UNIVERSITY of CALIFORNIA

Santa Barbara

Optimal Delayed Decisions in Encoding and Decoding of Audio Signals and General Sources

A dissertation submitted in partial satisfaction of the

requirements for the degree

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Vinay Melkote Krishnaprasad

Committee in charge:

Professor Kenneth Rose, Chair

Professor Jerry Gibson

Professor B. S. Manjunath

Professor Lawrence Rabiner

September 2010

The dissertation of Vinay Melkote Krishnaprasad is approved.

Professor Jerry Gibson

Professor B. S. Manjunath

Professor Lawrence Rabiner

Professor Kenneth Rose, Committee Chair

March 2010

Optimal Delayed Decisions in Encoding and Decoding of Audio Signals and General Sources

by

Vinay Melkote Krishnaprasad

To my parents

Acknowledgements

I thank Prof. Rose for his guidance and support through these tasking, yet fun, roller-coaster years of PhD. It is that, and the liberty for self-expression in our research group, that has brought this thesis to a personally satisfying culmination. I will through my life cherish the four and odd years that I have clocked here at beautiful Santa Barbara, and I owe much of what will constitute as happy memories, to Prof. Rose, and the UCSB community, in general.

Special thanks to Prof. Rabiner who has, during his every visit here, taken great interest in discussing my research, and provided valuable advice. Time spent assisting his classes has been a pleasant learning experience. I express my sincere thanks to Prof. Gibson and Prof. Manjunath for consenting to be on my doctoral committee. I have enjoyed pretty much every class I attended here, and I thank the faculty who taught these courses for their commitment. My gratitude to Val who, through her indefatigable self, has been an inspiration herself.

This research has been funded in part by the NSF under grant CCF-0917230, the University of California MICRO Program, Applied Signal Technology Inc., Cisco Systems Inc., Qualcomm Inc., and Sony Ericsson, Inc. Thanks to these funding agencies.

I thank all my lab-mates for the collaborations, interesting discussions (many times unrelated to any of our research), and their patience during those rare occasions when I nagged them insistently with questions at lab seminars. But most of all, I thank them for making my stay at SCL very pleasant, and one I will never regret. I thank all my friends here at SB, elsewhere in the US, as well as back in India, who have made these years of PhD seem not just all work.

My most heart-felt thanks and appreciation goes for the unfailing support, motivation, and affection provided by my parents and grand parents. Its their presence, even if far away, that has given the energy to overcome the many bumps on this road. Finally, I thank God, for everything.

Curriculum Vitæ

Vinay Melkote Krishnaprasad

Education

2010	PhD in Electrical and Computer Engineering, University of California, Santa Barbara.
2006	MS in Electrical and Computer Engineering, University of California, Santa Barbara.
2005	B. Tech. in Electrical Engineering, Indian Institute of Technology, Madras.

Experience

2007–2010	Graduate Student Researcher, University of California, Santa Barbara.
2005–2006	Teaching Assistant, University of California, Santa Barbara.
2006	Summer Intern, Audio Systems Group, Qualcomm Inc., San Diego.
2004	Summer Intern, Multimedia Codecs Group, Texas Instruments India, Bangalore.

Publications

- V. Melkote and K. Rose, "Optimal delayed decoding for predictive coding systems", *submitted to the IEEE Transactions on Signal Processing*
- V. Melkote and K. Rose, "Trellis-based approaches to rate-distortion optimized audio encoding", *IEEE Transactions on Audio, Speech, and Language Processsing*, February 2010
- V. Melkote and K. Rose, "Optimal delayed decoding of predictively encoded sources", *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2010
- V. Melkote and K. Rose, "A modified distortion metric for audio coding", *Proc. IEEE ICASSP*, April 2009 *(winner of the Best Student Paper Award)*

- V. Melkote and K. Rose, "An improved distortion measure for audio coding and corresponding two-layered trellis approach for its optimization", *Proc. 125th Audio Engineering Society (AES) Convention*, October 2008

- V. Melkote and K. Rose, "A two-layered trellis approach to audio encoding", *Proc. IEEE ICASSP*, April 2008

- V. Melkote and K. Rose, "Trellis-based joint optimization of window switching decisions and bit resource allocation for MPEG AAC", *Proc. 123rd AES Convention*, October 2007

- J. Han, V. Melkote, and K. Rose, "Transform-domain temporal prediction in video coding: exploiting correlation variation across coefficients", *Proc. IEEE International Conference on Image Processing (ICIP)*, 2010

- J. Han, V. Melkote, and K. Rose, "Estimation-theoretic approach to delayed prediction in scalable video coding", *Proc. IEEE ICIP*, 2010

- E. Ravelli, V. Melkote, T. Nanjundaswamy, and K. Rose, "Cross-layer rate-distortion optimization for scalable advanced audio coding", *Proc. 128th AES Convention*, May 2010

- J. Han, V. Melkote, and K. Rose, "Estimation-theoretic delayed decoding of video sequences encoded by prediction", *Proc. IEEE Data Compression Conference*, March 2010

- E. Ravelli, V. Melkote, T. Nanjundaswamy, and K. Rose, "Joint optimization of the perceptual core and lossless compression layers in scalable audio coding", *Proc. IEEE ICASSP*, March 2010

- E. Ravelli, V. Melkote, and K. Rose, "A perceptually enhanced scalable-to-lossless audio coding scheme and a trellis-based approach for its optimization", *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2009

- J. Han, V. Melkote, and K. Rose, "A recursive optimal spectral estimate of end-to-end distortion in video communications", *submitted to the Packet Video Workshop*, 2010

Abstract

Optimal Delayed Decisions in Encoding and Decoding of Audio Signals and General Sources

by

Vinay Melkote Krishnaprasad

This dissertation is concerned with algorithms that optimally exploit delay for encoding or decoding decisions in certain common scenarios involving signal compression.

In applications that involve off-line encoding, such as movie-streaming over the internet, music playback from hand-held devices, and so on, the end-user is not sensitive to encoding delay. Despite this fact, encoders typically compress frame after frame of the signal, thereby restricting encoding delay. As one focus of this dissertation, delayed-decision approaches are explored, to *optimize* the encoding operation over the entire signal. Standards based audio-compression is chosen as the candidate setting to demonstrate the benefits of the concept. A two-layered trellis effectively optimizes both intra- and inter-frame encoding decisions while minimizing a psychoacoustically relevant distortion measure under a prescribed bit-rate constraint. The bit-stream produced is standard compatible and there is no additional decoding delay. As an accompaniment to this rate-distortion optimization paradigm, and motivated by it, modifications are proposed to the audio distortion metric itself that enhance its psychoacoustic relevance, and endeavor to enable *subjectively optimal* decisions.

