University of California
Santa Barbara

Signal Coding Approaches for Spatial Audio and Unreliable Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Sina Zamani

Committee in charge:

Professor Kenneth Rose, Chair
Professor Shivkumar Chandrasekaran
Professor Nikil Jayant
Professor Ramtin Pedarsani

March 2019

The Dissertation of Sina Zamani is approved.

Professor Shivkumar Chandrasekaran

Professor Nikil Jayant

Professor Ramtin Pedarsani

Professor Kenneth Rose, Committee Chair

March 2019

Signal Coding Approaches for Spatial Audio and Unreliable Networks

by

Sina Zamani

To my beloved parents and sister

Acknowledgements

First and foremost, I would like to express my deep sense of gratitude to my advisor Professor Kenneth Rose for his peerless guidance, support and encouragement throughout my graduate studies. I have learned a great deal from him, especially on how to tackle challenging research problems, and how to present my research results. Working and interacting with him over the last four years was truly an invaluable learning experience.

I am very thankful to Dr Tejaswi Nanjundaswamy, the former post-doc of our research group. Our interesting discussions and his insightful comments have greatly assisted me in developing my research mindset.

I would like to express my gratitude to Professor Shiv Chandrasekaran, Professor Nikil Jayant and Professor Ramtin Pedarsani for serving on my dissertation committee and reviewing my work. My research has been partially funded by Mozilla Inc. and Google Inc. Thanks to these companies.

I am grateful to my friends here in Santa Barbara and elsewhere in the U.S., who have made my life during PhD much more fun and enjoyable. And I would like to dedicate this thesis to my beloved parents and my little sister, without whose unconditional love and support this work would not have been possible.

Curriculum Vitæ
Sina Zamani

Education

Mar 2019	Ph.D. in Electrical and Computer Engineering, University of California, Santa Barbara.
Jan 2015	MS in Electrical and Computer Engineering, University of California, Santa Barbara.
June 2013	B.S in Electrical Engineering, Sharif University of Technology.

Exprience

2015-2019	Graduate Research Assistant, University of California, Santa Barbara.
2013-2018	Teaching Assistant, University of California, Santa Barbara.
2017	Summer Audio Research Intern, Infocoding Labs, Santa Barbara, CA.

Publications

- S. Zamani, and K. Rose, ”Spatial Audio Coding Without Recourse to Background Signal Compression”, Proc. IEEE ICASSP, May. 2019..
- S. Zamani, and K. Rose, ”Spatial Audio Coding with Backward-Adaptive Singular Value Decomposition”, 145th AES Convention, Oct 2018.
- A. Elshafiy, T. Nanjundaswamy, S. Zamani, K. Rose, ”On Error Resilient Design of Predictive Scalable Coding Systems”, Proc. IEEE ICASSP, April. 2018.
- S. Zamani, T. Nanjundaswamy, and K. Rose, ”Frequency Domain Singular Value Decomposition for Efficient Spatial Audio Coding”, Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), Oct 2017.
- B.Vishwanath , T. Nanjundaswamy, S. Zamani, and K. Rose, ”Deterministic Annealing Based Design of Error Resilient Predictive Compression Systems”, Proc. IEEE ICASSP, Mar. 2017.
- S. Zamani, T. Nanjundaswamy, and K. Rose, ”Recursive End-To-End Distortion Estimation for Error-Resilient Adaptive Predictive Compression Systems”, Proc. IEEE Workshop on Statistical Signal Processing (SSP), Jun. 2016.
- S. Zamani, T. Nanjundaswamy, and K. Rose, ”Asymptotic Closed-Loop Design of Error Resilient Predictive Compression Systems”, Proc. IEEE ICASSP, Mar. 2016.

Abstract

Signal Coding Approaches for Spatial Audio and Unreliable Networks

by

Sina Zamani

This dissertation is divided into two parts. The first part is concerned with developing algorithms for the compression of emerging 3D audio format, while the second part investigates optimization techniques for error-resilient predictive compression systems design.

In the first part, advances in development of compression algorithms for higher order ambisonics (HOA) data is presented. HOA has proven to be the method of choice in virtual reality applications, given its capability in reproducing spatial audio and its rendering flexibility. Recent standardization for HOA compression adopted a framework wherein HOA data are decomposed into principal components that are then encoded by standard audio coding, i.e., frequency domain quantization and entropy coding to exploit psychoacoustic redundancy. A noted shortcoming of this approach is the occasional mismatch in principal components across blocks, and the resulting suboptimal transitions in the data fed to the audio coder. In this dissertation, we propose a framework where singular value decomposition (SVD) is performed after transformation to the frequency domain via the modified discrete cosine transform (MDCT). This framework not only ensures smooth transition across blocks, but also enables frequency dependent SVD for better energy compaction. Moreover, we introduce a novel noise substitution technique to compensate for suppressed ambient energy in discarded higher order ambisonics channels, which significantly enhances the perceptual quality of the reconstructed HOA signal. In the next step, to reduce the burden of side information, a new encoding architecture is presented, where transform matrices are estimated backward-adaptively. This framework allows a more frequent usage of optimal SVD, thereby approaching the full potential of frequency domain SVD. Also the division of HOA data into predominant and ambient components in current schemes, is difficult to perceptually optimize and ignores spatial inter channel masking effects. To address this issues, a new encoding framework for compression of HOA data is presented, where a null-space basis vector extension technique enables all compression to be performed in the SVD domain, and a jointly computed common masking threshold accounts for effects of spatial masking across components.

The second part is concerned with developing optimization techniques for error-resilient predictive compression systems design. Prediction is used in virtually all compression systems and when such a compressed signal is transmitted over unreliable networks, packet losses can lead to significant error propagation through the prediction loop. Despite this, the conventional design technique completely ignores the effect of packet losses, and estimates the prediction parameters to minimize the mean squared prediction error, and optimizes the quantizer to minimize the reconstruction error at the encoder. While some design techniques have been proposed to accurately estimate and minimize the end-to-end distortion (EED) at the decoder that accounts for packet losses, they operate in a closed-loop, which introduces a mismatch between statistics used for design and statistics used in operation, causing a negative impact on convergence and stability of the design procedure. The first contribution of the dissertation is this part is proposing an effective technique for designing a compression system with a first order linear predictor, that accounts for the instability caused by error propagation due to packet losses, and enjoys stable statistics during design by employing open-loop iterations that on convergence mimic closed loop operation.

End-to-end distortion (EED) estimation, accounting for error propagation and concealment at the decoder, has been originally developed for video coding, and enables optimal rate-distortion (RD) decisions at the encoder. However, this approach was limited to the video coders simple setting of a single tap constant coefficient temporal predictor. This thesis considerably generalized the framework to account for: i) high order prediction filters, and ii) filter adaptation to local signal statistics. We demonstrate how this EED estimate can be leveraged, by an encoder with short and long term linear prediction, to improve RD decisions and achieve major performance gains. The approach is further extended to estimate EED in speech coders. The error propagation problem is exacerbated in this case, as standard coders not only predict the signal from past frames, but also the parameters (in the line spectral frequency domain) employed for such prediction. Hence, the prediction loop propagates errors in the reconstructed signal as well as errors in the prediction parameters. A recursive algorithm is proposed to estimate, at the encoder, the overall EED, by the subterfuge of parallel tracking of decoder statistics for prediction parameters and signal reconstructions, in their respective domains, which are then combined to obtain the ultimate EED estimate.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Spatial Audio

Creating interactive and immersive experiences is currently an area of significant interest, with major investment in virtual and augmented reality covering all aspects of content acquisition, storage, transmission, and display/playback. Achieving a truly immersive experience requires new formats for multimedia content, and in particular three-dimensional (3D) 360-degree audio, to represent information in a 3D space. The higher order ambisonics (HOA) paradigm [1, 2, 3, 4, 5] is a surround sound recording and reproduction technique that captures information of a 3D sound-field in its transmission channels. The key benefit of HOA is its flexibility to enable playback with any speaker configuration ranging from headphones to complex surround sound systems, thus allowing for a diverse variety of approaches to create an immersive experience.

The first order ambisonics format (also known as B-Format), originally developed in the 70s [1], has directional information of a sound-field recorded (codified) in 4 channels, namely W, X, Y, Z as,

$$\begin{aligned} W(t) &= \sum\nolimits_i s_i(t)/\sqrt{2}, & X(t) &= \sum\nolimits_i s_i(t)\cos(\theta_i)\cos(\phi_i), \\ Y(t) &= \sum\nolimits_i s_i(t)\sin(\theta_i)\cos(\phi_i), & Z(t) &= \sum\nolimits_i s_i(t)\sin(\phi_i), \end{aligned} \tag{1.1}$$

where $s_i(t)$ is a sound signal source coming from direction $(\theta_i\ ,\ \phi_i)$ where θ denotes azimuth and ϕ denotes elevation. The W channel corresponds to the pressure of the sound-field at the origin, while X, Y, Z are proportional to its gradients. Thus, unlike the traditional multichannel audio (e.g. 5.1 or 7.1 surround), where each channel has information corresponding to a given loudspeaker, in ambisonics, channels carry the directional and physical information of an entire sound-field. Given a set of N speakers, the decodified signal for speaker i located at $(\theta_i,\ \phi_i)$ is given as,

$$d_i(t) = w_i W(t) + x_i X(t) + y_i Y(t) + z_i Z(t), \tag{1.2}$$

where the set of parameters (w_i, x_i, y_i, z_i) can be determined to optimally reconstruct the sound-field for physical or psychoacoustic accuracy [3]. Clearly, the ambisonics channels are completely independent of the loudspeaker layout chosen for decodifyng the sound-field. An ambisonic decodifier is therefore designed for a specific speaker layout, and an ambisonic codified sound-field can be reproduced with any loudspeaker layout by employing an appropriate decodifier.

Despite the solid theoretical foundation, ambisonics failed to gain commercial success at the time due to the limited size of usable listening area (the so-called sweet spot where the sound-field is accurately reproduced) and poor localization. In the 90s, the theory of higher order ambisonics (HOA) [2] extended the approach, by decomposing the sound-field into higher order spherical components, resulting in

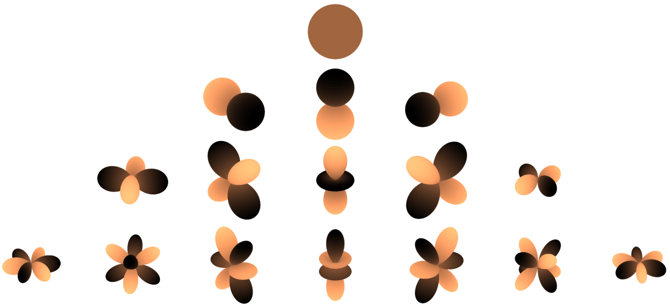


Figure 1.1: Spherical harmonics up to third order

improved localization and spatial resolution, and increased size of the sweet spot. The maximum order L at which the expansion is performed is called the HOA order. At each order, i, of the expansion, there are $2i+1$ channels and an overall HOA of order L has a total of $(L+1)^2$ channels. In this case the sound-field is expressed as,

$$s(t,\theta,\phi) = \sum\nolimits_{l=0}^{L} \sum\nolimits_{m=-l}^{l} B_{l,m}(t) Y_{l,m}(\theta,\phi), \tag{1.3}$$

where $B_{l,m}$ correspond to HOA channels (subsuming the traditional W, X, Y, Z of first order ambisonics) and $Y_{l,m}$ are the spherical harmonics (depicted in Figure 1.1 for $L=3$), which are the bases of the expansion. The codifying of the sound-field is done by projecting the sound sources onto these basis components, and the decodifying process reconstructs signals for arbitrarily located speakers in a manner similar to that of first order ambisonics.

1.1.1 Compression of HOA data

In practical applications, HOA data can include as many as 64 channels and given the enormous amount of data consumed by 3D audio, it is critical to achieve efficient compression for networking and transmission. The recent MPEG-H 3D audio standard [6] is the state-of-the-art for compression of HOA data. The encoder utilizes SVD to extract and encode distinct spatial audio objects, also refereed to as predominant or foreground sound components, which requires the SVD transform matrices to be encoded and sent to decoder as side information. The residual signal, not captured by the predominant components, is encoded in the ambisonics domain after it has been reduced in order, where the remaining ambisonic channels are called ambient or background components. Each foreground or background component is fed into a separate standard audio codec where it is independently encoded. Broadcast quality transmission and transparent quality transmission have been reported [7, 8] at bit-rates around 300 kbps and 500 kbps, respectively. However, one central concern with this framework, is the occasional mismatch between principal components across blocks, that could create abrupt transitions between adjacent frames. MPEG-H 3D employs an elaborate process of basis vector matching and interpolation to address this issue, however the transitions across frames remain sub optimal and degrade performance in terms of the achievable compression ratio and resulting perceptual quality. In **Chapter 2**, we demonstrate that considerable gains can be obtained by performing SVD in the frequency domain instead of on the original time sequence. This paradigm ensures smooth transitions between frames by leveraging the modified discrete cosine transform (MDCT) built-in overlap windows. Frequency domain SVD also enables SVD adaptation to frequency, but the increase in side information, to specify additional basis vectors, compromises the gains. **Chapter 3**, overcomes this shortcoming by introducing backward adaptive estimation of SVD basis vectors, at no cost in side information, thereby approaching the full potential of frequency domain SVD. **Chapter 4** is motivated by these observations: i) separate coding of SVD components ignores spatial inter channel masking effects; ii) compression in both SVD and ambisonic domains is difficult to perceptually optimize; iii) Only few predominant components are encoded due to the prohibitive side information cost of specifying SVD basis vectors. A novel coding architecture is presented to overcomes the first two concerns by performing all compression in the SVD domain with a masking threshold that is calculated jointly for all encoded components, thereby accounting for cross-component masking. The third shortcoming is circumvented by a novel method for extending a given set of SVD basis vectors at no side information cost, by computing (at both encoder and decoder) basis vectors to span the null space of the transmitted basis vectors

1.2 Error Resilient Predictive Compression System Design

Virtually all multimedia content (e.g., speech, audio, image, and video) consists of sources with memory, often exhibiting dominant temporal correlations. A subset of these signals is (locally) quasi-periodic and characterized by naturally occurring repetitive patterns, for example, voiced speech, mono and polyphonic music, and image textures. Other signals are not periodic but are nevertheless highly correlated temporally, such as video signals. Thus, exploiting temporal correlations is a critical component of all compression and communication systems. One central approach to do so involves prediction, typically a combination of linear short term and/or long term prediction filters, which are often employed in conjunction with transform, quantization, and entropy coding. An extensive literature exists, covering such coding techniques, including for speech [9], audio [10], image [11], and video [12] signals.

The theory underlying optimal prediction is well understood in its own right [13, 14], where the optimality criterion is typically specified in terms of the prediction error. Our work is motivated by the observation that, despite the importance and prevalence of prediction in real world compression and communication systems (see, e.g., [15, 16, 17, 18]), it has largely been studied in isolation, and its design falls short of accounting for the environment in which it is deployed, including its interaction with and impact on other modules, resulting in considerable unrealized performance gains. A major shortcoming of currently employed techniques is the mismatch in the cost function used for estimation and design of prediction parameters to that of the overall system. Most methods simply estimate parameters to minimize the mean squared prediction error, however this does not guarantee maximal reduction in the overall rate-distortion (RD) cost (for asymptotic RD theory see [19, 20]) of the system, nor does it account for other important aspects, such as the exacerbated impact of unreliable communication over networks, due to error propagation through the prediction loop. **Chapter 5** discusses an effective design technique for a first order predictive system to account for and overcome major stumbling blocks due to (i) the destabilizing effects of the prediction feedback loop during design, and (ii) the effects of error propagation when operating under unreliable network conditions. The proposed design scheme accurately estimate and minimize the end-to-end distortion (EED) at the decoder that accounts for packet losses. **Chapter 6** extends the end-to-end distortion estimation framework to systems employing adaptive higher order predictors by separately tracking statistics of the employed prediction parameters and the reconstructions at the decoder. We show incorporating the estimate obtained by the proposed approach in an RD framework to decide the number and location of prediction resets to achieve the right balance between compression and addition of redundancy to combat packet losses, results in significant performance improvements. The focus of **Chapter 7** is on estimating EED in speech coding and networking systems, where a combination of short term and long term prediction filters adapted to local signal statistics are employed. The error propagation problem is exacerbated in this case, as standard coders not only predict the signal from past frames, but also the parameters (in the line spectral frequency domain) employed for such prediction. Hence, the prediction loop propagates errors in the reconstructed signal as well as errors in the prediction parameters. A recursive algorithm to estimate, at the encoder, the overall EED, by the subterfuge of parallel tracking of decoder statistics for prediction parameters and signal reconstructions, in their respective domains, which are then combined to obtain the ultimate EED estimate. Experimental results provide evidence for substantial objective and subjective gains.

Part I

Towards Optimal Coding and Networking of Immersive 3D Audio

Chapter 2

Frequency Domain Singular Value Decomposition for Efficient Spatial Audio Coding

2.1 Introduction

Recently, a new spatial audio coding standard, MPEG-H 3D Audio [6], has emerged. The HOA input is decomposed into predominant sound elements and ambient background components, using standard singular value decomposition (SVD), and each of these are coded separately via an AAC based coder, where quantization and entropy coding are performed in the frequency domain to exploit psychoacoustic redundancies. While good broadcast quality has been reported for bit-rates around 300 kbps [8], the premise of this chapter is that higher compression efficiency and better perceptual quality can be achieved by employing SVD in the frequency domain. In the MPEG-H approach, there is often a mismatch of principal components across blocks, both in terms of order of components and their respective basis vectors.

MPEG-H employs an elaborate matching technique in combination with an overlap-add technique to mitigate this shortcoming. However, transitions between blocks remain suboptimal and introduce inefficiencies in the core codec and degrade the perceptual quality. Our approach completely eliminates this issue as we first transform to frequency domain via MDCT which ensures smooth transition across blocks with its built-in overlap. Moreover, optimal SVD can now be adapted to different frequencies, instead of a compromise decomposition for the entire spectrum. Finally, we employ noise substitution in a novel way to compensate for ambient energy loss and further improve perceptual quality of the rendered HOA data.

