees that in a finite number of steps the assignment converges to the optimal. Note that the complexity of the Algorithm 1 depends on the number of the defined codetypes. Similar to water filling problem, here we try to maximize the number of source bits. The greedy algorithm attempts to minimize the distortion.

4.3 Simulation Results

To evaluate and verify the performance of our system we used the first Autoregressive, AR(1), source with zero mean and variance of one. For the source encoder we assume DPCM source encoder with similar architecture that we explained in the previous chapter. For the variable-rate channel encoder, we used a memory 4, puncturing period 8, RCPC code [9] decoded with a soft Viterbi decoder. The constant frame size was chosen to be equal to 400 bits ($W = 400$) as discussed to comply with the constant frame size to maintain the synchronous nature of real-time communication. In all cases, we considered communication using BPSK modulation over an AWGN channel. CRC check is done at the receiver, for error detection.

We considered a model with no FER constraint independent of whether the feedback is used or not. Our design corresponds to the optimum source, and channel rate allocation so as to minimize the end-to-end distortion (i.e. including channel induced distortion). Unlike, the design with FER constraint, in the no FER constraint scenario, source coding does not contribute to most of the end-to-end distortion since the effects of more annoying channel-induced errors is also considered. We compare four systems in this section. System A, system B, system C and system D. Main difference between system A,C and system B,D is the transmission mechanism and whether feedback is used or not. Let us first introduce

the designed system with Hybrid ARQ transmission mechanism that uses ACK/NACK feedback, System A, and System C. In this mechanism we assume to have a system that works with two codetypes. A similar system to the Figure 4.2, with $N = 2$ in this model. It is important to notice that because $N = 2$ we also have maximum two operating modes. These operating modes are acquired through the use of Dynamic Programming algorithm. Note that the transition probability between the two states depends on the RCPC decoding performance as we discussed in Equation (2.20).

System A is designed based on practical limitations with the correlation factor in DPCM. Recall the Equation (3.1), where the source encoder codes the next source block based on the correlation between the current source block and the previous source block. In system A, we assume this correlation to be less than compare to system C. To configure this, we assume that the source correlation coefficient for system A to be $\rho = 0.70$ compare to system C, which is $\rho = 0.85$. It is evident from (3.10), that decreasing the value of ρ results in higher source encoding distortion, hence, higher overall distortion. Also note that channel-induced distortion depends on the error concealment method that is carried out at the source decoder.

On the other hand, systems B and D are designed based on the idea of only sending one source block, and if it fails, we perform error concealment without holding to receive incremental redundancy. This makes these systems highly conservative. With only using one codetype, the rate optimization problem will assign rates so as to minimize the overall distortion. Therefore, the only codetype must be robust enough to compensate for the lack of retransmissions. Therefore, often in these systems the channel codes are chosen to be of lower rates compared to the systems that work with incremental redundancy. The difference between system B and D is similar to the difference between systems A, D, with

source correlation coefficient values $\rho = 0.7$ and $\rho = 0.85$, respectively.

Figure 4.3 shows the results for the overall distortion of the four mentioned systems. To solve the rate allocation problem we consider (4.20) and (4.21) through the policy iteration dynamic programming. The optimal policy leads to optimal distortion for each system. The optimal policy will vary based on the channel SNR values. We simulate our result by assuming the channel SNR values are discrete and varying from $-1.5\ dB$ to $2\ dB$. Note that according to the Figure 4.3 the feedback based scheme obtains useful gains over no feedback (*i.e.* pure FEC) mechanism. In fact, feedback based with the same correlation coefficient factor consistently achieving better distortion performance. Also, note that moving towards lower channel SNR results in increase in channel-induced distortion and as a result increase in an overall distortion. Consequently, since the value of the channel-induced distortion is independent of ρ from (3.48), we see that system B and D suffer similar distortion when the SNR value is $-1.5\ dB$.