Subsequently the focus shifts to delay at the decoder end of the compression chain. Unlike at the encoder, there are no parameter choices to make. But can the decoder, by suitable application of delay, exploit correlations if any with future frames to improve the reconstruction of the current frame? This question is particularly relevant in predictive coding scenarios, where a correlated source model is explicitly assumed. The encoder predicts the current sample from the past, and codes the prediction residual. Correlations with future samples can be exploited at the decoder end, for instance by applying a non-causal filter to smooth the regular zero-delay reconstructions. In contrast, this dissertation proposes an estimation-theoretic framework where conditional probability densities, given both past and available future information (for a fixed delay), are recursively calculated, and *optimal* reconstruction computed via conditional expectation. This optimal delayed decoder in turn motivates a near-optimal low complexity approximation, that employs a time-invariant look-up table or codebook approach. Applications include video compression employing motion compensated prediction, and so called 'low-delay' applications, where predictive coding is used in lieu of transform coding to avoid large framing delays and encoding complexity.

Contents

List of Figures

List of Tables

List of Acronyms

AAC	Advanced Audio Coding
ABR	average bit-rate
ADPCM	adaptive differential pulse code modulation
ANMR	average NMR
AR	autoregressive
ATNMR	average TNMR
ATRAC	Adaptive Transform Acoustic Coder
CBR	constant bit-rate
DFT	discrete Fourier transform
DPCM	differential pulse code modulation
ET	estimation-theoretic
HCB	Huffman code book
IDPCM	interpolative DPCM
i.i.d	independent and identically distributed
IMDCT	inverse MDCT
KBD	Kaiser Bessel derived
kbps	kilo bits per second
kHz	kilo Hertz
LOT	lapped orthogonal transform

MDCT	modified discrete cosine transform
MDST	modified discrete sine transform
MMNMR	maximum MNMR
MNMR	maximum NMR
MSE	mean squared error
MTNMR	maximum TNMR
NMR	noise-to-mask ratio
NMR-BC	NMR with Bark Correction
NMT	noise-masking-tone
PAC	Perceptual Audio Coder
PAQM	perceptual audio quality measure
pdf	probability density function
PEAQ	Perceptual Evaluation of Audio Quality
QC	quantization and coding
R-D	rate-distortion
SDPCM	smoothed DPCM
SF	scale factor
SFB	scale factor band
TLS	two loop search
TNMR	total NMR
UTQ	uniform threshold quantizer
VM	verification model

Chapter 1

Introduction

The continued growth in multimedia applications has ensured the widespread utility of signal compression in its different forms, such as transform coding, predictive coding, etc. The general trend is to adopt a particular compression standard for a particular source type (for instance, H.264 for video coding [85], MPEG Advanced Audio Coding (AAC) [43, 44], etc.), which ensures that the same content is compatible with applications from different vendors. Additionally, it provides the opportunity to deploy the same encoding/decoding algorithms in various applications, although the implementation may be optimized for the specific platform. But such a straightforward deployment of existing algorithms may not ensure that system resources are well utilized. In particular, applications have varied sensitivity to encoding/decoding delays. For example, two-way communication generally imposes strict requirements for low encoding/decoding delay, live broadcasts can tolerate a relatively higher latency, applications that employ off-line coding are generally insensitive to encoding delays, and so on. Therefore, employing the same general purpose encoding/decoding techniques in

all applications will necessarily ignore the potential gains achievable due to acceptable latency in the compression chain. This dissertation reconsiders certain applications of compression with the objective of employing delay as a resource, and proposes novel algorithms that *optimally* utilize this resource to perform encoding/decoding decisions. As shall be demonstrated, substantial performance gains can be obtained via careful application of delay at the encoder or decoder for optimal coding parameter selection or data reconstruction.

We first focus on the optimal utilization of encoding delay for coding parameter selection in audio compression. Since many audio coding applications, such as audio streaming over the internet, or music playback from hand-held devices, involve content that is compressed off-line, this provides the ideal setting to realize the potential of allowed encoding delay. This focus is elaborated upon in Sec. 1.1. As we shall see, the proposed delayed decision-based audio coding algorithms considerably outperform conventional myopic general purpose encoders.

Subsequently we consider delay at the decoder end. Since the decoder is not involved in parameter selection but only reconstructs coded data, we consider the application of delay to optimally incorporate future information to improve the reconstruction of the current frame/sample. Predictive coding forms the appropriate setting here due to the implied correlations between coded data units. Although this research, discussed in more detail in Sec. 1.2, finds utility in any conventional predictive coding application, for example, motion-compensated video compression, it is of particular significance to certain emerging low-delay applications, such as audio/speech compression for blue-tooth devices, image-sensors and so on.

Before getting on with a more detailed exposition of the above topics, con-

sider the perspective of delay in classical source coding theory. Delay has been conventionally viewed via two different, yet related, formulations: block codes [78], and sliding block codes [35]. Consider a discrete source represented by the sequence of random variables $\{X_k\}$. In the former formulation, the n^{th} block of the source, $(X_{nN}, \ldots, X_{(n+1)N-1})$, is mapped by the encoder to an index i_n. The decoder, on receipt of this index, reconstructs the whole block at once to obtain $(\hat{X}_{nN}, \ldots, \hat{X}_{(n+1)N-1})$. Thus, the *blocking (or framing)* results in an average reconstruction latency or delay of $N/2$ samples. On the other hand, in sliding block codes the encoder maps the block $(X_{n-N_M}, \ldots, X_{n+N_D})$ containing the n^{th} sample, to the index i_n. The decoder typically reconstructs $\hat{X}_n$, as a function of the window of indices $(i_{n-K_M}, \ldots, i_{n+K_D})$, that includes the n^{th} index. Both encoder and decoder windows slide over by 1 sample after each encoding or decoding operation. In this case, *decision making* at the encoder and decoder entails delay, i.e., the encoder's choice of the n^{th} index takes into account the effect of this decision on N_D future samples, and the decoder decides its reconstruction of the n^{th} sample based not just on the current index i_n, but also on information from K_D future indices. With the above perspective, the focus of this dissertation can be alternately summarized as the application of delay in the tradition of sliding block codes, to optimize encoding or decoding decisions in certain well known scenarios of signal compression where, conventionally the problem is viewed as that of block coding and decisions made independently for each block, or even if a sliding block code viewpoint is adopted, the utility of delay is hardly (or sub-optimally) exploited.