2.2 MPEG-H approach for compression of HOA data

The MPEG HOA encoder [21] processes the input HOA data over frames of length $2L$ ($L = 1024$) with 50% overlap. Let the number of HOA channels be $M = (N+1)^2$, where N is the ambisonics order. For current frame f, the encoder operates on HOA data X_f, which is an $2L \times M$ matrix, and performs standard singular value decomposition (SVD),

$$X_f = U_f \Sigma_f V_f^T, \tag{2.1}$$

where U_f is an $2L \times 2L$ unitary matrix, Σ_f is a $2L \times M$ rectangular diagonal matrix with non-zero elements on the diagonal and V_f is an $M \times M$ unitary matrix. Each of the N vectors in U_f (of length $2L$ samples) can be interpreted as representing normalized separated audio signals that have been decoupled from any directional information, and Σ_f stores the energy of these sound components. The spatial characteristics are captured by individual columns of V_f, or basically the basis vectors of the SVD transform. The SVD construction ensures that predominant components, corresponding to the largest r singular values have as basis vectors the first r columns of V_f. Let V_f be truncated to the first r columns, and further be independently or differentially quantized to $\hat{V}_f$ and sent to the decoder as side information for each frame, so as to enable it to transform back the predominant components to the ambisonics domain. To keep encoder and decoder in sync, the quantized $\hat{V}_f$ is used to generate the predominant components $\tilde{Y}_f$ (now an approximation of first r columns of $U_f \Sigma_f$), as,

$$\tilde{Y}_f = X_f \hat{V}_f (\hat{V}_f^T \hat{V}_f)^{-1}. \tag{2.2}$$

Note that the inverse term is for renormalization of the quantized basis vectors (to maintain unitarity). The next step is to code the predominant or foreground components, each corresponding to a column of $\tilde{Y}_f$, using separate instances of the core audio codec. This requires concatenating components across frames. However, since SVD arranges the basis vectors based on the singular value magnitudes, the same foreground component might change position in $\tilde{Y}$ from frame to frame depending on the magnitude of its singular value relative to others. This can result in noticeable blocking artifacts, if blindly concatenated foreground components are fed to the core codec, and severe discontinuities across consecutive frames can reduce the achievable compression ratio and introduce significant artifacts in the HOA reconstructions. While there are several approaches to reorder and match components with the previous frame, we employ the magnitude of correlation between column vectors of $\hat{V}_f$ and $\hat{V}_{f-1}$ as the criterion in an Hungarian matching algorithm [22], which we found to be effective.

Even with matched components, simple concatenation across frames would introduce noticeable artifacts as a small change in the basis vector causes some mismatch

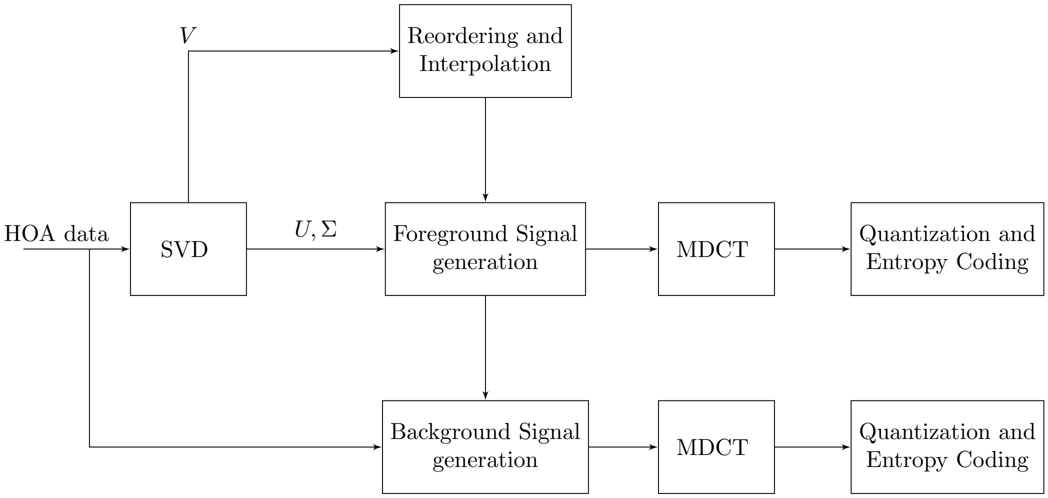

V
Reordering and Interpolation
HOA data
SVD
U, Σ
Foreground Signal generation
MDCT
Quantization and Entropy Coding
Background Signal generation
MDCT
Quantization and Entropy Coding

Figure 2.1: Overview of MPEG-H encoder

at the frame boundary. Hence the encoder interpolates the column vectors of $\hat{V}$ between current frame and previous frame to ensure continuity over time. Specifically, a different transform matrix is used for each sample of the current frame, whose column vectors are obtained as,

$$\bar{v}_f^i(l) = (1 - w(l))\hat{v}_{f-1}^i + w(l)\hat{v}_f^i, \tag{2.3}$$

where $\hat{v}_f^i, \hat{v}_{f-1}^i$ are the ith matched column vectors of $\hat{V}$ for current and previous frames, $\bar{v}_f^i(l)$ is ith column vector for sample l in current frame and $w(l)$ is a window function, which may be the triangular or Hanning window. The interpolation should also account for the fact that the vectors might get negated from one frame to next frame by performing a sign correction when needed.

An approximation of the HOA data, $\tilde{X}_f$, is generated by transforming the foreground components back to the ambisonics domain, which is then subtracted from the original data to produce the ambient (or background) HOA data. The foreground components are coded using separate instances of the core audio codec. The order of background HOA data is then reduced (from N to some t) and this lower order HOA data are also coded using the core audio codec. An illustration of the MPEG-H approach is shown in Figure 2.1.

2.3 Frequency Domain SVD for HOA data compression

Clearly, the MPEG-H approach performs an elaborate process of matching and interpolating transform basis vectors of consecutive frames to improve their continuity over time and to mitigate the artifacts stemming from blockwise SVD application. We propose to circumvent this underlying and fundamental shortcoming with a framework wherein SVD is employed *after* transformation to frequency domain via MDCT, which naturally achieves the required smoothness with its built-in overlap. Moreover, this framework enables the significant flexibility to make both the SVD and the number of components to be retained, adaptive to frequency, instead of using a compromise for the varying needs of different frequency bands.

In the proposed approach, the HOA data are processed in the encoder after segmenting each HOA channel into 50% overlapped frames of length $2L$. The samples of each channel are separately transformed via MDCT after windowing to obtain the transformed data for the current frame, S_f, which is an $L \times M$ matrix. S_f is now divided into different frequency bands, $S_f^T = [S_{f_1}^T S_{f_2}^T ... \, S_{f_n}^T]$ where n is the number of frequency bands with lengths $l_1, l_2, ..., l_n$ and $\sum_i l_i = L$. For each frequency band, a different SVD is obtained, $S_{f_i} = U_{f_i} \Sigma_{f_i} V_{f_i}^T$, the top r_i components are retained (which may vary over bands), and the correspondingly truncated V_{f_i} are coded to $\hat{V}_{f_i}$ and sent to the decoder as side information. Similar to (2.2), predominant components

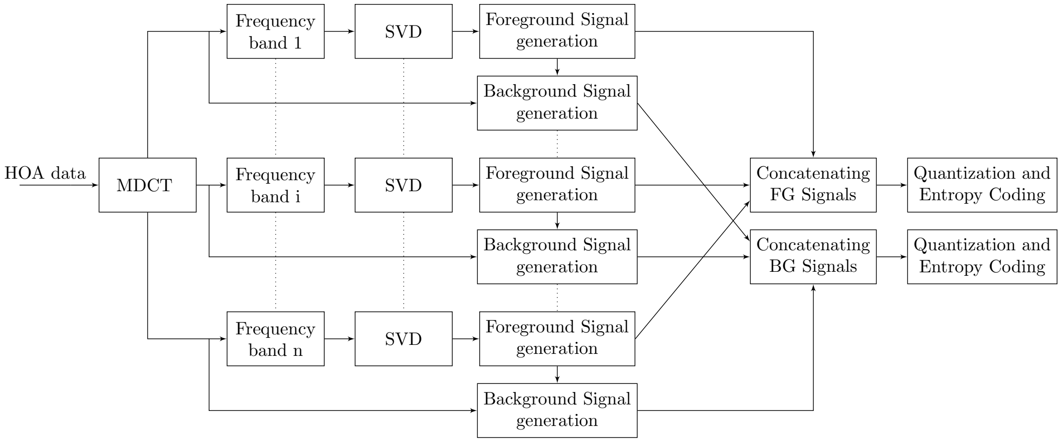

HOA data
MDCT
Frequency band 1
Frequency band i
Frequency band n
SVD
SVD
SVD
Foreground Signal generation
Background Signal generation
Foreground Signal generation
Background Signal generation
Foreground Signal generation
Background Signal generation
Concatenating FG Signals
Concatenating BG Signals
Quantization and Entropy Coding
Quantization and Entropy Coding

Figure 2.2: Overview of proposed method encoder

for each band are obtained as $\tilde{Y}_{f_i} = S_{f_i}\hat{V}_{f_i}(\hat{V}_{f_i}^T\hat{V}_{f_i})^{-1}$, and are concatenated to generate the foreground data for the entire frame, $\tilde{Y}_f^T = [\tilde{Y}_{f_1}^T\tilde{Y}_{f_2}^T... \tilde{Y}_{f_n}^T]$. Predominant components are mapped back to ambisonics domain to provide an approximation of HOA data in spectral domain, $\tilde{S}_f^T = [\tilde{S}_{f_1}^T\tilde{S}_{f_2}^T... \tilde{S}_{f_n}^T]$, where $\tilde{S}_{f_i} = \tilde{Y}_{f_i}\hat{V}_{f_i}^T$, and $\tilde{S}_f$ is subtracted from S_f to produce the background components. The predominant and ambient sound components are fed to different instances of core audio codec's quantization and entropy coding modules. An illustration of the proposed approach is shown in Figure 2.2.

2.3.1 Side Information Compression

To exploit the temporal correlations between transform matrices of consecutive frames, the Hungarian algorithm [22] is employed to match the column vectors of V_{f_i} matrices of consecutive frames corresponding to ith frequency band based on correlation coefficients. We used a scalar prediction coefficient (equal to correlation coefficient) for each vector. We selected approximately 10,000 frames from third order ambisonics files as training set to design a quantizer for prediction coefficients and prediction residuals using Generalized Lloyd Algorithm (GLA).

2.3.2 Perceptual noise substitution

Discarding higher orders of background data significantly suppresses the ambient sound of the ultimate HOA reconstruction and degrades the perceptual quality of the rendered data. To mitigate this issue, we introduced a novel noise substitution technique that replaces the content of the discarded channels with noise designed to be perceptually relevant. Specifically, for each of the 49 critical frequency groups defined in the MPEG standard, the spectral flatness is calculated for each discarded channel as,

$$\text{Flatness}_f^{ij} = \frac{exp(\frac{1}{|B_f^{ij}|}\sum_{k\in B_f^{ij}} \ln B_f^{ij}[k])}{\frac{1}{|B_f^{ij}|}\sum_{k\in B_f^{ij}} B_f^{ij}[k]}, \tag{2.4}$$

where B_f^{ij} are the power spectrum coefficients for channel i and frequency group j of the current frame background data. For each frequency group, these flatness values are averaged over all channels, and used as a measure of how "noise-like" the content of that frequency group is. If the average flatness is higher than a threshold, then the average energy is calculated for that frequency group across all channels and all its frequency bins. The decoder generates perceptual noise at the specified energy for all channels of the frequency group. Figure 2.3 shows an illustration of how background data is encoded, where the first order data is encoded using the core audio codec, and for each frequency group of higher order data, the average spectral flatness is compared to a threshold, and thus a maximum of 49 energy values are encoded (similar to scale factors) and sent to decoder as side information.

It is important not to confuse the perceptual noise substitution technique presented in this section with the well known noise filling and Intelligent Gap Filling

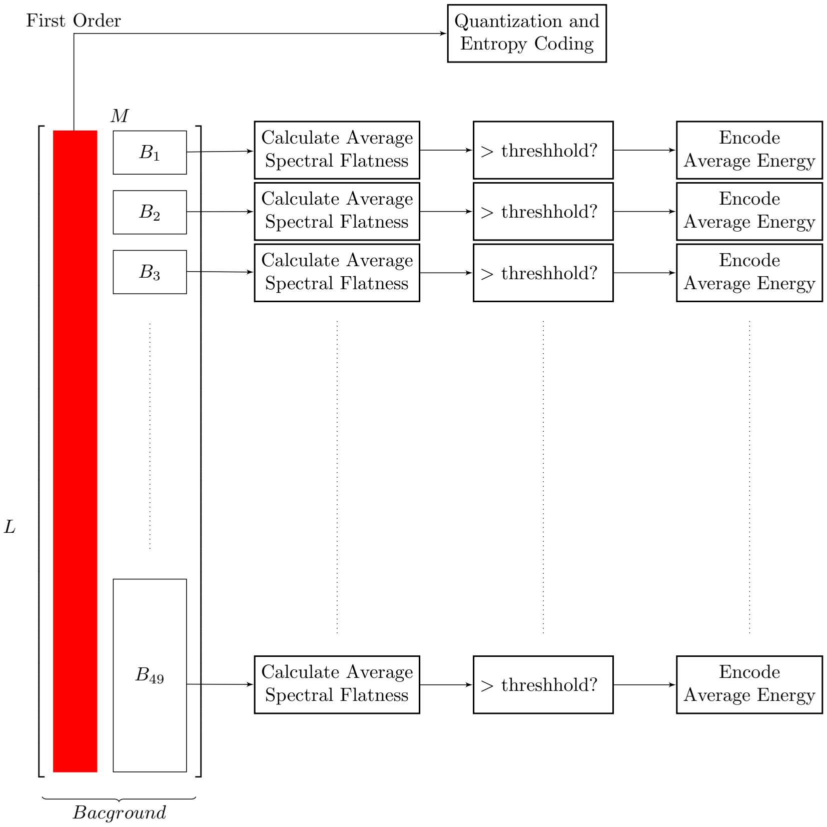


Figure 2.3: Perceptual Noise Substitution overview

(IGF) [23, 24] tools, also available in the MPEG-H core audio codec. The proposed noise substitution is performed in the ambisonics domain, to conceal suppressed energy in discarded ambisonics channels. In contradistinction, these MPEG-H tools are applied to PCM signals fed to the core audio codec, i.e., when the foreground and background components are encoded. We re-emphasize that the suppression of ambient sound in final rendered data is due to discarding higher order channels of the background data, and this information will not be retrieved by simple application of available MPEG-H 3D tools to predominant and ambient components as they are encoded by the core audio codec.

2.4 Experimental Results

To validate the efficacy of the proposed approach we conducted objective and subjective experiments. The experiment was on a dataset of recordings provided by Google, which consists of 6 third order ambisonics files. As the software for MPEG-H encoder is not yet publicly available, we implemented our own representative version of it, as described in Section 2.2, based on the published patents [25, 26] and the standard documentation [21] which serves as a baseline for comparison. Other than the explicit contributions of the new approach, the competitors are identical in terms of options enabled, etc. All side information is accounted for in the total bit-rate.

In all the experiments $r = r_i = 4, \forall i$ and $t = 1$, that is, the number of foreground and background channels are both set to 4 for all frames, which results in a total of 8 components being encoded with the core codec. In the proposed approach, we divided the frequency data into 4 uniformly sized bands and a different transform is obtained for each frequency band. While employing frequency dependent SVD always results in better compaction of energy, this does not always translate to improved RD performance for the fixed quantizers and entropy coders employed. We believe this limitation can be addressed by redesigning the quantizers and entropy coders for the new statistics. In order to obtain preliminary results we employed the “shortcut” of providing two encoding modes per frame, of using a single frequency band (mode $m_f = 0$), or using 4 frequency bands (mode $m_f = 1$), and selecting the one which minimizes the RD cost. When the mode switches between frames, the transform matrix (or matrices) of current frame are predicted from the best available previous transform matrix (or matrices).

	Bit-rate reduction at various operating points		
Sequence	∼308 kbps	∼375 kbps	∼500 kbps
2src	7.83%	8.03%	8.37%
A Round	-0.8%	0%	1.41%
doll_intro	6.64%	6.93%	7.26%
heli_fount	4.58%	5.96%	7.02%
lyon	3.98%	4.72%	3.58%
Murmur2	6.50%	8.89%	10.9%
Average	4.79%	5.75%	6.44%

Table 2.1: Proposed framework's reduction in bit-rate

2.4.1 Objective Results

Note that perceptual distortion optimization for foreground data obtained through SVD, especially in comparison to background data in ambisonics domain, is still an open problem. To obtain preliminary objective results, we simply encoded both the competing methods to minimize the bit-rates for a given maximum quantization noise to mask ratio (MNMR) constraint for all bands of all channels. Investigation of the true objective perceptual distortion measure and its corresponding optimization approach is part of future work. Percentage reduction in bit-rate for the proposed method in comparison to the MPEG-H approach, obtained at different operating points is presented in Table 2.1. Clearly, there is a consistent improvement in performance for the proposed framework. Table 2.2 also presents the contribution of foreground and background data to the total bit-rate for each file (averaged over the three operating points) for the two encoding methods. Clearly, the improved energy compaction of the proposed approach results in significant reduction in bit-rate required for background, while marginally increasing the foreground bit-rate.

	Foreground contribution		Background contribution	
Sequence	Proposed	MPEG	Proposed	MPEG
2src	67.78%	59.52%	25.06%	37.09%
A Round	56.37%	53.12%	37.27%	44.85%
doll_intro	63.93%	57.24%	26.99%	38.73%
heli_fount	54.6%	51.33%	41.76%	47.29%
lyon	59.65%	55.94%	27.65%	39.34%
Murmur2	63.08%	60.53%	26.47%	37.25%
Average	60.90%	56.26%	30.87%	40.76%

Table 2.2: Contribution of foreground and background data to total bit-rate for the two methods

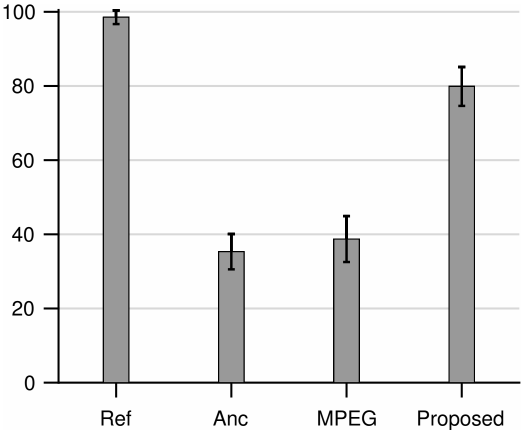


Figure 2.4: MUSHRA listening test results comparing the encoding techniques

2.4.2 Subjective Results

We conducted subjective evaluations to determine the true perceptual gains using the MUSHRA listening tests [27]. This is particularly important given the above reservations about the ability of the objective measure to fully capture the perceptual quality. The test items were scored on a scale of 0 (bad) to 100 (excellent) and the tests were conducted with 8 listeners. We extracted 10s portions of each file for evaluation. The test files includes challenging scenes with speech, music and objects moving. A binaural renderer was deployed to convert the reconstructed HOA coefficients to stereo signals. The binaural renderer works with HRTFs for a set of loudspeakers around the head. The HOA data is decoded to the positions of those loudspeakers using Max r_E [4, 5] mode-matching or L2-norm decoding techniques, and the decoded signal at each loudspeaker is convolved with the associated HRTFs for the left and right ear, respectively. Finally, the convolved signals for each ear are added together to generate the stereo output. Randomly ordered 4 versions of each audio sample (including a hidden reference, a 3.5 kHz low-pass filtered anchor, the encoded file using the proposed method and the encoded file using the MPEG method) were presented to the listeners. For these tests, the bit-rates (around 375 kbps) were matched for each competing file. The subjective evaluation results, including the mean and 95% confidence intervals, as presented in Figure 2.4 clearly demonstrates the substantially improved quality. This margin of improvement could not have been predicted from the moderate gains observed in objective results, clearly highlighting the critical need for further research in developing an appropriate objective perceptual distortion measure and corresponding optimization approach.