Note that channel induced distortion depends on the error concealment method. We assume optimal error concealment as described in previous chapter. With or without feedback we conceal a source block by replacing it with its predecessor source block. However, there is a subtle difference between the concealment mechanism between two with or without a feedback system. In the systems with feedback, we send incremental redundancy bits while lowering down our source rate to accommodate the necessary bits. Therefore, we wait for the duration of one frame and then if it failed we do error concealment. In systems without feedback the error concealment operation is done immediately after error detection.

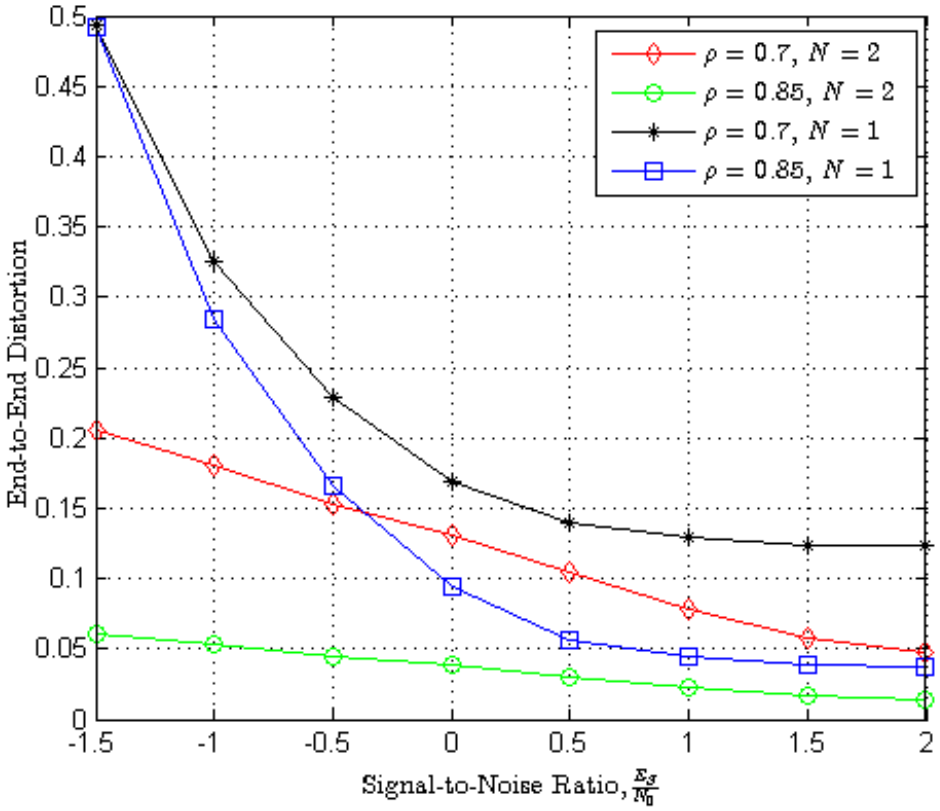

Figure 4.3: End-to-end distortion vs. E_s/N_0 for AWGN channels with and without feedback

Chapter 5

Conclusions

In this work, we considered the transmission schemes of loss-tolerant multimedia sources over noisy and lossy channels. In order to mimic the essential features of typical multimedia traffic we studied the first order autoregressive synthetic source. In order to obtain high efficiency in transmitting of multimedia source, we propose the use of predictive coding mechanism. Therefore, the end-to-end distortion is studied for a DPCM source decoder. This study showed that predictive coders when applying optimal error concealment suffers the same channel-induced distortion as memoryless codecs.