1.1 Optimal Delayed Encoding

In the context of encoding delay, we consider audio compression in the framework of the AAC standard. In AAC, the audio signal is divided into overlapping frames, each of which is transformed to the frequency domain, and the transform coefficients quantized and encoded. At the decoder the reconstructed coefficients are inverse transformed, and frames overlap-added to reconstruct the time domain signal. Audio encoders generally compress frame after frame of the signal, with encoding parameters chosen almost independently for each frame. This type of coding operation can typically be visualized as a block code operating within the frequency domain, i.e., there is blocking delay, but no delayed decisions at the encoder. In reality though, most applications of audio compression, such as streaming of music over the internet, hand-held music playback devices, gaming audio, etc., involve pre-compressed data, and encoding delay is not crucial to the end-user experience. Hence the myopic approach adopted by current encoders, that restricts delay in choosing encoding parameters, is sub-optimal. We are therefore motivated to consider a joint optimization of parameter choices for all frames of the audio file. In other words, akin to a sliding block code, improved encoding decisions for each frame can be achieved by considering the effect of such decisions on frames other the one being encoded. Needless to say the utility of this delayed encoding principle is not limited to audio compression, and can be applied in any situation where off-line coding is involved.

We cast the problem as one of rate-distortion (R-D) optimization: given a rate constraint, encoding parameters for the entire audio signal need to be chosen so that a psychoacoustically relevant distortion metric is minimized. As will

become obvious later, the complexity of a naïve search in the parameter space, for the optimal set, is exponential in the number of frames. To alleviate this problem we propose a dynamic programming-based two-layered trellis algorithm that jointly optimizes the choice of both inter- and intra-frame coding parameters via delayed decisions. Chapter 2 of this dissertation describes the proposed algorithm, along with requisite background information on AAC, and presents results, that demonstrate the superiority of the proposed encoder compared to standard algorithms.

Although the two-layered trellis guarantees the optimal parameter choices, optimality in terms of subjective quality critically depends on how well coding artifacts are captured in the distortion metric. The most commonly employed audio distortion metric is the noise-to-mask ratio (NMR) (more on this metric in Chapter 2). Based on observations from our R-D optimization approach to audio coding, we propose modifications to the NMR metric to enhance its subjective relevance. While this latter research on distortion metrics is orthogonal to the delayed source coding emphasis of this thesis, we include it in this dissertation, in part to indicate the practical importance of employing the right distortion metric in audio coding, and in part due to these modifications being inspired by the R-D optimization problem. Chapter 3 describes these modifications to the NMR metric, and the subjective improvements obtained when they are embedded into the encoder.

We note that although the emphasis in this thesis is on optimal encoding decisions for audio compression (where such delayed decisions are particularly useful), there has been considerable amount of prior work on incorporating encoding delay for decision making, in certain other areas of signal compression.

Notable amongst them include tree coding ([23], [48], etc.) and its application to speech coding ([5], [34], [86], etc.), trellis codes ([36], [58]), trellis coded quantization ([30], [57]) and its applications, for instance in image coding [33], [82], etc. Prior research on delayed decisions for audio compression is described in Chapter 2, where we contrast the proposed algorithm with such methods.

1.2 Optimal Delayed Decoding

In the case of decoding delay, we consider the scenario of predictive coding of autoregressive (AR) sources with a differential pulse code modulation (DPCM) scheme [27]. Although DPCM is a very simple predictive coding scheme, the operation of standard video compression algorithms, for instance, that employ inter-frame prediction can be cast into a DPCM setting. The DPCM encoder predicts the current sample from past reconstructed samples, and quantizes and encodes the prediction residual. Generally, the DPCM decoder on receipt of each index, *immediately* reconstructs the prediction error, and via prediction obtains the reconstruction of the corresponding sample, i.e., the decoder operates with zero delay. Such a predictive coder can be thought of as a sliding block code with infinite memory ($N_M = \infty$ and $K_M = \infty$), and zero look-ahead ($N_D = K_D = 0$). At very high bit-rates it can usually be argued that the prediction errors are almost the same as the innovations of the AR process, which form a sequence of independent random variables [27], [38], [41]. Hence the quantization indices too are almost independent, which implies that future indices provide no additional information on the current sample, and zero-delay decoding is optimal. But in practice bit-rates are limited and such arguments do not hold. Thus the

prediction errors, and indices, form a sequence of correlated random variables. In this case delay at the decoder can be exploited (i.e., a sliding block code with non-zero K_D can be employed) to improve the reconstruction of each sample.

Prior work such as [20], [77] that target delayed decoding of predictively encoded sources adopt a smoothing/filtering approach to the problem: the regular zero-delay reconstructions are just processed by a *non-causal* filter. In contrast to such ad hoc methods, we develop in this dissertation an estimation-theoretic (ET) optimal delayed decoder. This ET decoder recursively calculates the probability density function (pdf) of each sample, conditioned on all available past and future information, and obtains the optimal reconstruction via conditional expectation. The optimal delayed decoder in turn motivates an approximate decoder that employs a codebook or look-up table to obtain the delayed reconstructions, and hence is computationally efficient. In experiments, this codebook decoder is observed to have performance very close to that of the optimal decoder. Also presented are methods to curtail the storage needed for this codebook. Chapter 4 develops this theory of optimal delayed decoding, describes relevant prior work, and presents results in the setting of scalar AR sources.

As argued in the previous discussion, a sliding block code that incorporates future quantization indices finds utility only if these indices are sufficiently correlated. In audio codecs such as AAC, temporal correlations are exploited by time-to-frequency transforms, and transform coefficients from adjacent frames are somewhat less correlated. Therefore (inter-frame) prediction is infrequently employed in transform-based audio coders, and this delayed decoding approach is less relevant to such settings. Hence the reason for a different coding scenario from AAC of Sec. 1.1. Nevertheless, it needs to be emphasized that even the

simple DPCM setting considered here is very relevant in practice, and the efficacy of the methods proposed here has indeed been successfully demonstrated by implementation in the H.264 video decoder [39], although we exclude this latter work from the scope of this thesis. The proposed delayed decoding approaches are also useful in so called 'low-delay' or 'low-complexity' applications such as audio/speech coding for blue-tooth head-sets [1], image sensing [53], etc., where traditional transform-based coders are rejected in favor of predictive coders, due to the lower complexity (hence lower power requirements), and lesser encoding/decoding delays (compared to the bocking delay of transforms).