2.5 Concluding Remarks

This chapter presents a new framework for compression of higher order ambisonics data by first transforming the coefficients to MDCT domain and then decomposing into principal components. Unlike the current approaches, which suffer from suboptimal transitions between frames, the proposed approach not only ensures smooth transitions, it also enables frequency dependent decomposition and selection of dominant components. Furthermore, a novel way of employing noise substitution is introduced to enhance the perceptual quality of final reconstructions. Objective and subjective results illustrate the effectiveness of the proposed approach with significant performance improvements.

Chapter 3

Spatial Audio Coding with Backward-Adaptive Singular Value Decomposition

3.1 Introduction

In the previous chapter, we proposed a framework for compression of HOA data, where singular value decomposition is moved from the time to the frequency domain, i.e., it is performed after transformation by MDCT. This framework not only ensures smooth transition across blocks by leveraging the MDCT built-in overlap windows, but also enables frequency dependent SVD, instead of what is effectively a compromise decomposition for the entire spectrum. It thus achieves better energy compaction and hence improved compression performance. Moreover, we introduced a novel noise substitution technique to compensate for suppressed ambient energy in discarded HOA channels, which significantly enhanced the perceptual quality of the reconstructed HOA signal.

In any compression method there is an inherent tradeoff between adaptivity and the cost it incurs in side information. While performing SVD in the frequency domain opens the door to employing the optimal SVD for each frequency group, tailored to its needs, this benefit comes at significant cost in side information, which might outweigh the gains due to specialized transforms. In this chapter we introduce a novel encoding architecture to circumvent the prohibitive side information cost by estimating the SVD transforms backward-adaptively. This framework introduces considerable performance gains in terms of the objective rate-distortion tradeoff, as well as significant enhancement of perceptual quality of rendered data, especially at lower bit-rates.

3.2 Proposed Backward-Adaptive SVD for Compression of HOA data

Performing SVD in frequency domain allows for frequency dependent decomposition, instead of using a compromise transform for the entire frame. This often results in better energy compaction in predominant components, however the burden of side information associated with sending more transforms per frame represents a major obstacle to realizing the potential gains. As a shortcut to achieve some gains, in previous chapter we introduced two encoding modes per frame, one uses a single frequency band and the other uses multiple (4) bands. The encoder switched between the modes per frame to minimize the rate. Here, we propose an alternative framework to significantly improve the RD performance at the cost of minimal side information, by estimating the transform matrices backward-adaptively, as described below.

Let R be a $m \times m$ square matrix. Then an eigenvector of R, denoted by e is a vector that is mapped to a scaled version of itself, i.e. $Re = \lambda e$, and λ is the corresponding eigenvalue. If R is symmetric and positive definite, then eigenvalues of R are real and positive. We can group the eigenvalues in a $m \times m$ diagonal matrix Λ , and the eigenvectors in a $m \times m$ matrix E, and then, it is straightforward to show $RE = E\Lambda$ or equivalently,

$$R = E\Lambda E^{-1}, \tag{3.1}$$

and the above decomposition is called the eigenvalue decomposition for R.

Recall from previous chapter that in the frequency domain SVD decomposition approach, the encoder blocks HOA data into 50% overlapped frames of length $2L$. HOA data is mapped to the frequency domain by applying MDCT separately to each channel of X_f. The resulting $L \times M$ matrix, denoted S_f, is divided into smaller frequency bands, i.e., $S_f^T = [S_{f_1}^T S_{f_2}^T ... \, S_{f_n}^T]$, where n is the number of frequency bands with lengths $l_1, l_2, ..., l_n$, where $\sum_i l_i = L$. For each band, a different SVD decomposition is obtained, $S_{f_i} = U_{f_i} \Sigma_{f_i} V_{f_i}^T$.

If we consider the singular value decomposition for current frame data in frequency domain, $S_f = U_f \Sigma_f V_f^T$, a special matrix of interest is the correlation matrix denoted as $R_f = S_f^T S_f$, which can be calculated as,

$$R_f = S_f^T S_f = V_f \Sigma_f U_f^T U_f \Sigma_f V_f^T = V_f \Sigma_f^2 V_f^T. \tag{3.2}$$

If we compare (3.2) to (3.1), we see that the transform matrix for current frame, V_f can be obtained by eigenvalue decomposition of R_f.

To minimize the cost of side information, the basis vectors for current frame can be obtained backward-adaptively, using the correlation matrix of previous frame R_{f-1}. And the the correlation matrix for current frame can be updated backward-adaptively

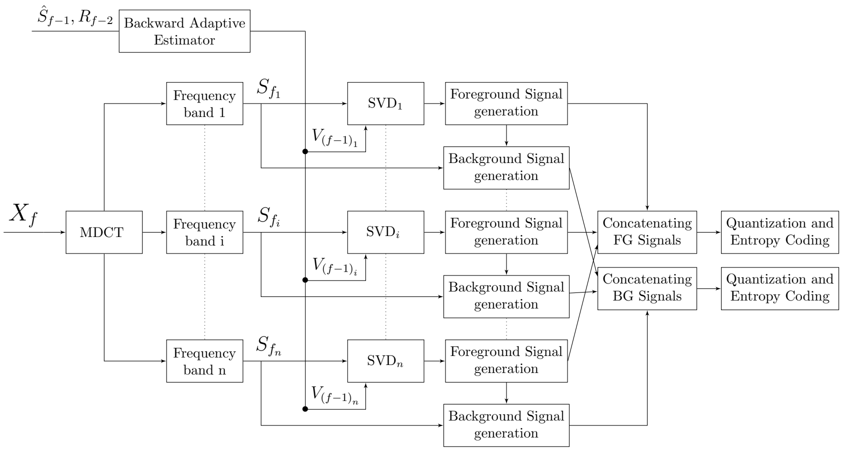

$\hat{S}_{f-1}, R_{f-2}$
Backward Adaptive Estimator
Frequency band 1
S_{f_1}
SVD$_1$
Foreground Signal generation
$V_{(f-1)_1}$
Background Signal generation
X_f
MDCT
Frequency band i
S_{f_i}
SVD$_i$
Foreground Signal generation
Concatenating FG Signals
Quantization and Entropy Coding
$V_{(f-1)_i}$
Background Signal generation
Concatenating BG Signals
Quantization and Entropy Coding
Frequency band n
S_{f_n}
SVD$_n$
Foreground Signal generation
$V_{(f-1)_n}$
Background Signal generation

Figure 3.1: Encoder architecture

at both decoder and encoder as,

$$R_f = w_1 \hat{S}_f^T \hat{S}_f + \sum_{j=1}^{l-1} w_{j+1} R_{f-j}, \tag{3.3}$$

where $w = [w_1 w_2 ... w_l]$ is a weight vector for a "leaky" weighted sum over a window of length l, where $\sum_{j=1}^{l} w_j = 1$, and $\hat{S}_f$ is the HOA reconstruction in frequency domain of the current frame. Similarly, the correlation matrix for each frequency band, R_{f_i}, can be updated as,

$$R_{f_i} = w_1 \hat{S}_{f_i}^T \hat{S}_{f_i} + \sum_{j=1}^{l-1} w_{j+1} R_{f_i - j}. \tag{3.4}$$

Figure 3.1 and Figure 3.2 illustrate an overview of the proposed encoder. Note that in the proposed encoding architecture, no side information is sent to decoder, as both decoder and encoder update correlation matrices backward-adaptively. In other words, as shown Figure 3.1, the transform matrix obtained in the current frame, using the reconstruction of the current frame and the previous frame correlation matrix, will be applied to the data of the next frame. Once the transform matrix is obtained, predominant and ambiant components for different frequency bands are generated similar

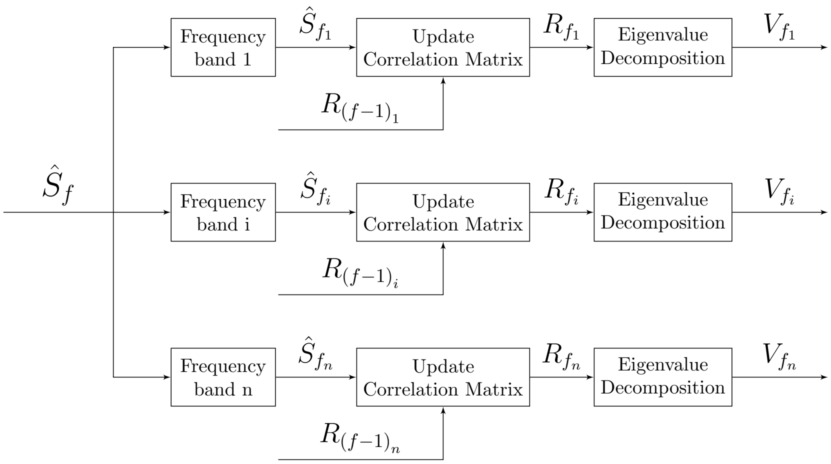


Figure 3.2: Backward-adaptive transform estimator

to the framework discussed in the previous chapter. The foreground and background components are concatenated across frequency bands and are independently coded using different instances of the core audio codec.

3.3 Experimental Results

In this section we present the results of subjective and objective evaluations that were conducted to compare three competing HOA data coders:

1. CMPEG: Our implementation of the MPEG standard, discussed in detail in the previous chapter.

2. CFSVD: The architecture we discussed in the previous chapter, where the maximum number of frequency bands for SVD adaptation is set to 4, and two encoding modes are provided per frame, one using a single frequency band and the other using multiple frequency bands. This coder employs the perceptual noise substitution technique discussed in 2.3.2.

3. CBW: The approach proposed in this chapter. The perceptual noise substitution framework is enabled for this coder as well. The number of frequency bands were set to 8 and $w = [0.1\ 0.9]$.

In all competing coders, the number of foreground and background components were set to 4, and a total of 8 components were encoded using the core audio codec. Note that the core audio encoders we used are standard compatible but not conventional, and achieve better optimization via a trellis approach to select encoding parameters (scale factors and Huffman codebooks), as described in [28]. The experiments were conducted with 6 (third order, 16 channels) HOA files provided by Google for UCSB research, which include speech, music, with static and moving objects.

3.3.1 Objective Results

A good distortion measure that accounts for human auditory perception for 3D audio, and an effective optimization framework to find encoding parameters to minimize such a distortion metric, are both still subjects of ongoing research. This difficulty is further exacerbated by the fact that the predominant sound components are encoded in the SVD domain, while the ambient data are encoded in the ambisonics domain, and it is not obvious how to properly account for masking effects and the contribution of quantization noise in each coded component to the final distortion. To obtain preliminary results, the core audio coders in all competing HOA codecs were run to minimize the maximum quantization noise to masking ratio (MNMR) criterion in all frequency bands for all encoded predominant and ambient channels. Percentage reduction in bit-rate for the CBW codec and CFSVD in comparison to the MPEG approach, obtained at different operating points, are presented in Table 3.2 and Table 3.1. It is clear that the proposed architecture consistently outperforms

	Bit-rate reduction of CFSVD over CMPEG at various operating points		
Sequence	∼175 kbps	∼240 kbps	∼375 kbps
2src_conv_office	7.05 %	7.66 %	8.26%
A Round Around-SpotMics	3.32 %	4.31 %	6.43 %
glass_lab_nr	6.42 %	7.07 %	7.93 %
helicopter_fountain	1.70 %	3.81 %	6.41 %
lyon	5.34 %	6.33 %	6.70 %
Murmur2	3.19 %	5.80 %	9.58 %
Average	4.50 %	5.83 %	7.56 %

Table 3.1: Bit rate reduction of the previous chapter approach, CFSVD, over CMPEG

	Bit-rate reduction of CBW over CMPEG at various operating points		
Sequence	∼175 kbps	∼240 kbps	∼375 kbps
2src_conv_office	11.67 %	13.16 %	13.35 %
A Round Around-SpotMics	13.92 %	14.31 %	16.45 %
glass_lab_nr	21.94 %	21.93 %	22.75 %
helicopter_fountain	3.57 %	4.97 %	11.71 %
lyon	27.04 %	26.97 %	26.87 %
Murmur2	11.30 %	14.87 %	16.38 %
Average	14.91 %	16.04 %	17.92%

Table 3.2: Bit rate reduction of the proposed approach in this chapter, CBW, over CMPEG

both CFSVD and CMPEG. In particular it achieves bit rate reduction of 16.3% on average, over CMPEG, which represents a major improvement over the reduction of 5.96% on average, over CMPEG offered by our previous approach CFSVD, where side information stood in the way of full exploitation of frequency-domain SVD. These results provide strong evidence for the benefits of backward-adaptive estimation of SVD basis vectors.

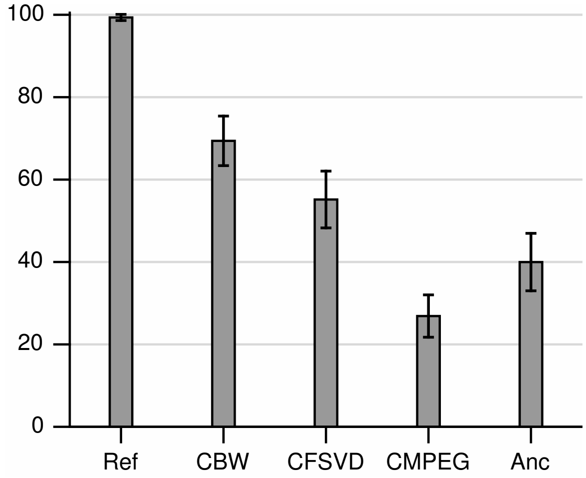


Figure 3.3: MUSHRA listening test results comparing the encoding techniques

3.3.2 Subjective Results

To measure the true perceptual audio quality gains we conducted MUSHRA listening tests [27]. The first 10 seconds of each file were selected for evaluation. The HOA data were converted to stereo signals using a binaural renderer. Ten listeners participated in the test and scored the listening files on a scale of 0 (bad) to 100 (excellent) based on the audio quality. Listeners were provided with 5 randomly ordered different versions of each binaurally rendered HOA file: the hidden reference (ref), a 3.5 kHz low passed anchor (anc), and files encoded with CMPEG, CFSVD and CBW. The bit-rates for competing files were matched at around 120 Kbps. The subjective evaluation results, including the mean and 95% confidence intervals, are depicted in Figure 3.3. The test provides a measure of subjective improvement and clearly indicates the substantially improved reconstruction quality achieved by the proposed architecture relative to both CFSVD and CMPEG.

3.4 Concluding Remarks

In this chapter we proposed a new encoding architecture for the compression of HOA data, where transform matrices are estimated backward-adaptively. This framework minimizes the side information cost, and thus allows a more frequent usage of optimal SVD that is adapted to the frequency group, instead of settling for a compromise transform for the entire frame. Considerable performance gains were demonstrated by both objective and subjective evaluations.

Chapter 4

Spatial Audio Coding Without Recourse to Background Signal Compression

4.1 Introduction

Several stumbling blocks stand in the way of optimal spatial audio coding. Psychoacoustic models and distortion measures that can accurately account for human perception of 3D audio, and an effective optimization framework to find the encoding parameters that minimize such a distortion metric, are both elusive and subjects of ongoing research. Moreover, existing approaches only encode a few predominant components, mainly due to the prohibitive cost in side information. Consequently, a significant portion of the main HOA data energy and directional information leaks back to the ambient data, and MPEG-H 3D reverts to encoding the first order background data to recapture some of this signal.

The difficulty of optimizing the encoding parameters is further exacerbated by the fact that the predominant sound components are encoded in the SVD domain, while the ambient data are encoded in the ambisonics domain, and it is not obvious how to properly account for masking effects and the contribution of quantization noise in each coded component to the final distortion. As a result, existing techniques default to the straightforward, though quite sub-optimal, route of encoding the predominant and ambient components independently. The main shortcomings of such approaches are that they completely neglect inter channel masking effects and fall short of realizing the full potential of SVD to decompose the HOA data into sound components thus eliminating spatial redundancies.

As a first "coarse" approach to achieve a proof of concept and demonstrate the potential benefits of circumventing the above shortcomings, this chapter proposes a novel encoding architecture where only predominant components are encoded. The premise is that the capability of SVD to extract and decorrelate distinct spatial audio objects, should be fully exploited and, moreover, the setting allows for better handling of perceptual masking effects. To show the potential gains from accounting for inter-channel masking effects, a first crude approach is proposed where the energy in a given frequency band, averaged across predominant components, is used to calculate the common quantization noise to masking threshold ratio for all encoded channels. The proposed framework, where all compression is performed in the SVD domain, offers increased adaptivity compared to existing techniques, as it encodes more predominant components. To reduce the prohibitive burden of side information, we propose a new paradigm that extend the set of predominant basis vectors with an approximate complementary set, at no side information cost.

4.2 Proposed Encoding Architecture

Compression often involves an inherent tradeoff between the benefits of adaptivity and its cost in side information. MPEG-H 3D's conversion of ambisonics data to the SVD domain opens the door to effective adaptation to spatial configurations. But, in practice, such adaptivity is severely restricted to very few predominant SVD components (typically 4), due to the prohibitive cost in side information to update the SVD basis vectors. In order to compensate for this limitation, MPEG-H 3D employs an ad hoc "fix", which consists of mapping the residual of the foreground procedure back to the HOA domain for background re-encoding to capture some of the loss. We propose to instead overhaul the framework such that it maximizes adaptivity by performing all compression in the SVD domain, but at no additional cost in side information. The subterfuge is to add basis vectors that span the null space of the predominant SVD components specified to the decoder. These additional basis vectors can be computed by encoder and decoder without side information.

Specifically, consider the set of r orthogonal vectors in the truncated version of the transform matrix for frame f and frequency band i, denoted by $V_{f_i,{\scriptscriptstyle SVD}}$. The goal is to find other vectors orthogonal to this set, i.e. $g \in \mathcal{R}^M$ such that $g^T V_{f_i,{\scriptscriptstyle SVD}} = 0$. In other words, we seek vectors spanning the null space of $V_{f_i,{\scriptscriptstyle SVD}}$. A trivial example in 3D is illustrated in Figure 4.1 where a principal vector (blue) is sent to decoder, which allows two additional vectors spanning the null space to be computed. The original data is projected along these vectors and the p vectors corresponding to highest energy signal components are selected to extend the set of predominant vectors obtained by SVD and are placed in a matrix denoted by $V_{f_i,{\scriptscriptstyle Null}}$. The only additional side information is an index specifying which p of the basis vectors were selected. Thus the effective transform matrix is obtained by concatenating $V_{f_i,{\scriptscriptstyle SVD}}$ and $V_{f_i,{\scriptscriptstyle Null}}$ as $V_{f_i} = [V_{f_i,{\scriptscriptstyle SVD}} V_{f_i,{\scriptscriptstyle Null}}]$.

Predominant components now can be obtained using V_{f_i} similar to Chapter 2.

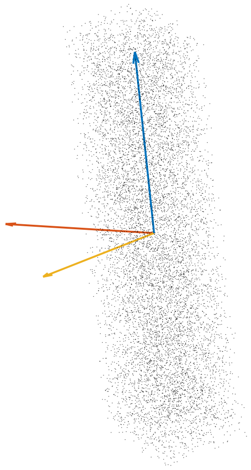


Figure 4.1: A simple 3D example: the blue vector is sent to decoder, and 2 complementary vectors spanning the null space are computed.