To efficiently communicate with the above source, we proposed a joint source-channel system based on incremental redundancy that works under strict fixed delay and error-prone channel constrains. This system comprised of DPCM as a predictive coding source encoder and a rate-compatible punctured convolutional (RCPC) as error control codec. Unlike current retransmission based system our proposed model designed to work with constant bit-rate so as to maintain the synchronous nature of pure FEC systems. We present the notion of codetypes, which allows developing a model based on Markov Decision Process. Based on this, we design an algorithm based on dynamic programming to solve the optimization problem of source and channel rate allocation. The optimal solution leads to minimized overall distortion. In our simulation results, we compared the feedback based on no feedback approach. Our results confirmed the gain of the feedback-based scheme over pure

FEC.

Chapter 6

Future Work

In addition to the proposed system that is discussed in this study, other possible extensions of our joint-source channel rate allocation approaches are listed in the following:

- **Source encoder with transform-based coding:**

  In chapter 3 DPCM source encoder has been studied. The joint source-channel rate assignment approach we have explored so far is based on the assumption that multimedia source is encoded by the predictive coding system. Our optimization problem considers the previously coded frames to be used as an error concealment. Therefore, based on this scheme the optimization problem depends on the previous samples. One may propose DCT-based source coding such as H.265 for video coding. In this approach video frames are ”intra-coded”. Therefore, the optimization problem can be accomplished by de-correlating the encoding rate and distortion of each video frame from that of other frames in the video sequence.

- **Solving the optimization for the other error control scheme:**

  In chapter 4 we studied the problem of multimedia transmission over a noisy channel, and utilized a Hybrid ARQ error control scheme for channel error resilience. Nevertheless, other error control schemes, such as pure ARQ, can be used in the system, and in is possible that our rate control approach can be extended to incorporate other

rate control schemes.

- **Extending the number of codetypes:**

  We have proposed an algorithm 1 to solve the optimization problem of assigning proper rates to source and channel codes in each codetype, so as to the overall distortion be minimized. In our simulation we only considered a maximum number of codetypes to be 2. One may extend this number to be more than two codetypes and ultimately solve the Dynamic Programming to address the infinite horizon problem. The search for the optimal solution for such a system will become extremely complicated. It is because of the dependence that introduced between each codetype and the future codetypes.

- **Packet Length selection:**

  In the algorithm developed in the simulation of chapter 4, we assumed fixed frame size to solve the optimization problem. We have seen that the performance of our scheme depends on the frame size. A systematic frame size selector to minimize the distortion of a given application system, in itself, merits further study.

Bibliography

[1] P. Cosman, A. Kwasinki, and V. Chande. *Joint Source-Channel Coding*. Wiley, 2017. in preparation.

[2] Philip A. Chou and Zhourong Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 8(2):390–404, 2006.

[3] Masoud Khansari. Performance of predictive coders over noisy channels with feedback. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3473–3476. IEEE, 1998.

[4] Andres Kwasinski, Vinay Chande, and Nariman Farvardin. Delay-constrained joint source-channel coding using incremental redundancy with feedback. In *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, pages 283–286. IEEE, 2003.

[5] Zhihai He, Jianfei Cai, and Chang Wen Chen. Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(6):511–523, Jun 2002.

[6] W.A. Pearlman and A. Said. *Digital Signal Compression: Principles and Practice*. Cambridge University Press, 2011.

[7] R. Totty and G. Clark, Jr. Reconstruction error in waveform transmission (corresp.). *IEEE Trans. Inf. Theor.*, 13(2):336–338, September 2006.

[8] C.C. Cutler. Differential quantization of communication signals, July 29 1952. US Patent 2,605,361.

[9] J. Hagenauer. Rate-compatible punctured convolutional codes (rcpc codes) and their applications. *Communications, IEEE Transactions on*, 36(4):389–400, Apr 1988.

[10] Chen Gong and Xiaodong Wang. Adaptive transmission for delay-constrained wireless video. *Wireless Communications, IEEE Transactions on*, 13(1):49–61, January 2014.

[11] Ke-yen Chang and Robert W Donaldson. Analysis, optimization, and sensitivity study of differential pcm systems operating on noisy communication channels. *Communications, IEEE Transactions on*, 20(3):338–350, 1972.