Chapter 2

Delayed Decision based Audio Compression

Audio compression has been fundamental to the success of many applications including streaming of music over the internet and hand-held music playback devices. Digital radio and gaming audio are other relatively new applications utilizing compressed audio. Most current audio coding techniques use psychoacoustic criteria to discard perceptually irrelevant information in the audio signal and achieve better compression. MPEG's AAC [43], [44], Sony's Adaptive Transform Acoustic Coder (ATRAC) [4], Lucent Technologies' Perceptual Audio Coder (PAC) [81], and Dolby's AC3 [28] are a few well known audio codecs. Descriptions of these coding techniques and general information regarding audio coding can be found in [72]. These techniques usually analyze the audio signal one frame or a small group of frames at a time and make encoding decisions on them, independently of other frames or frame-groups, thereby restricting encod-

ing delay. Restricted encoding delay enables real-time audio coding. But for the majority of audio coding applications, including those previously mentioned, compression is performed off-line. Hence the end user decodes pre-compressed audio and is not affected by any encoding delays. Moreover, encoding is a one time procedure while the coded audio is typically decoded many times. Thus, we propose here a coding technique that exploits encoding delay to make optimal decisions over the entire audio file, rather than processing each frame independently. The generated bitstream is standard compatible and decodable by standard decoder at no additional decoding delay.

As an example consider AAC (Fig. 2.1). The audio signal is split into overlapping frames. Depending on the stationarity of the signal, the framing is switched between a LONG window of 2048 samples and 8 SHORT windows of 256 samples each. Transition frames of suitable shape act as bridge windows between these configurations and this 'window switching' decision induces a one frame encoding delay. Subsequently, a time to frequency transformation is performed on the frame. The frequency domain coefficients are grouped into bands of unequal bandwidths to emulate the critical band structure of the human auditory system [88]. A psychoacoustic model provides masking thresholds for each of these bands, which determine the threshold of audibility of quantization noise in the bands. In AAC, a generic quantizer scaled by a parameter called the scale factor (SF) is used to quantize all the coefficients in the same band, and hence these bands are named scale factor bands (SFBs). The quantized coefficients in each SFB are then losslessly encoded using one of a prescribed set of Huffman code books (HCBs). Encoders try to find a set of SFs and HCBs that minimize a psychoacoustic distortion measure while satisfying a bit-rate constraint for the

frame. Though the target to be achieved maybe a particular mean bit-rate (average across frames) or file size, the instantaneous bit-rate, i.e., for individual frames, can fluctuate around this mean. This feature is generally implemented using a bit-reservoir technique wherein rate unused by frames of low demand is "saved" for use in later frames. Optional tools such as Temporal Noise Shaping and Perceptual Noise Substitution are not discussed here.

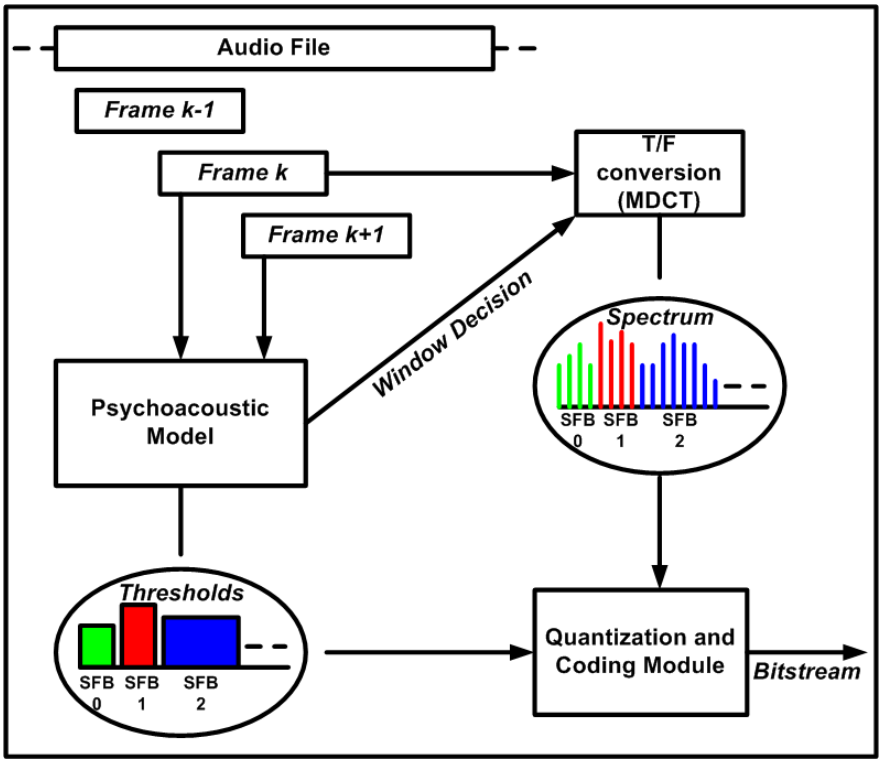


Figure 2.1. Schematic of a simple AAC encoder

The point to note is that the encoding procedure as described above makes decisions regarding each frame almost independently, with few minor exceptions: Due to window switching, the encoder encounters a delay of one frame to decide about transition windows. The bit-reservoir, in a limited sense, makes the encoding process dependent on past frames. But this encoding scheme, due to

its constrained delay, cannot foresee the demand for bits in future frames and deliberately save bits at some cost to the current frame. The drawbacks of this encoding procedure will be discussed in detail. For now, suffice it to say that constraining the encoding delay produces a bitstream of sub-optimal quality.

Thus there is merit in increasing encoding delay to search exhaustively over all combinations of encoding parameters, and choose the optimal set. But this may be computationally daunting. AAC, for example, provides a choice of 12 HCBs and nearly 60 SFs for each SFB. There are usually 49 SFBs in the LONG configuration and 56 SFBs for the 8 SHORT windows, although the exact number depends on other parameters such as sampling rate and short window grouping decisions [43], [44]. Including the choice of window configurations for each frame, a conservative estimate of such complexity would be $(2 \times (60 \times 12)^{49})^N$ for an audio file of N frames, i.e., exponential in the number of SFBs and frames. So it is desirable to pursue a dynamic programming [11] based approach with a corresponding trellis to search through these choices.