Without a good distortion measure that explicitly accounts for perceptual artifacts caused by 3D audio coding, current approaches encode all predominant and ambient sound components independently, thus neglecting inter-channel dependencies and masking effects. Leveraging the above framework where all compression is performed in the SVD domain, we propose a first "crude" framework to provide initial but strong evidence for the potential gains due to accounting for inter-channel masking effects. Specifically, we jointly calculate masking thresholds for all channels.

Let us consider the simple psychoacoustic model with a fixed signal-to-mask ratio, similar to the MPEG reference software. If we denote the energy of the i^{th} critical band by e_i , then then masking threshold for that critical band can be obtained as,

$$w_i = \begin{cases} c_i e_i & e_i > thr \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

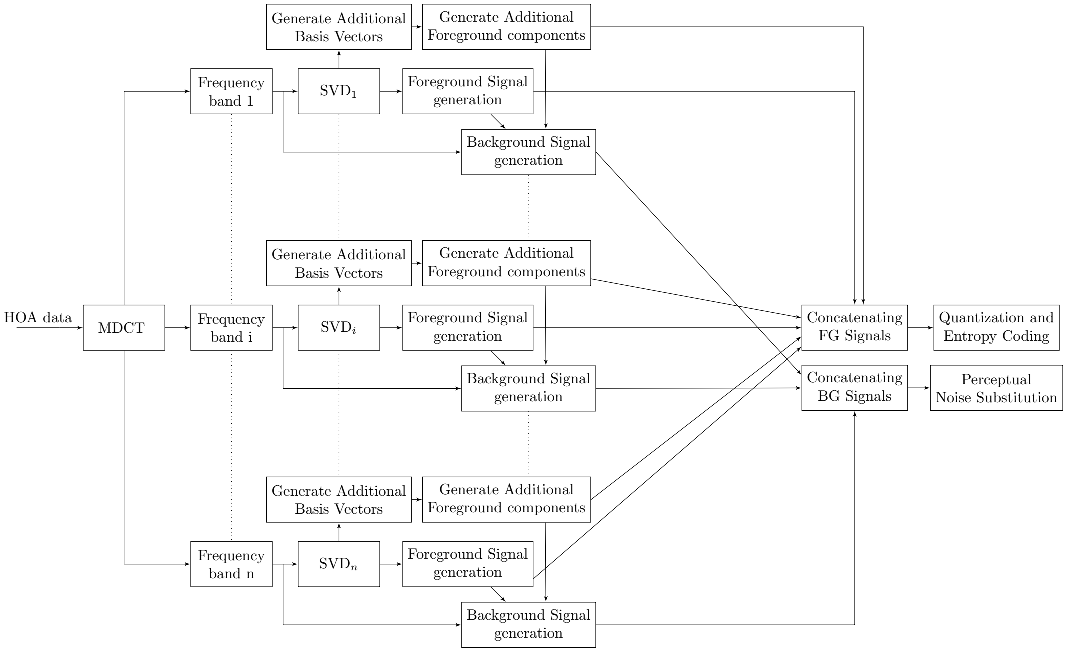

Generate Additional Basis Vectors
Generate Additional Foreground components
Frequency band 1
SVD$_1$
Foreground Signal generation
Background Signal generation
Generate Additional Basis Vectors
Generate Additional Foreground components
HOA data
MDCT
Frequency band i
SVD$_i$
Foreground Signal generation
Background Signal generation
Concatenating FG Signals
Quantization and Entropy Coding
Concatenating BG Signals
Perceptual Noise Substitution
Generate Additional Basis Vectors
Generate Additional Foreground components
Frequency band n
SVD$_n$
Foreground Signal generation
Background Signal generation

Figure 4.2: Overview of the proposed encoder

Where c_i is a pre-defined constant and thr is a global threshold value. Unlike current approaches, we propose to use a common masking threshold for all encoded channels, calculated from the average energy of all components, in a given band. Note that while we employ a simple psychoacoustic model with a fixed signal-to-mask ratio, the underlying approach based on the average energy can be extended to more sophisticated models in a straightforward manner. Next, quantization and entropy coding is performed for each predominant component using the common masking threshhold. Finally, there is no background signal compression, and the residual of the predominant components is converted back to the ambisonics domain, but only for the purpose of perceptual noise substitution as described in Chapter 2. Figure 4.2 depicts an overview of the proposed encoder.

4.3 Experimental Results

We conducted objective and subjective tests to validate the effectiveness of the proposed approach. The following codecs are compared in our experiments:

- CMPEG: Our implementation of the MPEG-H codec as described in Chapter 2.
- FSVD: Our frequency domain SVD framework described in Chapter 2.
- PROP: The approach proposed here and described in 4.2.

In CMPEG and FSVD the number of foreground and background components were set to 4 each, while PROP uses 4 predominant components from SVD and 4 additional components obtained by the proposed null-space technique. Thus in all competing coders a total of 8 components were encoded using the core audio codec. Two coding modes are available per frame in FSVD and PROP, one with a single band and the other with 4 frequency bands, where mode switching is performed to minimize rate. The test database consisted of eight third-order (16 ambisonics channels) HOA files provided by Google for UCSB research, with diverse type of audio including speech, music, singing with stationary and moving sound sources. The coders were run to minimize the maximum quantization noise to masking ratio (MNMR) criterion in all frequency bands for all encoded sound components. The coders can adjust the value of MNMR to match a given bit-rate. For both objective and subjective listening tests, HOA data were converted to stereo signals using a binaural renderer.

4.3.1 Objective Results

For preliminary evaluation, we used the average quantization noise-to-mask ratio (ANMR) of final binaural reconstructions, averaged over all frames, as the distortion

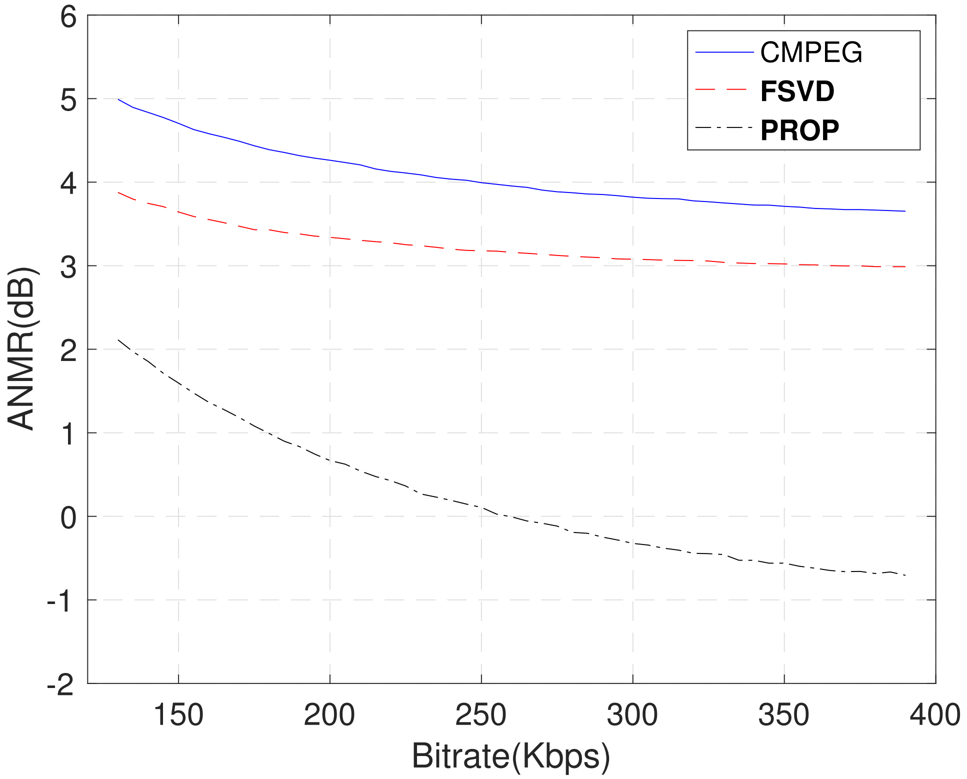


Figure 4.3: Average distortion versus bit-rate of the competing coders, evaluated and averaged over the dataset

metric to compare the competing codecs. For a meaningful comparison in this setting, perceptual noise substitution was disabled in FSVD and PROP. The performance of the three coders is compared in Figure 4.3, where average distortion is plotted versus bit-rate. Distortion at a given bit-rate has been averaged over the test files, and the bit-rate range was selected to cover a wide range of reconstruction quality. It is clear that the proposed approach provides consistent coding gains, up to 4.3dB and 3.7dB over CMPEG, and FSVD, respectively.

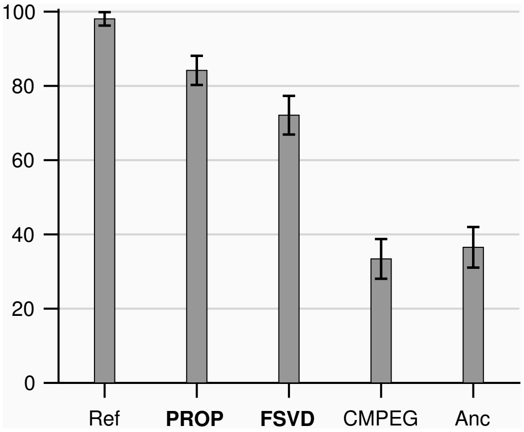


Figure 4.4: MUSHRA listening test results

4.3.2 Subjective Results

A MUSHRA [27] listening test was conducted to evaluate the perceptual gains of the proposed codec over competing methods. Ten seconds of each of the eight audio sequences were converted to stereo audio files using the binaular renderer. The following 5 versions of each of the audio items were presented in a random order to 9 listeners: the hidden reference (ref), a 3.5 kHz low passed anchor (anc), and files encoded with CMPEG, FSVD and PROP. The subjects were asked to rate each file on a scale of 0 (bad) to 100 (excellent) based on audio quality. The bit-rates were matched at about 200 Kbps. The averaged scores over all audio items and the 95% confidence intervals are depicted in Figure 4.4, where the proposed scheme is demonstrated to outperform its competitors.

4.4 Concluding Remarks

A new encoding framework for compression of HOA data is presented, where a null-space basis vector extension technique enables all compression to be performed in the SVD domain, and a jointly computed common masking threshold accounts for effects of spatial masking across components. Significant gains over existing approaches demonstrate the effectiveness of the proposed framework.

Part II

End-to-End Estimation and Design Techniques for Error-Resilient Signal Coding and Networking

Chapter 5

Asymptotic Closed-Loop Design of Error Resilient Predictive Compression Systems

5.1 Introduction

Linear prediction is widely used in speech coding, speech synthesis, speech recognition, audio coding, and video coding. In compression systems, the prediction module plays an important role in exploiting temporal and spatial redundancies. However, when such a compressed data is transmitted over unreliable networks, errors introduced due to the inevitable packet losses, propagate through the prediction loop, causing substantial, and sometimes catastrophic, deterioration of the received signal. Despite this, conventional compression system design completely ignores the effect of channel loss, and chooses the prediction parameters to minimize the mean squared prediction error, and optimizes the quantizer to minimize the reconstruction error at the encoder. This problem can be alleviated by optimizing the system for the overall end-to-end distortion (EED) observed at the decoder, which accounts for the effect of packet loss. An optimal recursive technique to estimate EED at the encoder was proposed in [29], and utilizing this distortion to optimally select the parameters for motion compensated prediction in video coding was proposed in [30]. However, designing optimal predictors and quantizers, while accounting for EED is a challenging task, as we need to work with a stable training set that accurately represents the true signal statistics. The open-loop (OL) and closed-loop (CL) approaches were proposed in [31] for predictive vector quantization and have been widely used since then. The OL approach uses the original data as the prediction reference during design, but since the decoder does not have access to the original data, the parameters designed are not suitable for the statistics seen at the decoder. The CL approach attempts to alleviates the problem of this mismatch by designing the parameters using reconstructed data obtained in a closed-loop system as the prediction reference. However, using these parameters in a closed-loop system generates new prediction and reconstruction, which implies they differ from the data the parameters were designed for. This mismatch in statistics between design and operation grows over time as the data is fed through the prediction loop in the coder, leading to instability in both estimation of prediction parameters, and design of quantizers, especially at lower bit rates. Note that this error propagation encountered during the design phase due to statistics mismatch, differs from the error propagation due to packet losses.

In this chapter we propose to address the challenging problem of tackling these two types of error propagation, by designing the system iteratively, wherein an estimated EED is minimized at each iteration to account for packet losses, and the prediction reference from the previous iteration is employed in an open-loop way to ensure statistics used for design and operation are matched. Once the parameters being designed converge, the prediction reference in the current iteration will match the reference

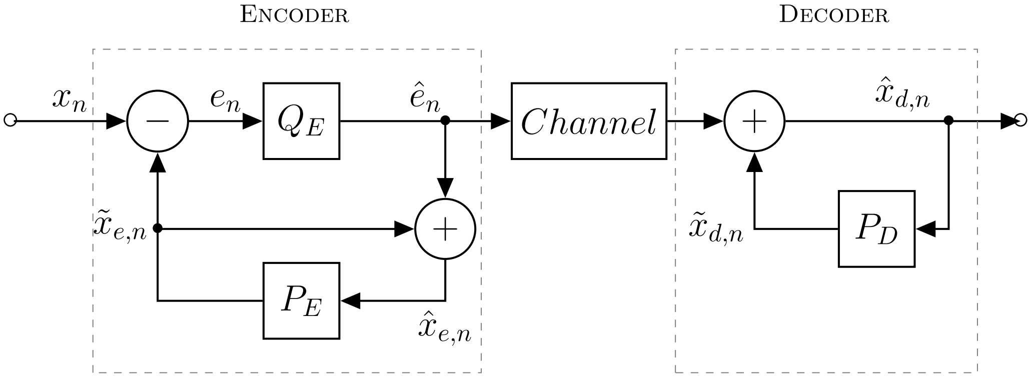


Figure 5.1: A predictive compression system

from the previous iteration, thus mimicking closed-loop operation. Hence, we call this the asymptotic closed-loop (ACL) approach, which is similar to the approach in [32, 33], wherein system design without accounting for packet losses is proposed. We specifically describe a framework for rate versus EED optimization of a compression system employing a first order linear predictor. We also propose a new encoder architecture, in which the prediction at encoder is based on the expected decoder reconstructions. Experimental results substantiate the utility of the proposed approach with significant performance improvements over existing design techniques.

5.2 Problem Setup

Figure 5.1 illustrates a predictive compression system, wherein input signal samples, $x_n,\ 0 \leq n < N$, are coded by the encoder to generate a bitstream, which is transmitted through a channel to the decoder, where it is decoded to generate the reconstructed samples. The encoder uses its previous reconstructed samples, $\hat{x}_{e,n}$ to generate predicted samples, $\tilde{x}_{e,n}$ and the prediction error, $e_n = x_n - \tilde{x}_{e,n}$. This is quantized to generate $\hat{e}_n$, and sent over the channel. When the decoder receives $\hat{e}_n$, it adds it to its predicted sample, $\tilde{x}_{d,n}$ to generate its reconstructed samples, $\hat{x}_{d,n}$. Note that the reconstructed samples at the encoder, $\hat{x}_{e,n}$, and the decoder, $\hat{x}_{d,n}$, will differ when the channel is unreliable and packets carrying $\hat{e}_n$ are lost. This uncertainty results in $\hat{x}_{d,n}$ being a random variable to the encoder. The problem at hand is to design optimal quantizers and predictors (P_E, Q_E and P_D) to minimize the expected EED at the decoder to account for packet losses. For the mean squared error distortion metric, expected EED at the decoder is,

$$\begin{aligned} E\{D\} &= \sum_{n=0}^{N-1} E\{(x_n - \hat{x}_{d,n})^2\} \\ &= \sum_{n=0}^{N-1} x_n^2 - 2x_n E\{\hat{x}_{d,n}\} + E\{(\hat{x}_{d,n})^2\}. \end{aligned} \tag{5.1}$$

Clearly, to estimate this distortion, first and second moments of the decoder reconstructions should be accurately estimated at the encoder.

5.3 Background

5.3.1 End to End distortion estimation and prediction

A recursive technique to optimally estimate the expected EED at the encoder in the presence of packet losses via the first and second moments of the decoder reconstructions was proposed in [29] for video coders. The recursive algorithm optimally estimates the decoder reconstructions' first moment, $E\{\hat{x}^j_{d,n}\}$, and the second moment, $E\{(\hat{x}^j_{d,n})^2\}$, for every pixel j in frame n. These moments are then used to estimate EED at the encoder using (5.1) to optimally switch between inter-frame prediction and intra-frame prediction, to control the error propagation through frames. In [30], a new prediction scheme is employed in conjunction with optimal EED estimation.

Conventional motion compensated prediction employs the encoder reconstructions for prediction, i.e., $\tilde{x}^j_{e,n} = \hat{x}^{j+v}_{e,n-1}$, where v is the optimal motion vector that minimizes the prediction error. Instead in [30], the prediction is based on expected decoder reconstructions, i.e., $\tilde{x}^j_{e,n} = E\{\hat{x}^{j+v}_{d,n-1}\}$, where v is the optimal motion vector that minimizes the EED of (5.1). This setup plays an important role in limiting error propagation during decoder operation by appropriately selecting motion vectors to predict from reference blocks that are less likely to be corrupted by error propagation.