[12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[13] Jin Lu, A. Nosratinia, and B. Aazhang. Progressive source-channel coding of images over bursty error channels. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 127–131 vol.2, Oct 1998.

[14] Fan Zhai, Y. Eisenberg, T.N. Pappas, R. Berry, and A.K. Katsaggelos. Rate-distortion optimized hybrid error control for real-time packetized video transmission. *Image Processing, IEEE Transactions on*, 15(1):40–53, Jan 2006.

[15] F. Hartanto and H.R. Sirisena. Hybrid error control mechanism for video transmission in the wireless ip networks. In *Local and Metropolitan Area Networks, 1999. Selected Papers. 10th IEEE Workshop on*, pages 126–132, 1999.

[16] P.A. Chou, A.E. Mohr, A. Wang, and S. Mehrotra. Error control for receiver-driven layered multicast of audio and video. *Multimedia, IEEE Transactions on*, 3(1):108–122, Mar 2001.

[17] J. Rosenberg and H. Schulzrinne. An rtp payload format for generic forward error correction. *RFC*, 1999.

[18] Guijin Wang, Qian Zhang, and Wenwu Zhu. Channel-adaptive error protection for scalable video over channels with bit errors and packet erasures. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 2, pages II–712–II–715 vol.2, 2002.

[19] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source - channel coding. *Information Theory, IEEE Transactions on*, 33(6):827–838, Nov 1987.

[20] G.M. Davis, J.M. Danskin, and Xiyong Song. Joint source and channel coding for internet image transmission. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 21–24 vol.1, Sep 1996.

[21] T. Stockhammer. Progressive video transmission for packet lossy channels exploiting feedback and unequal erasure protection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–169–II–172 vol.2, 2002.

[22] M. Gallant and F. Kossentini. Rate-distortion optimized layered coding with unequal error protection for robust internet video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(3):357–372, Mar 2001.

[23] Qian Zhang, Wenwu Zhu, and Ya-Qin Zhang. Network-adaptive scalable video streaming over 3g wireless network. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 579–582 vol.3, 2001.

[24] Rui Zhang, S.L. Regunathan, and K. Rose. End-to-end distortion estimation for rd-based robust delivery of pre-compressed video. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, volume 1, pages 210–214 vol.1, Nov 2001.

[25] S. Appadwedula, D.L. Jones, K. Ramchandran, and Leiming Qian. Joint source channel matching for a wireless image transmission. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 137–141 vol.2, Oct 1998.

[26] Joohee Kim, R.M. Mersereau, and Y. Altunbasak. Error-resilient image and video transmission over the internet using unequal error protection. *Image Processing, IEEE Transactions on*, 12(2):121–131, Feb 2003.

[27] L.P. Kondi, F. Ishtiaq, and A.K. Katsaggelos. Joint source-channel coding for motion-compensated dct-based snr scalable video. *Image Processing, IEEE Transactions on*, 11(9):1043–1052, Sep 2002.

[28] Dapeng Wu, Yiwei Thomas Hou, and Ya-Qin Zhang. Transporting real-time video over the internet: challenges and approaches. *Proceedings of the IEEE*, 88(12):1855–1877, Dec 2000.

[29] Dapeng Wu, Yiwei Thomas Hou, Bo Li, Wenwu Zhu, Ya-Qin Zhang, and H.J. Chao. An end-to-end approach for optimal mode selection in internet video communication: theory and application. *Selected Areas in Communications, IEEE Journal on*, 18(6):977–995, June 2000.

[30] Shengjie Zhao, Zixiang Xiong, and Xiaodong Wang. Joint error control and power allocation for video transmission over cdma networks with multiuser detection. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 5, pages 3212–3216 vol.5, 2002.

[31] A. Ortega. Optimal bit allocation under multiple rate constraints. In *Data Compression Conference, 1996. DCC '96. Proceedings*, pages 349–358, Mar 1996.

[32] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.