It is obvious that the search for the 'optimal' encoding parameters presupposes a criterion or distortion measure to compare the effects of various choices of these parameters. The most commonly used audio distortion measure is the NMR [14], [67], [68] - the ratio of quantization noise to masking threshold in each coding band (SFB in AAC). The distortion for a frame of audio and subsequently for the entire audio file is usually derived from the NMR. It should be noted that our methods are fairly general and could accommodate any additive distortion measure.

The problem of finding the optimal SFs and HCBs within an AAC frame (i.e., minimizing the frame distortion given a bit budget constraint) has been

previously addressed in earlier work of our research group [2] and [3], under the assumption of fixed bit-rate per frame, and that all frames were in the LONG configuration. Thus no decisions were delayed beyond the given frame. A low-complexity sub-optimal alternative was proposed in [87]. A mixed integer linear programming-based solution to the same problem was proposed by Bauer and Vinton in [8] and was extended to compare window decisions per frame in [7], where window decisions were independently performed for each frame, while neglecting dependence through transition windows. Bit-reservoir optimization, using a tree structured search, was proposed in [19], without optimization of window decisions or quantization and coding parameters. Rate-distortion optimal time segmentation of audio frames have been proposed in [13], [69], and [70] without optimization of parameters within a frame or distribution of bits across all frames.

We emphasize that we are, in fact, optimizing *all* the encoding decisions (window choice, SFs and HCBs as well as bit budget per frame) of the aforementioned simplistic AAC encoder. The eventual results show that there are significant gains over the reference encoder in terms of both objective metrics and subjective measures such as MOS scores within the MUSHRA test framework [46], and for a variety of audio samples drawn from the EBU-SQAM database [89]. The methods proposed are of higher complexity than the reference encoder but such complexity only impacts encoding which is typically an off-line operation, while the end-user does not experience any additional decoding delay. Results of this work have been reported in [61], [62], and [65].

The organization of this chapter is as follows. Sec. 2.1 provides additional background to the problem. The problem within the AAC setting is formulated

in Sec. 2.2. The two-layered trellis solution to the problem is described in Sec. 2.3. Sec. 2.4 summarizes the results.

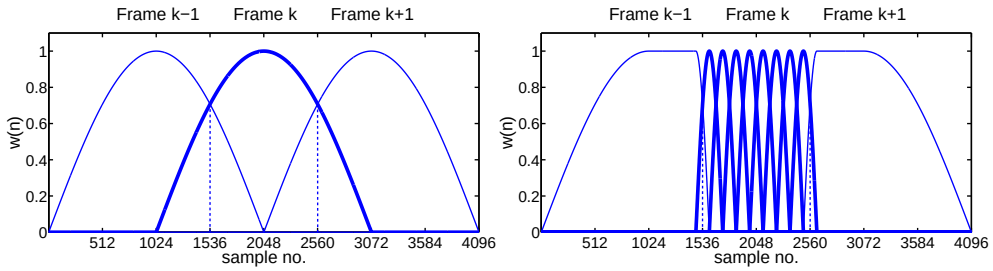
2.1 Background

2.1.1 MPEG Advanced Audio Coding

The implementation of the proposed approach is in the MPEG AAC setting. The high-level description of AAC given before is refined here with more details for the relevant blocks.

Window switching

The audio file is divided into overlapping frames and each frame is multiplied by a window. The frames are 2048 samples each in the LONG configuration (Fig. 2.2a). If the 1024 samples in the center of the frame (between the dotted lines of Frame k in Fig. 2.2a) are non-stationary, the frame is instead encoded as a series of 8 SHORT overlapped windows of 256 samples each (Frame k in Fig. 2.2b) to achieve better time resolution. Adjacent LONG and SHORT windows, due to their incompatible shapes, would disrupt the perfect reconstruction properties of the transform discussed further. This is prevented by replacing the LONG window preceding a series of SHORT windows with a START window of suitable shape (Frame $k-1$ in Fig. 2.2b) and the one succeeding a SHORT window with a STOP window (Frame $k+1$ in Fig. 2.2b). Window switching was first suggested for audio coding by Edler in [26]. Window switching decisions are usually made by the psychoacoustic model, based on heuristic thresholds of perceptual entropy



(a) Frame k in LONG configuration (b) Frame k in SHORT configuration

Figure 2.2. Frame k in LONG and SHORT configurations and corresponding effect on neighboring LONG frames

[49] or transient detection [90], [91].

Modified discrete cosine transform (MDCT)

Each audio frame is transformed to the frequency domain using the forward MDCT [55], [73], [79]. Despite requiring overlapped frames, the MDCT is critically sampled. MDCT of a LONG (also START and STOP) frame yields 1024 transformed coefficients and 128 coefficients for each SHORT block (or 1024 total for the 8 SHORT windows). More on the MDCT in Sec. 3.2.2.

Quantization and coding (QC) module

The quantization and coding module receives MDCT coefficients grouped into SFBs and corresponding masking thresholds from the psychoacoustic model, selects the SFs and HCBs, and quantizes and encodes the coefficients. The difference in SF values of consecutive SFBs is encoded using a single standard specified Huffman table. The HCB values are run-length coded, i.e., a fixed number of bits is used to convey the HCB value (whenever it changes from an SFB to the next),

and the number of consecutive SFBs having the same HCB. The SF and HCB bits thus consume part of the bit-rate and have to be accounted for in the rate calculation. In the MPEG verification model (VM) [90] the implicit rate-distortion trade-off is accomplished using a two loop search (TLS). The TLS inner loop is a distortion loop that searches through the set of SFs for each SFB such that a near-uniform target NMR is maintained across SFBs. Once this is achieved the encoder steps into the outer, rate loop, finds the best HCBs to encode the quantized spectra and calculates the total number of bits consumed by the frame. If the rate constraint for that frame is not met the target NMR is increased (to spend fewer bits), and the inner loop executed again.

Bit reservoir

AAC allows coding different frames with a different number of bits, though achieving a target average bit-rate might still be necessary. The VM implementation employs a bit-reservoir. If the QC module spends less than the available bit quota for the frame (e.g., when the frame corresponds to silence), excess bits may be used by future frames of higher demand.

2.1.2 Distortion measure

A distortion metric for audio coding should be able to properly account for the various perceptual artifacts caused by coding. Simple measures, such as the mean squared quantization error of the spectral coefficients, ignore psychoacoustic effects, while complicated metrics such as the Perceptual Evaluation of Audio Quality (PEAQ) [45], [83], entail intractable optimization complexity. The most

widely used metric is NMR [14], [18], [67], [68] which divides the squared quantization error in a coding band (SFB) by the band's masking threshold.