5.3.2 Closed-Loop versus Asymptotic Closed-Loop Design

In closed-loop iterative design [34], the coder operates in closed-loop at each iteration to generate prediction errors and reconstructed samples that are used to design the updated quantizer and the updated predictor, respectively. At iteration $i-1$, given a quantizer, $Q^{(i-1)}$, and a predictor, $P^{(i-1)}$, a training set of prediction errors, $T^{(i)} \ : \ \{e^{(i)}_n\}^N_{n=1}$, for iteration i is generated as,

$$e^{(i)}_n = x_n - P^{(i-1)}(\hat{x}^{(i)}_{n-1}), \tag{5.2}$$

where,

$$\hat{x}^{(i)}_n = P^{(i-1)}(\hat{x}^{(i)}_{n-1}) + Q^{(i-1)}(x_n - P^{(i-1)}(\hat{x}^{(i)}_{n-1})). \tag{5.3}$$

These two equations are calculated sequentially for all values n. Then given $T^{(i)}$, we design a new quantizer, $Q^{(i)}$. Using $Q^{(i)}$, a new set of reconstructed samples, $\hat{x'}^{(i)}_n$, is generated as per (5.3) and based on this, we design a new predictor, $P^{(i)}$. These steps are repeated until convergence. Figure 5.2 illustrates this closed-loop iterative design. The major issue with this approach is that when the updated parameters are employed in closed-loop at an iteration, new prediction errors are generated, which differ from

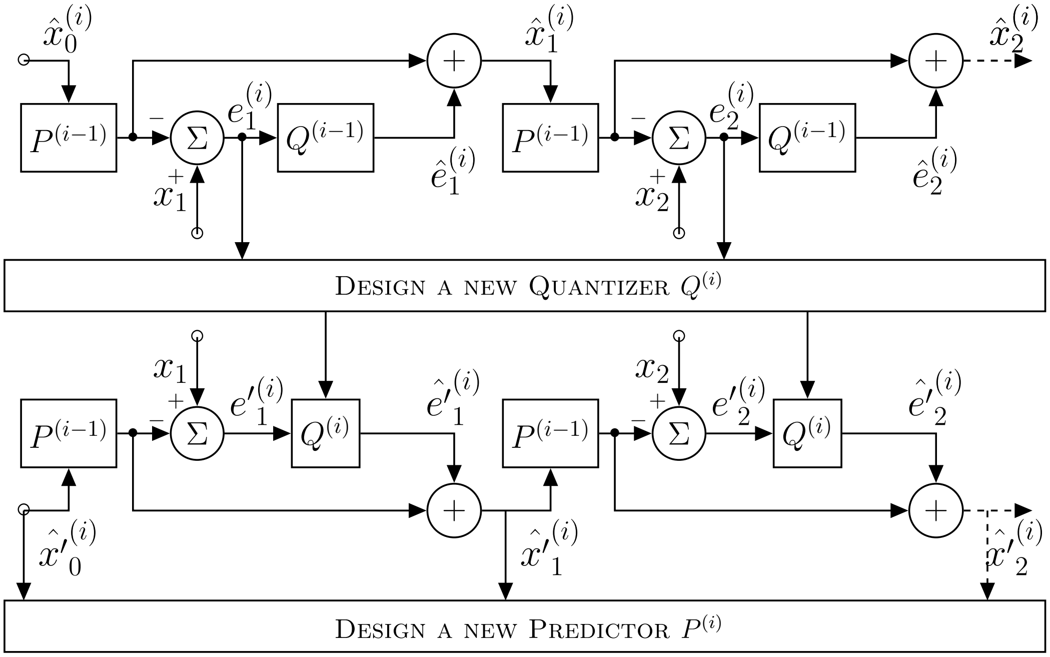


Figure 5.2: Closed-loop training approach

the errors the quantizer was designed for, and this implies different reconstructions are generated, which differ from the reference reconstructions the predictor was designed for. This mismatch in statistics between design and operation builds up over time as the data is fed through the prediction loop in the coder, leading to instability in the iterative design of both the predictor and the quantizer, especially at lower bit rates.

The ACL design technique proposed in [32, 33], tackles this statistics mismatch issue by designing the predictor and the quantizer in an open-loop fashion, while ultimately optimizing the system for closed-loop operation. Specifically, the prediction is based on reconstructions of previous iteration, i.e.,the prediction errors are generated as,

$$e_n^{(i)} = x_n - P^{(i-1)}(\hat{x}_{n-1}^{(i-1)}). \tag{5.4}$$

Given the new prediction errors, we design a new quantizer, $Q^{(i)}$. This $Q^{(i)}$ is now

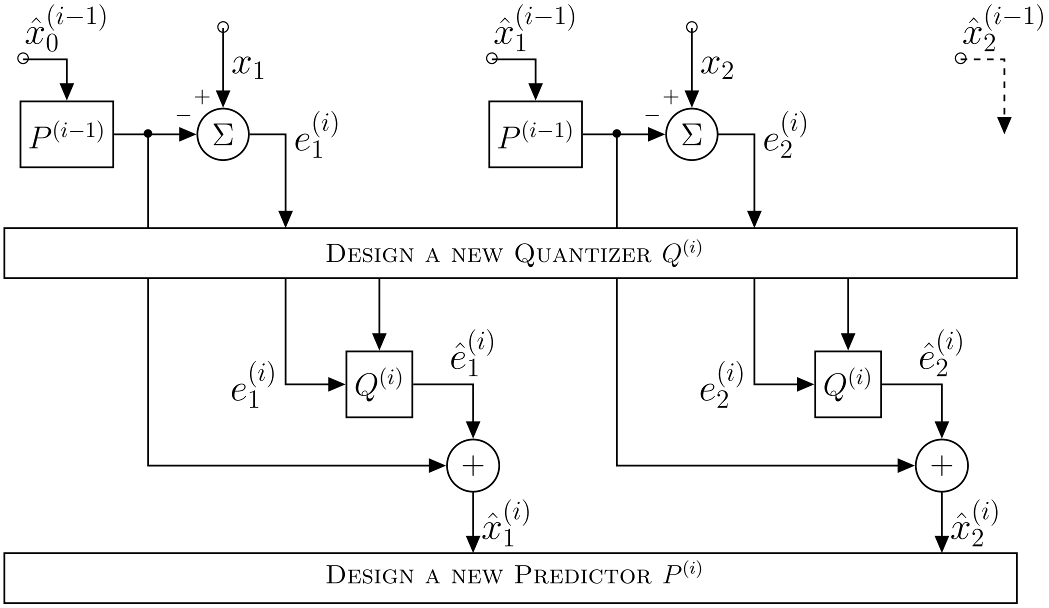


Figure 5.3: Asymptotic closed-loop training approach

employed to generate the reconstructed samples of next iteration as,

$$\hat{x}_n^{(i)} = P^{(i-1)}(\hat{x}_{n-1}^{(i-1)}) + Q^{(i)}(x_n - P^{(i-1)}(\hat{x}_{n-1}^{(i-1)})), \tag{5.5}$$

again using the reconstructions of previous iteration for prediction. Given these new reconstructions, we design a new predictor, $P^{(i)}$. Note that the equations (5.4) and (5.5) are executed independently for each sample of the sequence in an open-loop way. The main steps of this technique are depicted in Figure 5.3. The open-loop format ensures the predictor and quantizer employ exactly the same reconstructed data and prediction error used for their design, eliminating the statistical mismatch issue seen in closed-loop design. On convergence, the predictor and the quantizer do not change, which implies, $\hat{x}_{n-1}^{(i)} = \hat{x}_{n-1}^{(i-1)}$, i.e., predicting from previous iteration reconstructions is the same as predicting from the current iteration reconstructions, which is effectively closed-loop operation.

5.4 Proposed Approach

We propose a framework to design a first order predictor and a quantizer to minimize the EED in (5.1). We first develop the EED estimation algorithm, then we propose an encoder architecture in which predictions are based on the expected reconstructions at the decoder and finally we propose the ACL design approach that accounts for packet loss to improve coding efficiency and design stability.

5.4.1 Expected Decoder Distortion and Reconstructions

We assume for simplicity of presentation that each packet contains one sample (or alternatively that interleaving is used). The packet (or sample) loss rate is denoted as p. The prediction model employed at the decoder is a simple first order linear predictor,

$$\tilde{x}_{d,n} = \alpha \hat{x}_{d,n-1}, \tag{5.6}$$

where α is the prediction coefficient that needs to be estimated. The quantized prediction error, $\hat{e}_n$, transmitted over the channel, may or may not be received by the decoder. If the current packet is received (with probability $1-p$), the decoder uses it to generate the reconstructed sample as,

$$\hat{x}_{d,n} = \tilde{x}_{d,n} + \hat{e}_n. \tag{5.7}$$

When the packet is lost, a simple concealment of setting residue to zero is employed, which gives the reconstructed sample as,

$$\hat{x}_{d,n} = \tilde{x}_{d,n}. \tag{5.8}$$

Thus the first and second moment of the decoder reconstructed samples, required to estimate EED given in (5.1), are calculated recursively at the encoder as,

$$\begin{aligned} E\{\hat{x}_{d,n}\} &= (1-p)E\{\hat{e}_n + \alpha\hat{x}_{d,n-1}\} + pE\{\alpha\hat{x}_{d,n-1}\} \\ &= (1-p)\hat{e}_n + \alpha E\{\hat{x}_{d,n-1}\} \end{aligned} \tag{5.9}$$

$$\begin{aligned} E\{(\hat{x}_{d,n})^2\} &= (1-p)E\{(\hat{e}_n + \alpha\hat{x}_{d,n-1})^2\} + pE\{(\alpha\hat{x}_{d,n-1})^2\} \\ &= (1-p)(\hat{e}_n^2 + 2\alpha\hat{e}_n E\{\hat{x}_{d,n-1}\}) + \alpha^2 E\{(\hat{x}_{d,n-1})^2\} \end{aligned} \tag{5.10}$$

5.4.2 Prediction Based on the Expected Decoder Reconstructions

Packet losses cause the reconstructions at the encoder and the decoder to differ. Thus to close the gap between prediction at the encoder and the decoder, we employ the expected decoder reconstructions for prediction at the encoder, i.e.,

$$\tilde{x}_{e,n} = \alpha E\{\hat{x}_{d,n-1}\}. \tag{5.11}$$

The prediction error, $e_n = x_n - \tilde{x}_{e,n}$, is then quantized to generate, $\hat{e}_n$. The overall proposed architecture is shown in Figure 5.4.

We design the prediction coefficient α to minimize the EED, by solving for α in the equation given by setting the partial derivative of EED with respect to α to 0. The EED in (5.1) is dependent on α through equations (7.11) and (5.10). The

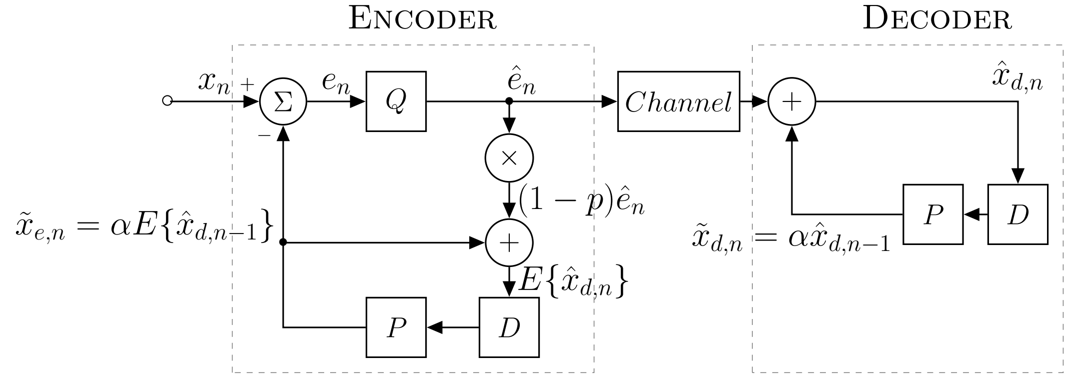


Figure 5.4: Architecture of the proposed coder

equation to be solved is,

$$\begin{aligned}
\frac{\partial E\{D\}}{\partial \alpha} &= \sum_{n=0}^{N-1} -2x_n E\{\hat{x}_{d,n-1}\} + \\
&\quad \sum_{n=0}^{N-1} 2(1-p)\hat{e}_n E\{\hat{x}_{d,n-1}\} + 2\alpha E\{(\hat{x}_{d,n-1})^2\} \\
&= 0, \qquad (5.12)
\end{aligned}$$

which gives us the solution as,

$$\alpha = \frac{\sum\limits_{n=0}^{N-1} E\{\hat{x}_{d,n-1}\}(x_n - (1-p)\hat{e}_n)}{\sum\limits_{n=0}^{N-1} E\{(\hat{x}_{d,n-1})^2\}}. \qquad (5.13)$$

Note that although $\hat{e}_n$ is dependent on α, we assume that the modifications in α across our design iterations are small enough to not change the quantization intervals.

5.4.3 Asymptotic Closed-Loop Design

We employ the ACL approach for a stable system design by eliminating the statistical mismatch issue of closed-loop design. This is achieved by operating in an open-loop way, wherein we employ previous iteration's first and second moments of the decoder reconstructions, to estimate current iteration's moments and prediction. Given a set of decoder reconstructions' first moments, $E\{\hat{x}_d\}^{(i-1)}$, and second moments, $E\{(\hat{x}_d)^2\}^{(i-1)}$, of iteration $i-1$, the predictor and quantizer are iteratively designed in an inner loop. In a subiteration s of the inner loop, given a set of quantized prediction errors, $\hat{e}_n^{(i,s-1)}$, the optimal prediction coefficient is estimated as,

$$\alpha^{(i,s)} = \frac{\sum\limits_{n=0}^{N-1} E\{\hat{x}_{d,n-1}\}^{(i-1)}(x_n - (1-p)\hat{e}_n^{(i,s-1)})}{\sum\limits_{n=0}^{N-1} E\{(\hat{x}_{d,n-1})^2\}^{(i-1)}}. \tag{5.14}$$

The new prediction errors are now generated in an open-loop fashion as,

$$e_n^{(i,s)} = x_n - \alpha^{(i,s)} E\{\hat{x}_{d,n-1}\}^{(i-1)}. \tag{5.15}$$

An optimal quantizer, $Q^{(i,s)}$, is now designed for this set of new prediction errors, which is used to generate a new set of quantized prediction errors, $\hat{e}_n^{(i,s)} = Q^{(i,s)}(e_n^{(i,s)})$. These subiterations are repeated until convergence to obtain current subiterations' final quantizer, $Q^{(i)}$, final prediction coefficient, $\alpha^{(i)}$, and final set of quantized prediction errors, $e_n^{(i)}$. The first and second moments of the decoder reconstructions are now updated in the outer loop in an open-loop way as,

$$E\{\hat{x}_{d,n}\}^{(i)} = (1-p)\hat{e}_n^{(i)} + \alpha^{(i)} E\{\hat{x}_{d,n-1}\}^{(i-1)} \tag{5.16}$$

$$\begin{aligned} E\{(\hat{x}_{d,n})^2\}^{(i)} = {} & (1-p)((\hat{e}_n^{(i)})^2 + 2\alpha^{(i)}\hat{e}_n^{(i)} E\{\hat{x}_{d,n-1}\}^{(i-1)}) + \\ & (\alpha^{(i)})^2 E\{(\hat{x}_{d,n-1})^2\}^{(i-1)}. \end{aligned} \tag{5.17}$$

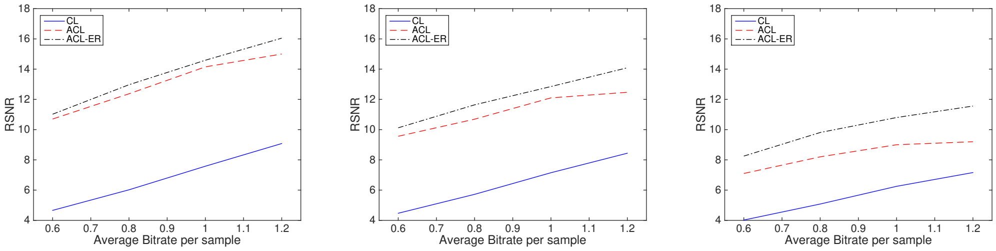
These moments are now used in the next iterations inner loop to update the predictor and quantizer. Iterations are repeated until convergence. Note that although the entire design is in open-loop, on convergence it emulates closed-loop operation. This is achieved as on convergence the quantizer and predictor do not change, i.e., $Q^{(i)} = Q^{(i-1)}$ and $\alpha^{(i)} = \alpha^{(i-1)}$, which implies $E\{\hat{x}_{d,n}\}^{(i)} = E\{\hat{x}_{d,n}\}^{(i-1)}$, thus employing previous iteration's moments is the same as estimating current iteration's moments recursively and employing them for prediction in a closed-loop way.

5.5 Experimental Results

To validate our proposed method, we evaluated it for a compression system with first order linear prediction and an entropy constrained scalar quantizer. The Generalized Lloyd Algorithm (GLA) was used to design the entropy constrained scalar quantizer. We used the 6 speech files available in the EBU SQAM database [35] as our dataset, as linear prediction is commonly employed in speech coding. However, note that the proposed approach is applicable to predictive compression of any signal with temporal correlations. The first half of the speech files were used as training set (resulting in more 2 million samples) and the second half as test data. The prediction coefficient was initialized to zero. We evaluated the following three different design techniques:

1. The closed-loop design procedure discussed in Section 5.3.2, which completely ignores the packet losses (referred as CL).

2. The ACL algorithm discussed in [33], which also ignores the packet losses and can be obtained by setting $p = 0$ in the update formulas of Section 5.4.3 (referred as ACL).



(a) 5%

(b) 10%

(c) 20%

Figure 5.5: Average decoder distortion (in dB) of CL, ACL and the proposed ACL-ER design approaches, for a first order predictive coder with entropy constrained scalar quantizer, at various bit rates for various packet loss rates

3. Our proposed method (referred as ACL-ER).

The system performance is evaluated by plotting the decoder reconstruction error's signal to noise ratio (RSNR) averaged over ten different loss patterns versus average bitrate, as shown in Figure 5.5 for different packet loss rates. Clearly, the proposed approach consistently outperforms both ACL and CL under all testing scenarios, with gains of up to 7 dB over CL and gains of up to 2.5 dB over ACL. The proposed approach provides higher performance improvements as the packet loss rate increases, since accounting for error propagation due to packet losses becomes critical in these cases. Compared to ACL, our method provides larger gains at higher bitrates, as ACL relies more on the high quality previous reconstructions for prediction, unaware of the fact that these reconstructions at the decoder will be corrupted as a result of error propagation due to packet losses. These results substantiate the significant utility of the proposed approach.

5.6 Concluding Remarks

In this chapter we proposed an effective and robust design technique for a predictive compression system. We account for the presence of an unreliable channel by designing the system to optimize the EED at the decoder. We then eliminate the statistical mismatch issue suffered by conventional closed-loop approaches, by employing a stable iterative design approach that operates in an open-loop way and on convergence mimics closed-loop operation. By carefully designing the system parameters, error propagation at the decoder is effectively contained. Significant performance improvements seen in experimental evaluation results demonstrate the utility of the proposed approach. Future research directions include, extending the proposed design technique to higher order predictors, and employing a powerful optimization technique for design of the entropy constrained scalar quantizer to account for packet losses.

Chapter 6

Recursive End-To-End Distortion Estimation for Error-Resilient Adaptive Predictive Compression Systems

6.1 Introduction

As discussed in previous chapter, robustness to packet loss is a crucial requirement, especially in the case of predictive coding, where the prediction loop propagates errors and causes substantial, and sometimes catastrophic, deterioration of the received signal. The problem of packet loss is mitigated by adding redundancy in the bitstream to recover from errors, e.g., by resetting prediction [36] at appropriate intervals to stop propagation of error, or employing error correcting codes [37] to protect critical information. In such scenarios, the overall performance of coders depends on optimizing the trade-off between compression and redundancy for error resilience. A formal illus-

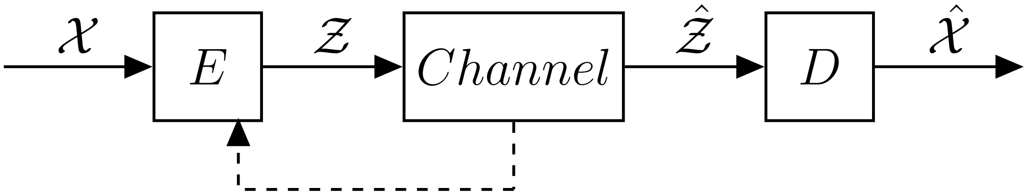


Figure 6.1: A general compression and communication system

tration of the problem setup is shown in Figure 6.1. Encoder E compresses source $\mathcal{X}$ to $\mathcal{Z}$, while accounting for channel or network unreliability. The Decoder D receives $\hat{\mathcal{Z}}$ and decodes the reconstructed source $\hat{\mathcal{X}}$. The overall problem is formally posed as optimizing encoder parameters and decisions to minimize the end-to-end distortion (EED), which accounts for quantization, packet loss, error propagation and concealment at the decoder, given the prescribed bit rate. Clearly, effective EED estimation at the encoder is critical to solving this problem.

In [29] the recursive optimal per-pixel estimate (ROPE) of EED for video coders was proposed, wherein EED is estimated at the encoder via tracking the first and second moments of the reconstructed signal at the decoder, which are recursively updated. ROPE was demonstrably optimal for the video coding setting it addressed, and its superior accuracy yielded significant performance gains over earlier heuristic methods. Nevertheless, ROPE was derived for a rather simple setting, which limits its applicability to more general settings. Specifically, ROPE was derived for a predictor with single tap for every pixel in a video frame (pointing to a motion-compensated position in the previous frame), but many coders employ a combination of short term and long term prediction filters, which lead to complex dependencies across consecutive samples. Moreover, ROPE assumes a fixed temporal prediction coefficient, which is obviously not affected by packet loss, while many compression techniques use time varying prediction parameters adapted to the local statistics. When a packet generated by an adaptive predictive coder is lost, information necessary to determine the prediction parameters is lost as well. Thus recursively estimating the EED while accounting for this uncertainty entails considerable challenges. Some techniques were previously proposed in [38, 39] to extend ROPE for handling cross correlation terms that arise due to basic interpolation filters, by employing certain approximations relevant to the context of video coding, but they do not account for adaptive prediction parameters. Note that a somewhat related problem setup exists in networked control systems (NCS) [40, 41], wherein observations or innovations from sensors are transmitted over unreliable networks and the receiver performs state estimation to make controller decisions. These systems only account for channel unreliability for state estimation at the receiver, which is equivalent to packet loss concealment in networked compression systems. Instead we propose tackling the problem of accounting for the network reliability at the encoder.

In this chapter we substantially generalize the ROPE framework to estimate EED at the encoder for a compression system which employs a higher order predictor with adaptive prediction parameters. We specifically derive a recursive procedure to estimate EED by separately tracking statistics of both prediction parameters and the reconstructed signal at the decoder, which are then effectively combined to estimate the overall EED. The accuracy and efficacy of the estimation is shown via simulation results which substantiate that incorporating such information in making RD optimal decisions of prediction resets at the encoder can achieve significant performance gains.

6.2 Proposed End-to-end Distortion Estimation

In this section we describe EED estimation for a compression system employing a higher order predictor with adaptive prediction parameters by simultaneously tracking statistics for relevant decoder quantities, namely, prediction parameters and the reconstructed signal. First we explain the general estimation algorithm, then we describe extension for a compression system using a cascade of short term and long term predictor, and finally we describe how EED is employed to optimize RD decisions at the encoder.

6.2.1 General EED Estimation Framework

Recall from previous chapter that in order to estimate distortion at the decoder, first and second moments of the decoder reconstructions should be accurately estimated at the encoder. Here we illustrate how the decoder reconstructions' moments can be estimated for an adaptive predictive system employing higher order predictors.

The common approach for adapting to local statistics is to divide the input signal into frames and employ different prediction parameters for each frame. Let $x^f[n]$ and $\hat{x}_e^f[n]$ denote the original and encoder reconstruction value of sample n in frame f, respectively. The predicted samples of frame f using a higher order predictor are given by,

$$\tilde{x}_e^f[n] = \sum_{i=1}^{P} \gamma_e^f[i]\hat{x}_e^f[n-i], \tag{6.1}$$

where P is the prediction order and $\gamma_e^f[i]$ is the ith prediction coefficient used for samples in frame f. Given the predicted samples, the quantized prediction error, $\hat{e}^f[n]$, is generated as in 5.2, and is transmitted along with the prediction parameters, $\boldsymbol{\gamma}_e^f = [\gamma_e^f[1],\ \ldots,\ \gamma_e^f[P]]$, in a single packet over the channel. Due to lossy nature of the channel the packet may either be received by the decoder, or lost. For simplicity of presentation (and without loss of generality) let us model the channel loss with a Bernoulli model, where each packet is lost independently of other packets, with probability p, called packet lost rate (PLR). Upon receiving the packet, the decoder adds the quantized error, $\hat{e}^f[n]$, to its predicted sample, $\tilde{x}_d^f[n]$, and generates its reconstructed sample, $\hat{x}_d^f[n]$. The predicted samples at the decoder are given by,

$$\tilde{x}_d^f[n] = \sum_{i=1}^{P} \gamma_d^f[i]\hat{x}_d^f[n-i], \tag{6.2}$$

where $\gamma_d^f[i]$ is the ith prediction coefficient employed in frame f at the decoder. If a packet is lost, concealment is done by assuming the quantized prediction error was zero and copying the prediction parameters from the previous reconstructed frame, $\boldsymbol{\gamma}_d^f = [\gamma_d^{f-1}[1], \ \ldots, \ \gamma_d^{f-1}[P]]$. Recall that because of packet loss, the predicted samples, reconstructions, and prediction parameters employed ($\tilde{x}_d^f[n]$, $\hat{x}_d^f[n]$, and $\boldsymbol{\gamma}_d^f$, respectively) at the decoder differ from corresponding quantities at the encoder, and must be viewed as random variables by the encoder.

Let f_R denote the event that the packet containing information of frame f is received and let f_L denote the event that it is lost. Then the first moment of the reconstructed sample at the decoder can be expressed as,

$$E\{\hat{x}_d^f[n]\} = (1-p)E\{\hat{x}_d^f[n]|f_R\} + pE\{\hat{x}_d^f[n]|f_L\}, \tag{6.3}$$

where,

$$
\begin{aligned}
E\{\hat{x}_d^f[n]|f_R\} &= E\{(\hat{e}^f[n] + \sum_{i=1}^{P} \gamma_e^f[i]\hat{x}_d^f[n-i])|f_R\} \\
&= \hat{e}^f[n] + \sum_{i=1}^{P} \gamma_e^f[i] E\{\hat{x}_d^f[n-i]|f_R\} \qquad (6.4)
\end{aligned}
$$

$$
E\{\hat{x}_d^f[n]|f_L\} = E\{\sum_{i=1}^{P} \gamma_d^{f-1}[i]\hat{x}_d^f[n-i])|f_L\}. \qquad (6.5)
$$

Note that in (7.13), both $\gamma_d^{f-1}[i]$ and $\hat{x}_d^f[n-i]$ are random variables. If we further assume them to be uncorrelated, we can approximate the first moment for event f_L as,

$$
E\{\hat{x}_d^f[n]|f_L\} \approx \sum_{i=1}^{P} E\{\gamma_d^{f-1}[i]\} E\{\hat{x}_d^f[n-i]|f_L\}. \qquad (6.6)
$$

Note that while one may object from a source coding perspective that a source and its prediction parameters would normally be correlated, but it is important to keep in mind that the only uncertainty of the encoder (and hence the only source of randomness) about the reconstructed samples and the prediction parameters is due to unreliability of the channel. The validity of this assumption is verified in the experimental results. Based on the concealment strategy adopted, the first moment for the prediction parameter vector employed at the decoder is also estimated recursively as,

$$
E\{\boldsymbol{\gamma}_d^f\} = (1-p)\boldsymbol{\gamma}_e^f + pE\{\boldsymbol{\gamma}_d^{f-1}\}. \qquad (6.7)
$$

Substituting (7.12) and (7.15) in (7.11) we obtain a recursive estimate for the first moment of reconstructed samples at the decoder. Similarly, for the second moment,

$$E\{(\hat{x}_d^f[n])^2\} = (1-p)E\{(\hat{x}_d^f[n])^2|f_R\} + pE\{(\hat{x}_d^f[n])^2|f_L\}, \tag{6.8}$$

where,

$$\begin{aligned} E\{(\hat{x}_d^f[n])^2|f_R\} &= E\{(\hat{x}_d^f[n](\hat{e}^f[n] + \sum_{i=1}^{P} \gamma_e^f[i]\hat{x}_d^f[n-i]))|f_R\} \\ &= \hat{e}^f[n]E\{\hat{x}_d^f[n]|f_R\} + \\ &\quad \sum_{i=1}^{P} \gamma_e^f[i]E\{\hat{x}_d^f[n]\hat{x}_d^f[n-i]|f_R\} \qquad (6.9) \\ E\{(\hat{x}_d^f[n])^2|f_L\} &= E\{(\hat{x}_d^f[n](\sum_{i=1}^{P} \gamma_d^{f-1}[i]\hat{x}_d^f[n-i]))|f_L\} \\ &\approx \sum_{i=1}^{P} E\{\gamma_d^{f-1}[i]\}E\{\hat{x}_d^f[n]\hat{x}_d^f[n-i]|f_L\} \qquad (6.10) \end{aligned}$$

The correlation terms in (7.17) and (7.18) can be calculated from the past correlation terms as,

$$\begin{aligned} E\{\hat{x}_d^f[n]\hat{x}_d^f[n-i]|f_R\} &= E\{(\hat{x}_d^f[n-i](\hat{e}^f[n] + \\ &\quad \sum_{j=1}^{P} \gamma_e^f[j]\hat{x}_d^f[n-j]))|f_R\} \\ &= \hat{e}^f[n]E\{\hat{x}_d^f[n-i]|f_R\} + \\ &\quad \sum_{j=1}^{P} \gamma_e^f[j]E\{\hat{x}_d^f[n-i]\hat{x}_d^f[n-j]|f_R\} \end{aligned} \tag{6.11}$$

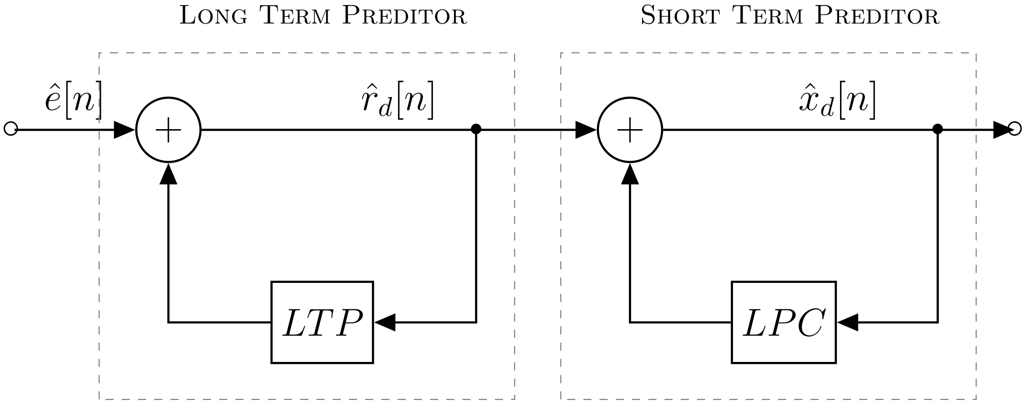


Figure 6.2: Decoder section of a speech coder with cascade of predictors

$$\begin{aligned}E\{\hat{x}_d^f[n]\hat{x}_d^f[n-i]|f_L\} &= E\{(\hat{x}_d^f[n-i](\sum_{j=1}^{P}\gamma_d^{f-1}[j]\hat{x}_d^f[n-j]))|f_L\}\\ &\approx \sum_{j=1}^{P} E\{\gamma_d^{f-1}[j]\}E\{\hat{x}_d^f[n-i]\hat{x}_d^f[n-j]|f_L\}\end{aligned} \tag{6.12}$$

Overall, equations (7.17) to (7.22) are employed to recursively estimate the second moment of reconstructed samples at the decoder. Given the first and second moments, EED is estimated using (5.1).

6.2.2 EED Estimation for Cascaded Predictors

In many real-world predictive coders, higher order predictors are implemented as a combination of multiple predictors. For example, speech coders [9] employ a cascade of a short term prediction filter (known as the linear predictive coding (LPC) filter) and a long term prediction (LTP) filter. Figure 6.2 illustrates the decoder section of an example speech coder. The decoder processes the received quantized prediction error, $\hat{e}^f[n]$, through the LTP synthesis filter to reconstruct the excitation signal,

$\hat{r}_d^f[n]$, as,

$$\hat{r}_d^f[n] \quad = \quad \sum_{i=0}^{P_1-1} \beta_d^f[i]\hat{r}_d^f[n - T_d^f - i] + \hat{e}^f[n], \tag{6.13}$$

where $\beta_d^f[i]$ is the ith LTP filter coefficient, T_d^f is the lag parameter, and P_1 is the number of LTP filter taps. The LPC synthesis filter uses the reconstructed excitation signal to generate the reconstructed samples as,

$$\hat{x}_d^f[n] \quad = \quad \sum_{j=1}^{P_2} \alpha_d^f[j]\hat{x}_d^f[n-j] + \hat{r}_d^f[n], \tag{6.14}$$

where $\alpha_d^f[i]$ is the jth LPC prediction coefficient and P_2 is the LPC filter order. We can easily combine (7.7) and (7.8) to form a single prediction filter, $\hat{x}_d^f[n] = \hat{e}^f[n] + \tilde{x}_d^f[n]$, where,

$$\begin{aligned} \tilde{x}_d^f[n] \quad = \quad & \sum_{j=1}^{P_2} \alpha_d^f[j]\hat{x}_d^f[n-j] + \sum_{i=0}^{P_1-1} \beta_d^f[i](\hat{x}_d^f[n - T_d^f - i] - \\ & \sum_{j=1}^{P_2} \alpha_d^f[j]\hat{x}_d^f[n - T_d^f - i - j]). \end{aligned} \tag{6.15}$$

Clearly, (7.9) is similar to (6.2), wherein P and $\gamma_d^f[i]$ of (6.2) can be written in terms of P_1, P_2, $\beta_d^f[i]$, $\alpha_d^f[j]$, and T_d^f of (7.9). Thus, as would be expected, the estimation framework proposed in Section 6.2.1 is applicable to coders with cascaded predictors.

6.2.3 Employing Estimated EED for Encoder Decisions

A common approach to combat error propagation through the prediction loop is to introduce prediction resets [42] at the encoder to halt dependency on past frames. While these resets stop the error propagation due to packet losses, they come at the cost of increased rate, so optimizing the number and location of resets plays a crucial role in achieving the right balance between compression efficiency and robustness to packet losses. Conventional methods use random resets at a rate equal to the PLR to stop error propagation. Instead, we leverage the proposed EED estimate as computed by the encoder to directly minimize the EED for the prescribed bit rate all within the encoder RD optimization framework. This results in optimal selection of location and number of resets. Specifically to encode frame f, we choose the mode (reset or no reset) to minimize the rate-distortion cost function,

$$J^f = D^f + \lambda R^f, \tag{6.16}$$

where R^f is the rate needed, D^f is the estimated EED, and Lagrange multiplier λ controls the RD operating point.

6.3 Results

To validate the accuracy and efficacy of our proposed method we employed it in a coder with cascade of predictors similar to 6.2.2. We used the 6 speech files available in the EBU SQAM database [35] as our dataset. We set $P_1 = 5$ and $P_2 = 12$, while operating with frames of 20ms sampled at 16 kHz. We estimated the LPC and LTP parameters in an open-loop for each frame and used them to generate the open-loop prediction error. We then designed a fixed rate 4-bits scalar quantizer for the entire prediction error sequence. Finally, we employed the prediction parameters and the designed quantizer in a closed-loop system to generate the quantized prediction error that is sent to the decoder along with all the parameters every frame at a fixed rate.

In Figure 6.3 we plot the actual SNR experienced at the decoder (averaged over

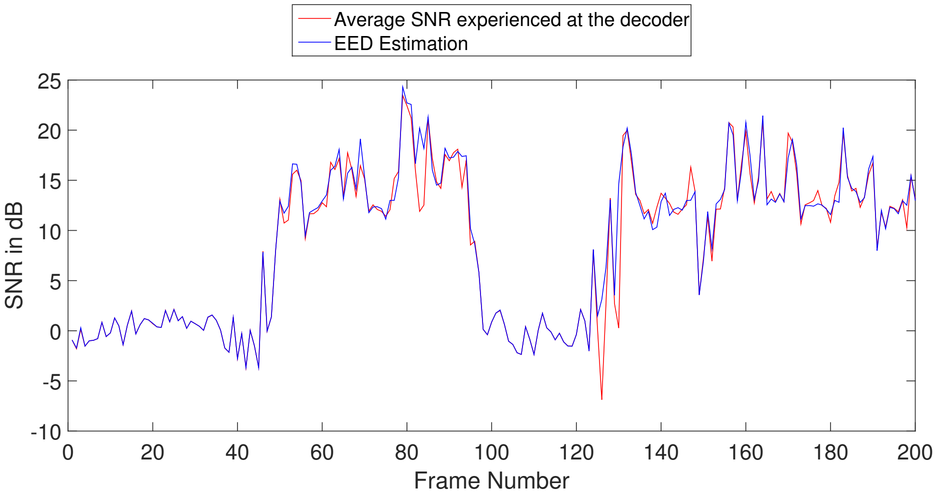


Figure 6.3: Comparison of average SNR experienced at the decoder (red) and estimated SNR (blue) for a the speech file *English Male* with a PLR of 5 %

200 different loss patterns) and the estimated SNR obtained at the encoder by our proposed framework, for 200 frames of the speech file *English Male*, operating at 5% PLR. It is clearly evident that our estimate is fairly accurate in tracking the actual SNR experienced at the decoder.

We then compared our proposed strategy for deciding resets to that of using random resets at a rate equal to PLR. Since we employ fixed rate quantizers in our experimental setup, the cost used to decide resets, as explained in Section 6.2.3, simplifies to only the EED estimate. We limited the evaluation to 8 seconds of each speech file for time efficient evaluation. In Table 6.1 and Table 6.2 we compare SNR experienced at the decoder (averaged over 50 loss patterns) for the two competing prediction reset strategies at 5% and 10% PLR, respectively. For the random reset strategy, we additionally tried 10 different reset patterns, thus obtaining the final SNR as an average over 500 simulations. Clearly, the proposed approach consistently outperforms the random reset scheme under all testing scenarios, with gains of up to 7.8 dB, and an average gain of 2.9 dB and 4.6 dB for 5% and 10% PLR, respectively.

Sequence	Average SNR in dB Random Resets	Average SNR in dB Proposed Approach
English Female	9.72	11.87
English Male	8.99	12.57
French Female	8.78	11.89
French Male	9.7	12.62
German Female	4.64	7.68
German Male	8.81	11.61
Average	8.44	11.37

Table 6.1: Comparison of average SNR experienced at the decoder for random reset and proposed reset strategies at PLR = 5%

Sequence	Average SNR in dB Random Resets	Average SNR in dB Proposed Approach
English Female	5.38	8.52
English Male	5.87	8.38
French Female	-1.21	6.72
French Male	5.31	8.48
German Female	-5.46	1.79
German Male	5.01	8.89
Average	2.48	7.13

Table 6.2: Comparison of average SNR experienced at the decoder for random reset and proposed reset strategies at PLR = 10%

6.4 Concluding Remarks

This chapter proposed an effective technique to estimate EED in an adaptive predictive compression system. Specifically, we proposed to account for the effect of packet losses on distortion at the decoder by separately tracking statistics of the employed prediction parameters and the reconstructions at the decoder. We then demonstrated incorporating the estimate obtained by the proposed approach in an RD framework to decide the number and location of prediction resets to achieve the right balance between compression and addition of redundancy to combat packet losses. Significant performance improvements seen in experimental evaluation results demonstrate the utility of the proposed approach.