Consider a frame of AAC whose MDCT coefficients have been grouped into L SFBs. Let e_i be the squared quantization error of the coefficients in SFB i. Let μ_i be the reciprocal of the masking threshold in the band. The NMR, d_i, in SFB i is given by

$$d_i = \mu_i e_i, \quad 0 \leq i \leq L-1 \tag{2.1}$$

Several variants of the frame distortion can be derived from the above definition, for example, the total NMR (TNMR) denoted by D_T, is

$$D_T = \sum_{i=0}^{L-1} d_i \tag{2.2}$$

In [3], [7], [67], [68] the average NMR (ANMR), i.e., NMR averaged across SFBs has been used (clearly, $ANMR = D_T/L$). Since the number of SFBs varies for LONG and SHORT windows, TNMR is used in this work for a fair comparison between window configurations. Note that L in the SHORT configuration corresponds to the total number of SFBs of the 8 SHORT windows together. Alternatively, the distortion of a frame could be defined as the maximum NMR (MNMR) [3], [7], [68], [87], D_M, across all SFBs, i.e.,

$$D_M = \max_{i=0}^{L-1} d_i \tag{2.3}$$

Using the above as building blocks we can extend to consider distortion evaluation for the entire audio file (say of N frames):

$$Average\ TNMR\ (ATNMR): \qquad \mathcal{D}_{\mathcal{AT}} = \frac{1}{N}\sum_{k=0}^{N-1} D_T(k) \tag{2.4}$$

$$Maximum\ TNMR\ (MTNMR): \qquad \mathcal{D}_{\mathcal{MT}} = \max_{k=0}^{N-1} D_T(k) \tag{2.5}$$

$$Maximum\ MNMR\ (MMNMR): \qquad \mathcal{D}_{\mathcal{MM}} = \max_{k=0}^{N-1} D_M(k) \tag{2.6}$$

$D_T(k)$ and $D_M(k)$ denote the distortion of frame k according to TNMR of (2.2) and MNMR of (2.3), respectively. It is important to note that there is no single audio distortion measure that is known to capture well, all artifacts produced by restricted bit-rate audio coding and the consideration of all the above candidates will demonstrate the generality of the proposed approach.

2.1.3 Problem motivation and challenges

Window switching

As already mentioned, current encoders rely on heuristics to make decisions about window switching. But such decisions are not optimal in the sense of minimizing a pre-specified distortion measure. One approach (see [7] and [13]) is to design an encoder that compares the frame distortion under different window configurations and makes a window choice for that frame. But different windows encompass a different number of samples, as is evident in Fig. 2.2, and such comparison would not be fair. In addition, two consecutive frames cannot independently be encoded as a LONG-SHORT pair and thus, independent window choices for each frame may not form an 'allowable' window sequence. One could, on the other hand, compare distortion in two sequences of window decisions which start and end in the same audio samples, for instance, the LONG-LONG-LONG sequence of Fig. 2.2a and START-SHORT-STOP sequence of Fig. 2.2b. This of course entails delay. This simple example provides motivation for investigating delayed decisions for window switching.

Bit reservoir

The bit-reservoir of VM allows a frame to utilize bits saved (i.e., unused) in the past but cannot "borrow from the future". Nor can it optimally borrow from the past, as the encoder cannot anticipate future needs. Some encoders, including 3GPP's Enhanced AACplus [91] encoder, intentionally save some bits for future use by employing perceptual entropy based algorithms that specify the bit requirement for a frame. Such algorithms involve heuristic thresholds. Fig. 2.3 compares the effect on distortion (TNMR) due to the distribution of bit resource according to VM versus MTNMR minimization by the delayed-decision approach discussed later. The spikes in TNMR values for VM correspond to artifacts caused by a lack of sufficient bits in non-stationary frames of the audio sample (glockenspiel). It is evident that delayed decision redistributes bits to mitigate such coding artifacts.

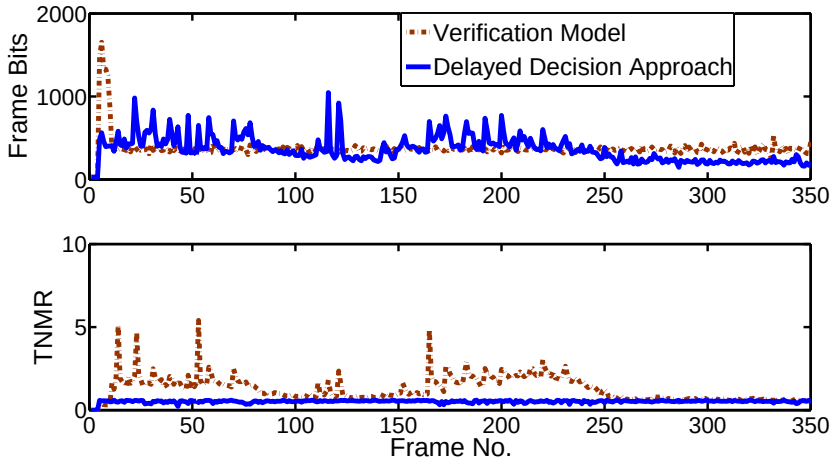


Figure 2.3. Distribution of rate and distortion (TNMR) across frames when using the VM and delayed-decision based approach for glockenspiel at 16 kbps

Quantization and coding module

TLS, as described previously, separates the calculation of rate and distortion into individual loops and does not simultaneously control them. Moreover, SFs for consecutive SFBs are differentially encoded, and HCBs are run length encoded. Hence, selecting these parameters for each band independently is sub-optimal. The trellis-based optimal parameter selection of [2] and [3] is a rate-distortion optimal alternative to TLS. But the procedure there was based on the assumption that the bit-rate for each frame was fixed. Modifications are necessary to incorporate this trellis into a system that relies on delayed decisions for distributing bits to frames. Another limiting assumption was that all windows were encoded in the LONG configuration. Modifications are also necessary to jointly deal with 8 SHORT frames.

2.2 Joint Selection of Encoding Parameters: Problem Formulation

We describe here the problem formulation in the AAC setting.