Chapter 7

Recursive Estimation of End-To-End Distortion in Speech Coders

7.1 Introduction

In the previous chapter, we extended the ROPE framework to a more general setting, where higher order predictors adapted to local signal statistics are employed in the coder. Specifically we effectively estimated the EED for a compression system which uses a combination of short term and long term prediction filters with frame-wise varying parameters by simultaneously tracking the decoder statistics of the reconstructed signal and the prediction parameters.

Implementing this framework in real speech coders, introduces further challenges, as speech coders often employ prediction and interpolation of short term filter parameters in line spectral frequency (LSF) [43] domain, while these parameters are later used in LPC domain to generate reconstructions. As a result, when packets are lost,

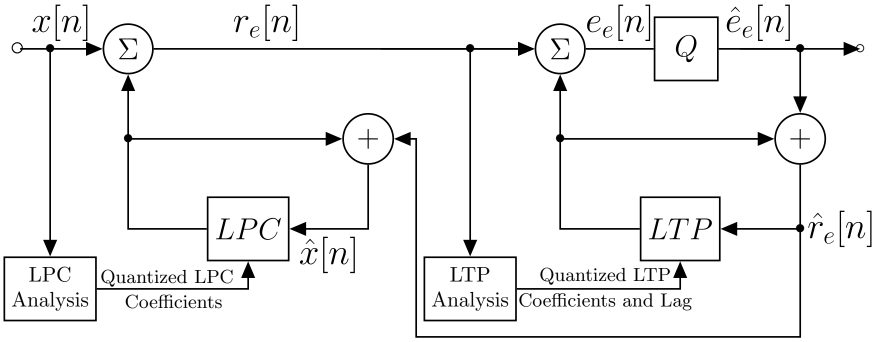


Figure 7.1: Encoder section of a speech coder with cascade of predictors

prediction of filter parameters causes error propagation in prediction parameters and subsequently in reconstructed signal. In this chapter, we propose a recursive algorithm to estimate the overall distortion at the decoder, to account for packet losses and error concealment by separately tracking the statistics of the prediction parameters and reconstructions at the decoder in their respective domains, and fusing them together for the final EED estimate. The proposed approach is incorporated within the G723.1 [44] speech coder and EED estimating formulations are derived specifically for the codec's concealment and packetization schemes. The accuracy of estimation is illustrated via simulations, and evaluation results also demonstrate how incorporating this estimate in making RD decisions at encoder can lead to considerable performance improvements under unreliable channel conditions.

7.2 Background: Speech Coding

Virtually all modern speech coders [9] exploit linear prediction [15, 45] to remove temporal correlations naturally present in speech signals. First a short term prediction filter (known as the linear predictive coding (LPC) filter), which models the human vocal tract, decorrelates speech samples that are close by, and then a long term prediction (LTP) filter removes additional redundancies due to periodicity of voiced parts of speech. In this section we provide an overview of speech coding in general and whenever necessary we discuss G723.1 coder specifications. Figure 7.1 illustrates an overview of the encoder section of an example speech coder where a cascade of a short term and long term predictor is used. To adapt to local signal statistics, the input speech signal is first divided into frames, which are further divided into smaller subframes, and different prediction parameters are estimated for each subframe. G723.1 coder operates on speech signal sampled at 8 kHz with frames of 240 samples and four subframes of 60 samples. Let $x^{f_s}[n]$ denote the original value of sample n in frame f and subframe s (where $0 \leq s < 4$ for G723.1). LPC Analysis on the input signal (often done in an open-loop fashion) creates different sets of LPC coefficients for each subframe, which are denoted by $\boldsymbol{\alpha}_e^{f_s} = [\alpha_e^{f_s}[1], \ \ldots, \ \alpha_e^{f_s}[P_1]]$, where P_1 is LPC filter order. The quantization of these coefficients is usually done in the line spectral frequency (LSF) domain, as the quantization of LSF parameters proved to be less sensitive to quantization noise and interpolation [46]. Any LPC filter, $A(z)$, can be split into a symmetric part $P(z)$ and an anti-symmetric part $Q(z)$ such that,

$$A(z) = 1 - \sum\nolimits_{i=1}^{P_1} \alpha[i] z^{-i} = \frac{1}{2}(P(z) + Q(z)), \tag{7.1}$$

with

$$\begin{aligned} P(z) &= A(z) + z^{-(P_1+1)} A(z^{-1}), & (7.2) \\ Q(z) &= A(z) - z^{-(P_1+1)} A(z^{-1}). & (7.3) \end{aligned}$$

The roots of the polynomials $P(z)$ and $Q(z)$ are called the LSFs, and all of these roots lie on the unit circle. $P(z)$ and $Q(z)$ can be reconstructed from a set of LSF values, $\boldsymbol{k}$, as

$$P(z) = (1 + z^{-1}) \sum_{n=0}^{\frac{P_1}{2}-1} (1-2cos(\pi k[2n])z^{-1} + z^{-2}), \tag{7.4}$$

$$Q(z) = (1 - z^{-1}) \sum_{n=0}^{\frac{P_1}{2}-1} (1-2cos(\pi k[2n+1])z^{-1} + z^{-2}). \tag{7.5}$$

The LPC coefficients of current frame are converted to a set of LSF coefficients (denoted as $\boldsymbol{k}_e^{f_s}$), and usually differentially coded. In G723.1, the quantized LSFs from previous frame's last subframe, is used to predict the LSFs of the current frame's last subframe and the LSF prediction error, $\bar{\boldsymbol{k}}_e^f$, generated as,

$$\bar{\boldsymbol{k}}_e^f = (\boldsymbol{k}_e^{f_3} - c_1) - c_2(\hat{\boldsymbol{k}}_e^{(f-1)_3} - c_1), \tag{7.6}$$

is quantized and sent to decoder, where c_1 and c_2 are constants. The LSF values for other subframes are determined by linearly interpolating between the known LSF values of last subframes. These reconstructed LSF parameters, $\hat{\boldsymbol{k}}_e^{f_s}$, are converted to LPC domain to obtain reconstructed LPC parameters, $\hat{\boldsymbol{\alpha}}_e^{f_s}$. These coefficients are used to filter the input signal and generate the LPC residual, which is the input to the LTP analysis filter.

Different sets of LTP coefficients and lags, denoted by $\boldsymbol{\beta}_e^{f_s}$ and $T_e^{f_s}$, respectively, are estimated for each subframe to minimize the LTP residual signal energy. The coefficients are quantized and used along with the lags to filter the LPC residual and generate the LTP residual signal, $e^{f_s}[n]$. The LTP residual signal is then quantized (via a fixed codebook in G723.1) and transmitted to the decoder along with quantized LSF residuals, LTP lags and coefficients, in a single packet over channel. It is worth emphasizing here that the EED framework is concerned with estimating the distortion at the decoder, and hence, is completely independent of encoder parameter selection.

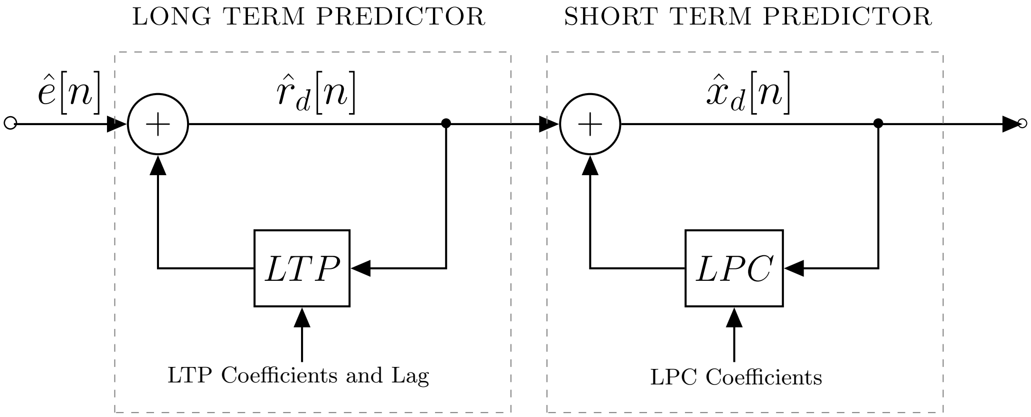


Figure 7.2: Decoder section of a speech coder with cascade of predictors

7.3 Proposed Approach to Estimate EED

Figure 7.2 shows the decoder section of the speech codec. As the channel is lossy, each packet may or may not be received by the decoder. Based on the concealment method employed, decoder generates its own version of the quantized LTP residue, prediction parameters, and reconstructed samples, which might differ from the corresponding quantities at the encoder, and hence, must be viewed as random variables by the encoder. Let f_R denote the event that the packet containing information of frame f is received. In this case, previously reconstructed LPC residual is used with the received quantized LTP residue, $\hat{e}^{f_s}[n]$, in the LTP synthesis filter to generate newly reconstructed LPC residual,

$$\hat{r}_d^{f_s}[n]|f_R = \sum\nolimits_{i=K_L}^{K_U} \hat{\beta}_e^{f_s}[i]\hat{r}_d^{g(\nu)}[n - T_e^{f_s} - i] + \hat{e}^{f_s}[n], \tag{7.7}$$

where, $\hat{\beta}_e^{f_s}[i]$ is the ith LTP filter coefficient, K_L and K_U are lower and upper limits of LTP filter ($K_L = -2$ and $K_U = 2$ in G.723.1), $T_e^{f_s}$ is the lag parameter, and $g(\nu)$ is a function that determines subframe index in the previous frame, depending on a set of parameters, $\nu = [n, T_e^{f_s}, i]$. The LPC synthesis filter uses the reconstructed LPC residual to generate the reconstructed samples as,

$$\hat{x}_d^{f_s}[n]|f_R \;=\; \sum\nolimits_{j=1}^{P_1} \hat{\alpha}_d^{f_s}[j]\hat{x}_d^{f_s}[n-j] + \hat{r}_d^{f_s}[n]|f_R, \tag{7.8}$$

where $\hat{\alpha}_d^{f}[j]$ is the decoder reconstructed jth LPC coefficient and P_1 is the LPC filter order. We can easily combine (7.7) and (7.8) to form a single synthesis filter,

$$\begin{aligned}\hat{x}_d^{f_s}[n]|f_R = \hat{e}^{f_s}[n] + &\sum\nolimits_{j=1}^{P_1} \hat{\alpha}_d^{f_s}[j]\hat{x}_d^{f_s}[n-j] + \\ &\sum\nolimits_{i=K_L}^{K_U} \hat{\beta}_e^{f_s}[i]\Big(\hat{x}_d^{g(\nu)}[n-T_e^{f_s}-i] - \\ &\sum\nolimits_{j=1}^{P_1} \hat{\alpha}_d^{g(\nu)}[j]\hat{x}_d^{g(\nu)}[n-T_e^{f_s}-i-j]\Big) \\ = \hat{e}^{f_s}[n] + &\sum\nolimits_{i=1}^{P} \gamma_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i],\end{aligned} \tag{7.9}$$

where $h(\cdot)$ provides the index of the previous subframe to be used, P is the required number of previously reconstructed samples, and $\boldsymbol{\gamma}_d^{f_s}$ can be constructed from $\hat{\boldsymbol{\beta}}_e^{f}$, $\hat{\boldsymbol{\alpha}}_d^{f}$ and $\hat{\boldsymbol{\alpha}}_d$ of previous subframes. Note that even if the frame is received, uncertainty in previously reconstructed LSF parameters carries over to the LSF parameters of the current frame through prediction, making the overall filter coefficients, $\boldsymbol{\gamma}_d^{f_s}$, a random variable to the encoder (in contrast to the formulation in previous chapter). When a frame is lost, LSF prediction residue is assumed to be zero and LSF parameters of current frame are extrapolated from previously reconstructed parameters, resulting in a different set of overall filter coefficients, $\boldsymbol{\zeta}_d^{f_s}$. The LTP residual is also assumed to be zero and the current frame is reconstructed as prediction from previously reconstructed samples, i.e.,

$$\hat{x}_d^{f_s}[n]|f_L = \sum\nolimits_{i=1}^{P} \zeta_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i]. \tag{7.10}$$

In G723.1 during concealment, LTP filters are also assumed to be 1-tap with constant 1 coefficient, which is equivalent to using LPC residual that is a pitch repeated version of past LPC residual.

For simplicity of presentation (and without loss of generality), let us model the channel loss with a Bernoulli model, where each packet is lost independent of other packets, with probability p (which is also the packet lost rate (PLR)). Then the first moment of the reconstructed sample at the decoder can be expressed as,

$$E\{\hat{x}_d^{f_s}[n]\} = (1-p)E\{\hat{x}_d^{f_s}[n]|f_R\} + pE\{\hat{x}_d^{f_s}[n]|f_L\}, \tag{7.11}$$

where,

$$\begin{aligned} E\{\hat{x}_d^{f_s}[n]|f_R\} &= E\{(\hat{e}^{f_s}[n] + \sum_{i=1}^{P} \gamma_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i])|f_R\}, \\ &= \hat{e}^{f_s}[n] + \sum_{i=1}^{P} E\{\gamma_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i]|f_R\}, \end{aligned} \tag{7.12}$$

$$E\{\hat{x}_d^{f_s}[n]|f_L\} = E\{\sum_{i=1}^{P} \zeta_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i])|f_L\}. \tag{7.13}$$

Note that in (7.12) and (7.13), $\gamma_d^{f_s}[i]$, $\zeta_d^{f_s}[i]$, and $\hat{x}_d^{h(\nu)}[n-i]$ are all random variables. If we further assume them to be uncorrelated, we can approximate the first moments as,

$$E\{\hat{x}_d^{f_s}[n]|f_R\} \approx \hat{e}^{f_s}[n] + \sum_{i=1}^{P} E\{\gamma_d^{f_s}[i]\}E\{\hat{x}_d^{h(\nu)}[n-i]|f_R\}, \tag{7.14}$$

$$E\{\hat{x}_d^{f_s}[n]|f_L\} \approx \sum_{i=1}^{P} E\{\zeta_d^{f_s}[i]\}E\{\hat{x}_d^{h(\nu)}[n-i]|f_L\}. \tag{7.15}$$

While a source and its prediction parameters would normally be correlated from a source coding perspective, in our framework the only source of randomness is due to unreliability of the channel. Hence our assumption of otherwise is reasonable, and is validated by accurate EED estimation results in experiments. Substituting (7.14) and (7.15) in (7.11) gives the recursive formulation to estimate the first moment of reconstructed samples at the decoder. Similarly, for the second moment,

$$E\{(\hat{x}_d^{f_s}[n])^2\} = (1-p)E\{(\hat{x}_d^{f_s}[n])^2|f_R\} + pE\{(\hat{x}_d^{f_s}[n])^2|f_L\}, \tag{7.16}$$

If we denote $E\{\hat{x}_d[n]\hat{x}_d[n-i]\}$ by $R_n(i)$ then, $E\{(\hat{x}_d^{f_s}[n])^2|f_R\}$ is $R_n(0)|f_R$, and $E\{(\hat{x}_d^{f_s}[n])^2|f_L\}$ is $R_n(0)|f_L$. Now

$$\begin{aligned} R_n(0)|f_R &= E\{(\hat{x}_d^{f_s}[n](\hat{e}^{f_s}[n] + \textstyle\sum_{i=1}^{P} \gamma_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i]))|f_R\} \\ &\approx \hat{e}^{f_s}[n]E\{\hat{x}_d^{f_s}[n]|f_R\} + \\ &\quad \textstyle\sum_{i=1}^{P} E\{\gamma_d^{f_s}[i]\}E\{\hat{x}_d^{f_s}[n]\hat{x}_d^{h(\nu)}[n-i]|f_R\}, \qquad (7.17) \\ R_n(0)|f_L &= E\{(\hat{x}_d^{f_s}[n](\textstyle\sum_{i=1}^{P} \zeta_d^{f_s}[i]\hat{x}_d^{h(\nu)}[n-i]))|f_L\} \\ &\approx \textstyle\sum_{i=1}^{P} E\{\zeta_d^{f_s}[i]\}E\{\hat{x}_d^{f_s}[n]\hat{x}_d^{h(\nu)}[n-i]|f_L\}. \qquad (7.18) \end{aligned}$$

The correlation terms in (7.17) and (7.18), $E\{\hat{x}_d^{f_s}[n]\hat{x}_d^{h(\nu)}[n-i]|f_R\}$ and $E\{\hat{x}_d^{f_s}[n]\hat{x}_d^{h(\nu)}[n-i]|f_L\}$, can be expressed as $R_n(i)|f_R$ and $R_n(i)|f_L$, respectively. These can be calculated from the past correlation terms as,

$$
\begin{aligned}
R_n(i)|f_R = & E\{(\hat{x}_d^{h(\nu)}[n-i](\hat{e}^{f_s}[n] + \\
& \textstyle\sum_{j=1}^{P} \gamma_d^{f_s}[j]\hat{x}_d^{h(\nu)}[n-j]))|f_R\} \\
\approx & \hat{e}^{f_s}[n]E\{\hat{x}_d^{h(\nu)}[n-i]|f_R\} + \\
& \textstyle\sum_{j=1}^{P} E\{\gamma_d^{f_s}[j]\}E\{\hat{x}_d^{h(\nu)}[n-i]\hat{x}_d^{h(\nu)}[n-j]|f_R\}, \qquad (7.19) \\
R_n(i)|f_L = & E\{(\hat{x}_d^{h(\nu)}[n-i](\textstyle\sum_{j=1}^{P} \zeta_d^{f_s}[j]\hat{x}_d^{h(\nu)}[n-j]))|f_L\} \\
\approx & \textstyle\sum_{j=1}^{P} E\{\zeta_d^{f_s}[j]\}E\{\hat{x}_d^{h(\nu)}[n-i]\hat{x}_d^{h(\nu)}[n-j]|f_L\}. \qquad (7.20)
\end{aligned}
$$

If $E\{\boldsymbol{\gamma}_d^{f_s}\}$ and $E\{\boldsymbol{\zeta}_d^{f_s}\}$ are known, equations (7.17) to (7.20) can be employed to recursively estimate the second moment of reconstructed samples at the decoder. Given the first and second moments, EED is estimated using (5.1).