2.2.1 Problem setting

Consider an audio file of N frames. Frame k ($0 \leq k \leq N-1$) is associated with a window configuration w_k from the set $\{\ LONG,\ START,\ SHORT,\ STOP\ \}$. The number of SFBs L_k in frame k depends on the window configuration. In the SHORT configuration, L_k corresponds to the number of SFBs of all 8 SHORT

windows. SFB i of frame k is associated with a scalefactor s_i^k and Huffman code book h_i^k ($0 \leq i \leq L_k - 1$). Parameters s_i^k and h_i^k take value in finite sets of SF and HCB choices as prescribed in the AAC standard. Thus the intra-frame decisions produce L_k-tuples $S_k = (s_0^k, \ldots, s_{L_k-1}^k)$ and $H_k = (h_0^k, \ldots, h_{L_k-1}^k)$. All the above encoding parameters for a frame are summarized in $P_k = (w_k, S_k, H_k)$. Additionally, we denote by $\mathbf{x}_k$ the segment of 2048 audio samples encompassed by frame k in the LONG configuration. Clearly, other window configurations use a subset of $\mathbf{x}_k$.

The number of bits of information representing frame k depends on the actual samples it contains and the choice of encoding parameters and is, hence, denoted by $B(\mathbf{x}_k, P_k)$. An average rate constraint $\mathcal{R}$ is imposed on the encoding process, i.e.,

$$\frac{1}{N} \sum_{k=0}^{N-1} B(\mathbf{x}_k, P_k) \leq \mathcal{R} \tag{2.7}$$

The window decisions sequence is also constrained so that a START window is always used when transitioning from a LONG to a SHORT window, and a STOP window is inserted between SHORT and LONG windows. These conditions will be referred to as the ***Window Switching Constraints***.

2.2.2 Rate and distortion calculation

The information, in the bitstream, about SFB i of frame k can be summarized as follows:

- We denote by $\mathcal{Q}(\mathbf{x}_k, w_k, s_i^k, h_i^k)$ the number of bits needed to encode the spectral coefficients in SFB i, as it naturally depends on the audio samples in the frame, $\mathbf{x}_k$, in addition to the quantizer (scalefactor s_i^k), the huffman

code book h_i^k and the window choice w_k (which influences the transform applied on $\mathbf{x}_k$ and hence the unquantized spectral coefficient values).

- The scalefactor s_i^k is transmitted as $s_i^k - s_{i-1}^k$. Therefore the scalefactor bits for SFB i can be written as $\mathcal{E}(s_{i-1}^k, s_i^k)$ (with $s_{-1}^k = 0$).

- The run-length encoding of HCBs produces a fixed number of bits to indicate the run-length whenever $h_i^k \neq h_{i-1}^k$ and 0 bits otherwise. Thus the number of HCB information bits for SFB i is of the form $\mathcal{F}(h_{i-1}^k, h_i^k)$ (with $h_{-1}^k \neq h_0^k$).

Additionally, the encoder conveys the window configuration using $\mathcal{G}(w_k)$ bits. Thus the total number of bits to encode the frame with parameters P_k can be enumerated as,

$$B(\mathbf{x}_k, P_k) = \mathcal{G}(w_k) + \sum_{i=0}^{L_k-1} \left\{ \mathcal{Q}(\mathbf{x}_k, w_k, s_i^k, h_i^k) + \mathcal{E}(s_{i-1}^k, s_i^k) + \mathcal{F}(h_{i-1}^k, h_i^k) \right\} \quad (2.8)$$

where the number of SFBs L_k depends on w_k.

The psychoacoustic model produces a masking threshold for each SFB of a frame by analyzing it in the frequency domain. Thus, the weight μ_i in (2.1) is a function of the audio signal $\mathbf{x}_k$ and the transform (and hence w_k) used for time to frequency conversion. Similarly, the squared quantization error e_i depends on the quantizer (i.e., scalefactor s_i^k) and the unquantized transform coefficients. Thus, using (2.1), the distortion d_i in SFB i of frame k can be represented as,

$$d_i(\mathbf{x}_k, w_k, s_i^k) = \mu_i(\mathbf{x}_k, w_k) e_i(\mathbf{x}_k, w_k, s_i^k) \quad (2.9)$$

The above definition of d_i is subsequently used in (2.2) or (2.3) to obtain the frame distortion. In either case we employ the generic notation $D(\mathbf{x}_k, P_k)$, where

it is clear from the context whether $D_T(k)$ or $D_M(k)$ is in use. The distortion of the entire file is then obtained from (2.4) - (2.6). Let the encoding parameter set for the entire file be $\mathcal{P} = (P_0, \ldots, P_{N-1})$, while $\mathcal{X}$ represents the entire audio signal itself. The overall distortion, therefore, can be denoted as $\mathcal{D}(\mathcal{X}, \mathcal{P})$, and the overall bit consumption is given by

$$\mathcal{B}(\mathcal{X}, \mathcal{P}) = \sum_{k=0}^{N-1} B(\mathbf{x}_k, P_k) \tag{2.10}$$

Note that H_k is specified in P_k and needed to determine the rate, but it plays no role in determining the value of $D(\mathbf{x}_k, P_k)$, as is evident from (2.9).

2.2.3 Problem definition

Find the parameter set $\mathcal{P}^*$ that minimizes the overall distortion, i.e.,

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \ \mathcal{D}(\mathcal{X}, \mathcal{P}) \tag{2.11}$$

subject to the rate constraint $\frac{1}{N}\mathcal{B}(\mathcal{X}, \mathcal{P}) \leq \mathcal{R}$ and the window switching constraints of Sec. 2.2.1.

Depending on the choice of definition of $\mathcal{D}(\mathcal{X}, \mathcal{P})$ from (2.4) - (2.6) we have three different problems which will be referred to as the ***ATNMR***, ***MTNMR***, and ***MMNMR problems***, respectively.

2.3 Optimization with a Two-Layered Trellis

2.3.1 Minimizing average overall distortion

We address here the problem of minimizing the average distortion of the file,

$$\mathcal{D}(\mathcal{X}, \mathcal{P}) = \frac{1}{N} \sum_{k=0}^{N-1} D(\mathbf{x}_k, P_k) \tag{2.12}$$

given the rate constraint (2.7). Note that if $D(\mathbf{x}_k, P_k)$ is defined as TNMR (2.2) then $\mathcal{D}(\mathcal{X}, \mathcal{P})$ would be ATNMR (2.4). The above problem is similar to the classical problem of minimizing average distortion of quantizers given a rate constraint. The problem was originally addressed for independent quantizers in [80] and later for dependent quantizers in [74] using a Langrangian based iterative procedure. The constrained optimization problem is converted to that of minimizing the Lagrangian cost,

$$\mathcal{J}_A(\mathcal{X}, \mathcal{P}) \;\; = \;\; \mathcal{D}(\mathcal{X}, \mathcal{P}) + \lambda \frac{1}{N} \mathcal{B}(\mathcal{X}, \mathcal{P}), \tag{2.13}$$