7.3.1 Prediction Parameters Statistics

During concealment, LSF parameters are obtained for the current frame by setting the LSF prediction residue to zero and extrapolating LSF values from previous frame. In G723.1, it is given as,

$$\hat{\boldsymbol{k}}_d^{f_3} = c_3(\hat{\boldsymbol{k}}_d^{(f-1)_3} - c_1) + c_1, \qquad (7.21)$$

where c_3 is a constant. From (7.6) and (7.21) the first moment of LSF parameters for current frame can be recursively updated from the first moment of the previous frame as,

$$
\begin{aligned}
E\{\hat{\boldsymbol{k}}_d^{f_3}\} &= (1-p)(\bar{\boldsymbol{k}}_e^{f_3} + c_2E\{\hat{\boldsymbol{k}}_d^{(f-1)_3}\} - c_1c_2 + c_1) + \\
& \quad p(c_3E\{\hat{\boldsymbol{k}}_d^{(f-1)_3}\} - c_1c_3 + c_1). \qquad (7.22)
\end{aligned}
$$

However, we need moments for LPC coefficients to estimate EED. To obtain them we assume that the transformation from LSF to LPC domain (denoted by $f(\cdot)$) can be approximated to be locally linear, then the moments are given as,

$$E\{\boldsymbol{\alpha}_d^f\} = E\{f\{\boldsymbol{k}_d^f\}\} \approx f\{E\{\boldsymbol{k}_d^f\}\}. \tag{7.23}$$

Given these moments of LPC coefficients and the received LTP filter parameters, $\hat{\boldsymbol{\beta}}_e^{f_s}$ and $T_e^{f_s}$, the $E\{\boldsymbol{\gamma}_d^{f_s}\}$ can be calculated. For $E\{\boldsymbol{\zeta}_d^{f_s}\}$ however, even the LTP parameters are random variables, as during concealment, we assume LTP filter parameters, $\boldsymbol{\beta}^f, T^f$, are copied from previously reconstructed frame, but there is uncertainty about having received the previous frame due to packet loss. Hence, the best the encoder can do is to employ their expected values in previous frame while calculating $E\{\boldsymbol{\zeta}_d^{f_s}\}$. Since the lag needs to be an integer, we employ the rounded version of the expected value in previous frame. Note that our LTP lag concealment approach of copying from previous frame is a small deviation from G723.1's approach, which refines the previous frame's lag in a small neighborhood around its original value, but this has minimal impact on final quality. The first moment of T_d^f is updated as,

$$E\{T_d^f\} = (1-p)T_e^f + pE\{T_d^{f-1}\}. \tag{7.24}$$

In G723.1, there is no need to track expected moment of, $\boldsymbol{\beta}^f$, as during concealment LTP filter is assumed to be 1-tap with constant 1 coefficient. However, if needed it can be tracked similar to lags in (7.24). With estimates of $E\{\boldsymbol{\gamma}_d^{f_s}\}$ and $E\{\boldsymbol{\zeta}_d^{f_s}\}$, we have all the information to estimate EED.

7.3.2 Optimizing coding mode selection

Packet loss resilience can be improved by resetting prediction in selected frames at the encoder [42]. A reset makes coding of current data independent of previously coded data, thus it can stop error propagation in the prediction loop when packets are lost. But introducing a reset is associated with either degradation of quality at a fixed rate, or increased rate for fixed quality. This represents a trade-off between removing redundancy for compression, and adding redundancy for error-resilience. Previously proposed reset approaches superficially treat this trade-off and select its location to be random or periodic at a rate equal to the PLR. Instead, we propose to incorporate the EED estimate computed by our algorithm in an RD optimization framework at the encoder, to optimally select the number and location of resets. Specifically for each frame we select the coding mode at the encoder to minimize the estimated RD cost at the decoder, $J^f = D^f + \lambda R^f$, where D^f is the expected EED, R^f is the bit rate and λ is the Lagrange multiplier. As G723.1 is a very low rate codec, we quantize and send as side information the history for the LTP filter, the LPC filter and the LSF prediction, while resetting a frame to maintain a reasonable quality of reconstruction. In this setting, the λ of the RD cost controls the amount of redundancy added for error resilience.

7.4 Experimental Results

We used the 6 speech files available in the EBU SQAM database [35] to conduct our experiments. We implemented the proposed EED estimation and coding mode selection framework in the G723.1 implementation (operating at 6.3 kbps in the multipulse mode) available online [47]. For simplicity of presentation, we have disabled the perceptual filters (formant perceptual weighting and harmonic noise weighting

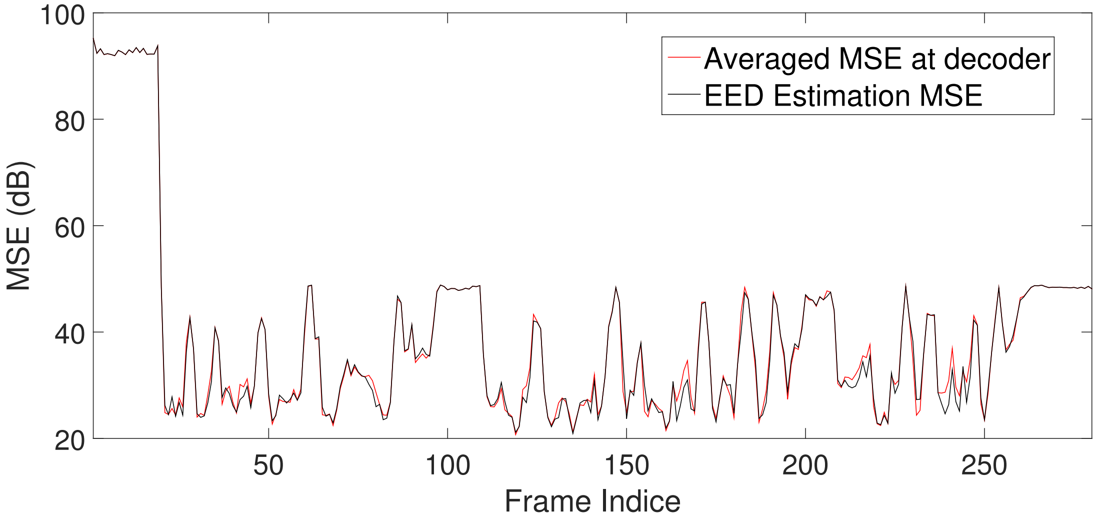


Figure 7.3: Comparison of average MSE (in dB) experienced at the decoder (red) and estimated MSE (black) for a the speech file *English Male* with a PLR of 5 %

filter), i.e., we estimate and optimize end-to-end mean squared error (MSE). We plan to incorporate perceptual filters in our EED estimation framework in future work (as they are linear filters too), with which we can estimate and optimize for end-to-end perceptual distortion criteria. In Figure 7.3 the actual MSE in dB experienced at the decoder per frame (averaged across 200 realizations of loss patterns) is compared against the estimate obtained by our algorithm, for 280 frames of the speech file *English Male* at PLR = 5% with additional redundancy of 0.85 kbps. The results clearly demonstrate our framework's capability to quite accurately estimate the overall decoder distortion.

Next we compare the performance of our EED based coding mode (reset vs no reset) selecting algorithm and the conventional random resetting technique. System performance is measured via average SNR at the decoder over 200 different packet loss patterns. For random resetting strategy, 10 different reset pattern were tested (thus the averaging was effectively over 2000 different realizations). Simulation results for PLR at 5% and 10% is provided in Figure 7.4 and Figure 7.5, respectively where we have also plotted the base G723.1 performance as a reference. Our algorithm

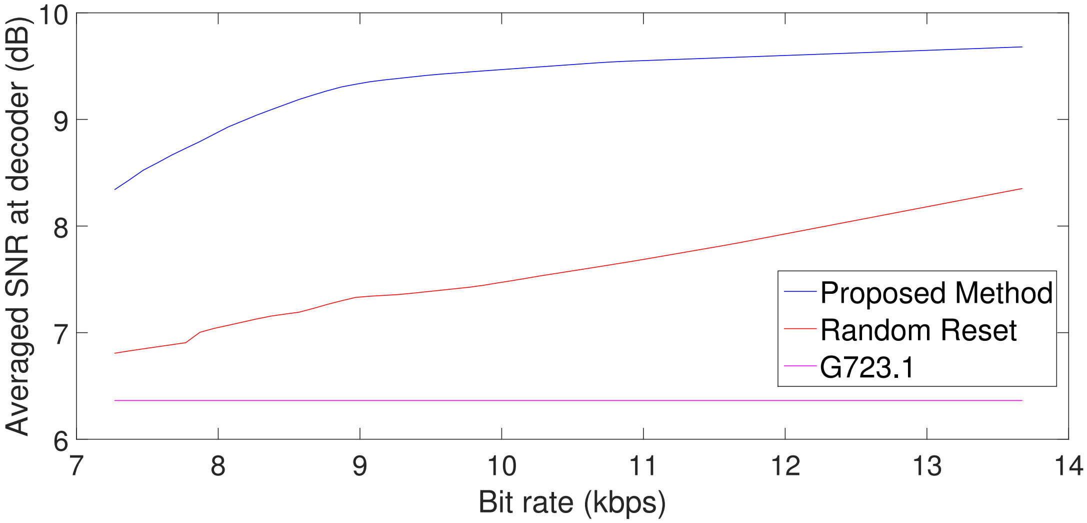


Figure 7.4: Comparison of average SNR experienced at the decoder for random reset versus the proposed reset strategy at PLR = 5%

outperforms the competition in all testing scenarios, and provides an average gain of 1.78 dB and 1.95 dB for 5% and 10% PLR, respectively.

7.5 Concluding Remarks

This chapter discusses a framework to estimate EED in speech coders. The expected moments of reconstructed samples and prediction parameters at the decoder, are independently tracked in their respective domains, and then fused to obtain the EED estimate. The accuracy of estimation is illustrated via simulations, and the utility of incorporating the estimate in RD optimization framework at the encoder is demonstrated with significant performance improvements.

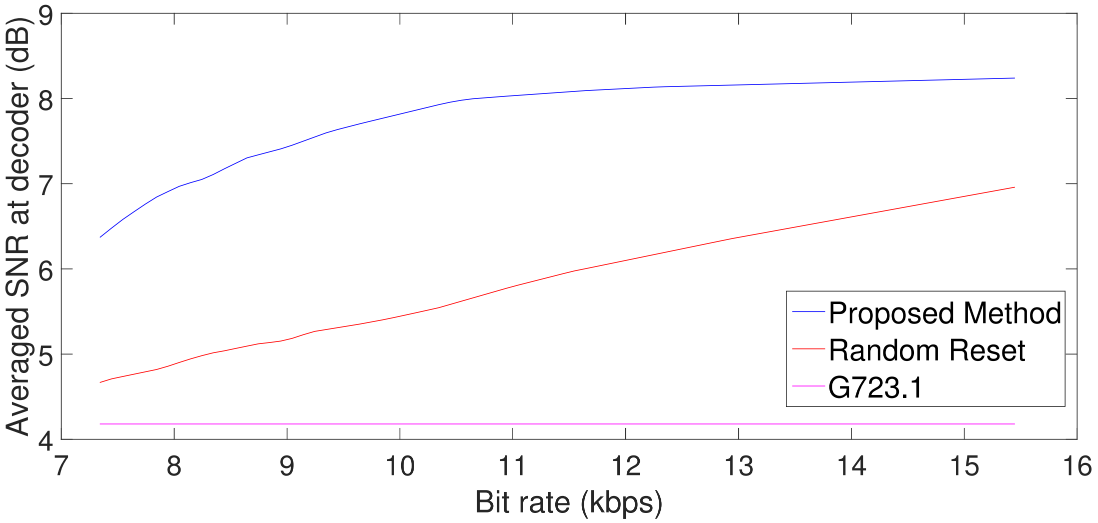


Figure 7.5: Comparison of average SNR experienced at the decoder for random reset versus the proposed reset strategy at PLR = 10%

Bibliography

[1] M. A. Gerzon, *Periphony: With-height sound reproduction*, *Journal of the Audio Engineering Society* **21** (1973), no. 1 2–10.

[2] J. Daniel, *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* PhD thesis, University of Paris VI, France, 2000.

[3] M. A. Gerzon, *General metatheory of auditory localisation*, in *Proc. 92nd AES convention*, March, 1992.

[4] J. Daniel, S. Moreau, and R. Nicol, *Further investigations of high-order ambisonics and wavefield synthesis for holophonic sound imaging*, in *Audio Engineering Society Convention 114*, 2003.

[5] J. Daniel, J.-B. Rault, and J.-D. Polack, *Ambisonics encoding of other audio formats for multiple listening conditions*, in *Audio Engineering Society Convention 105*, Sep, 1998.

[6] J. Herre, J. Hilpert, A. Kuntz, and J. Plogsties, *MPEG-H 3D audio - the new standard for coding of immersive spatial audio*, *IEEE Journal of Selected Topics in Signal Processing* **9** (2015), no. 5 770–779.

[7] R. L. Bleidt, D. Sen, A. Niedermeier, B. Czelhan, S. Fg, S. Disch, J. Herre, J. Hilpert, M. Neuendorf, H. Fuchs, J. Issing, A. Murtaza, A. Kuntz, M. Kratschmer, F. Kch, R. Fg, B. Schubert, S. Dick, G. Fuchs, F. Schuh, E. Burdiel, N. Peters, and M. Y. Kim, *Development of the mpeg-h tv audio system for atsc 3.0*, *IEEE Transactions on Broadcasting* **63** (March, 2017) 202–236.

[8] N. Peters, D. Sen, M.-Y. Kim, O. Wuebbolt, and S. M. Weiss, *Scene-based audio implemented with higher order ambisonics (HOA)*, in *SMPTE Annual Technical Conference and Exhibition*, pp. 1–13, 2015.

[9] A. S. Spanias, *Speech coding: a tutorial review*, *Proceedings of the IEEE* **82** (1994), no. 10 1541–1582.

[10] T. Painter and A. Spanias, *Perceptual coding of digital audio*, *Proceedings of the IEEE* **88** (2000), no. 4 451–515.

[11] G. K. Wallace, *The JPEG still picture compression standard*, *IEEE Transactions on Consumer Electronics* **38** (1992), no. 1 xviii–xxxiv.

[12] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, *Overview of the H. 264/AVC video coding standard*, *IEEE Transactions on Circuits and Systems for Video Technology* **13** (2003), no. 7 560–576.

[13] P. Strobach, *Linear Prediction Theory.* Springer-Verlag, Berlin, 1990.

[14] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, 2002.

[15] B. S. Atal and M. R. Schroeder, *Predictive coding of speech signals*, in *Proc. Conf. Commun., Processing*, pp. 360–361, Nov., 1967.

[16] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Prentice Hall, 1984.

[17] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Press, Boston, 1992.

[18] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann, 1996.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley-Interscience, 1991.

[20] T. Berger, *Rate Distortion Theory: Mathematical Basis for Data Compression.* Prentice Hall, 1971.

[21] *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 3: 3D audio*, 2015.

[22] H. W. Kuhn, *The hungarian method for the assignment problem*, *Naval research logistics quarterly* **2** (1955), no. 1-2 83–97.

[23] C. R. Helmrich, A. Niedermeier, S. Disch, and F. Ghido, *Spectral envelope reconstruction via igf for audio transform coding*, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 389–393, April, 2015.

[24] S. Disch, A. Niedermeier, C. R. Helmrich, C. Neukam, K. Schmidt, R. Geiger, J. Lecomte, F. Ghido, F. Nagel, and B. Edler, *Intelligent gap filling in perceptual transform coding of audio*, in *Audio Engineering Society Convention 141*, Sep, 2016.

[25] D. Sen and N. Peters, *Interpolation for decomposed representations of a sound field*, Dec. 4, 2014. WO2014194099 A1.

[26] D. Sen and S.-U. Ryu, *Compression of decomposed representations of a sound field*, Dec. 4, 2014. US20140358563 A1.

[27] *Method of Subjective Assessment of Intermediate Quality Level of Coding Systems*, 2001.

[28] A. Aggarwal, S. L. Regunathan, and K. Rose, *A trellis-based optimal parameter value selection for audio coding*, *IEEE Transactions on Audio, Speech, and Language Processing* **14** (March, 2006) 623–633.

[29] R. Zhang, S. L. Regunathan, and K. Rose, *Video coding with optimal inter/intra-mode switching for packet loss resilience*, *IEEE Journal on Selected Areas in Communications* **18** (2000), no. 6 966–976.

[30] H. Yang and K. Rose, *Source-channel prediction in error resilient video coding*, in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, pp. II–233, 2003.

[31] V. Cuperman and A. Gersho, *Vector predictive coding of speech at 16 kbits/s*, *IEEE Transactions on Communications* **33** (1985), no. 7 685–696.

[32] H. Khalil, K. Rose, and S. L. Regunathan, *The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding*, *IEEE Transactions on Image Processing* **10** (2001), no. 1 15–23.

[33] H. Khalil and K. Rose, *Robust predictive vector quantizer design*, in *Data Compression Conference (DCC)*, pp. 33–42, 2001.

[34] P.-C. Chang and R. M. Gray, *Gradient algorithms for designing predictive vector quantizers*, *IEEE Transactions on Acoustics, Speech and Signal Processing* **34** (1986), no. 4 679–690.

[35] G. Waters, *Sound quality assessment materialrecordings for subjective tests: Users handbook for the ebu–sqam compact disk*, *European Broadcasting Union (EBU), Tech. Rep* (1988).

[36] G. Côté and F. I. Kossentini, *Optimal intra coding of blocks for robust video communication over the internet*, *Signal Processing: Image Communication* **15** (1999), no. 1 25–34.

[37] Y. Wang and Q.-F. Zhu, *Error control and concealment for video communication: A review*, *Proceedings of the IEEE* **86** (1998), no. 5 974–997.

[38] Z. Chen, P. V. Pahalawatta, A. M. Tourapis, and D. Wu, *Improved estimation of transmission distortion for error-resilient video coding*, *IEEE Transactions on Circuits and Systems for Video Technology* **22** (2012), no. 4 636–647.

[39] H. Yang and K. Rose, *Advances in recursive per-pixel end-to-end distortion estimation for robust video coding in H. 264/AVC*, *IEEE Transactions on Circuits and Systems for Video Technology* **17** (2007), no. 7 845–856.

[40] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, *Kalman filtering with intermittent observations*, *IEEE Transactions on Automatic Control* **49** (2004), no. 9 1453–1464.

[41] R. T. Sukhavasi and B. Hassibi, *The Kalman-like particle filter: optimal estimation with quantized innovations/measurements*, *IEEE Transactions on Signal Processing* **61** (2013), no. 1 131–136.

[42] B. S. Atal, V. Cuperman, and A. Gersho, *Advances in speech coding*, vol. 114. Springer Science & Business Media, 1991.

[43] F. Itakura, *Line spectrum representation of linear predictor coefficients of speech signals*, *The Journal of the Acoustical Society of America* **57** (1975), no. S1 S35–S35.

[44] I.-T. Recommendation, *G.723.1 : Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*, 2006.

[45] R. P. Ramachandran and P. Kabal, *Pitch prediction filters in speech coding*, *IEEE Transactions on Acoustics, Speech, and Signal Processing* **37** (1989), no. 4 467–478.

[46] W. B. Kleijn and K. K. Paliwal, *Speech coding and synthesis.* Elsevier Science Inc., 1995.

[47] P. Kabal, *ITU-T G.723.1 speech coder: A matlab implementation, TSP lab technical report*, tech. rep., Dept. Electrical and Computer Engineering, McGill University, 2009.