where λ is the Lagrange parameter. Rewriting (2.13) as a summation over frames we obtain,

$$\mathcal{J}_A(\mathcal{X}, \mathcal{P}) = \sum_{k=0}^{N-1} J_A(\mathbf{x}_k, P_k) \tag{2.14}$$

where,

$$J_A(\mathbf{x}_k, P_k) = \frac{1}{N} \{ D(\mathbf{x}_k, P_k) + \lambda B(\mathbf{x}_k, P_k) \} \tag{2.15}$$

is the contribution of a particular frame to the Lagrangian cost. Minimization of $\mathcal{J}_A(\mathcal{X}, \mathcal{P})$ for a specific value of λ yields an operating point on the rate-distortion curve. One may adjust λ and re-optimize until the rate constraint is satisfied, to obtain the choice of parameters $\mathcal{P}^* = (P_0^*, \ldots, P_{N-1}^*)$ that minimize the distortion

in (2.12) under the constraint (2.7). Note that $J_A(\mathbf{x}_k, P_k)$, the Lagrangian cost for frame k, is independent of encoding decisions $P_l,\ l \neq k$ and therefore,

$$\min_{\mathcal{P}} \mathcal{J}_A(\mathcal{X}, \mathcal{P}) = \sum_{k=0}^{N-1} \min_{P} J_A(\mathbf{x}_k, P) \tag{2.16}$$

where $P = (w, S, H)$ is a generic point in the encoding parameter space for a single frame. Thus, for a given value of λ, the overall minimization problem seems separable into N intra-frame minimization problems. Note, however, that $P_k = (w_k, S_k, H_k)$ depends on the window choice. Independent minimization of $J_A(\mathbf{x}_k, P_k)$ over all window choices may violate the window switching constraints and yield incompatible windows for neighboring frames, as discussed in Sec. 2.1.3. To circumvent this difficulty we define the minimum frame Lagrangian for a given window configuration w as,

$$J_k^*(w) = \min_{S,H} J_A(\mathbf{x}_k, \{w, S, H\}), \tag{2.17}$$
$$\forall w \in \{LONG,\ START,\ SHORT,\ STOP\}$$

The dependence of $J_k^*(\cdot)$ on $\mathbf{x}_k$ is implicit in the subscript k. The above minimization which will henceforth be referred to as the **Intra-frame Minimization Problem I** is discussed in Sec. 2.3.3. Assume for now that for every frame k the above minimum cost $J_k^*(w)$, the minimizing parameters $S_k^*(w)$ and $H_k^*(w)$, corresponding distortion $D_k^*(w)$ and frame bit consumption $B_k^*(w)$ have been calculated for every window configuration w. The overall cost $\mathcal{J}_A$ is, therefore, minimized by the window decisions $w_0^*, \ldots, w_{N-1}^*$ given by,

$$(w_0^*, \ldots, w_{N-1}^*) = \arg \min_{(w_0,\ldots,w_{N-1})} \sum_{k=0}^{N-1} J_k^*(w_k) \tag{2.18}$$

with $(w_0, \ldots, w_{N-1})$ obeying the window switching constraints (Sec. 2.2.1). The search complexity of the above problem can be reduced drastically while simultaneously imposing these constraints by using a *trellis*-based search, such as the

Viterbi algorithm [31], [84]. A trellis (the Outer Trellis in Fig. 2.4) is con-

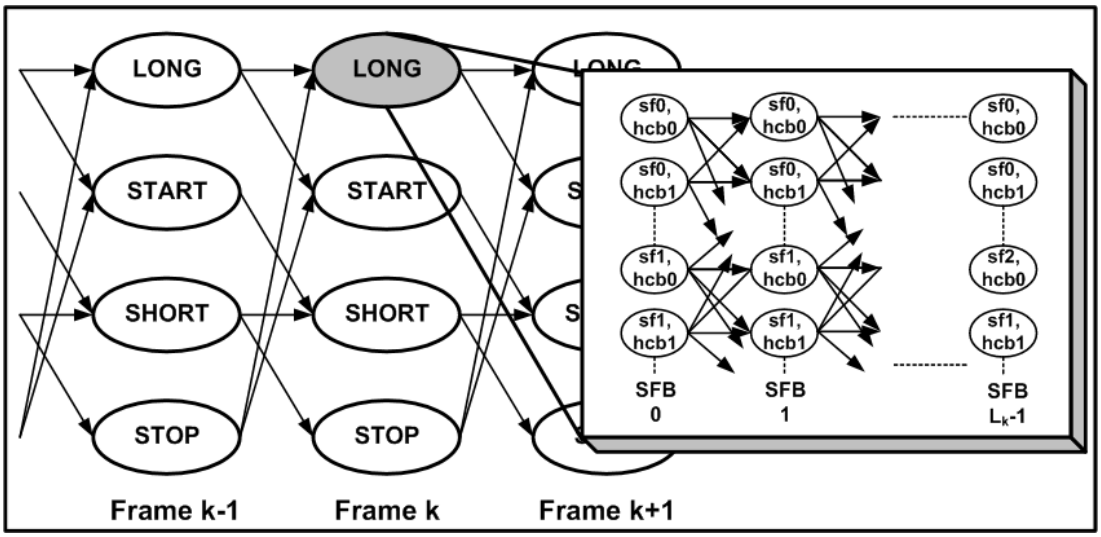


Figure 2.4. **Two-Layered Trellis:** The *Window Switching Trellis* (or *Outer Trellis*) runs across frames, with states as window choices. The *Inner Trellis* (in the inset) spans across SFBs and is used in each node of the Outer Trellis to find the best intra-frame parameters.

structed with stages corresponding to frames and nodes to window choices per frame. Transitions are allowed only between compatible window choices, e.g., LONG to LONG, LONG to START, etc. Each node is associated with a specific window decision w and is populated with corresponding quantities $J_k^*(w)$, $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, and $B_k^*(w)$. The solution $w_0^*, \ldots, w_{N-1}^*$ to (2.18) is the path $(w_0, \ldots, w_{N-1})$ through the trellis that minimizes the total cost $\sum_{k=0}^{N-1} J_k^*(w_k)$ along that path.

To formally implement the window switching constraints, associate the window configurations LONG, START, SHORT, and STOP with the numbers 1-4, respectively. We denote by $\mathbf{W}_m$, $1 \leq m \leq 4$, the set of window choices which

could precede the window choice m. For example, $\mathbf{W}_1 = \{1, 4\}$ - a LONG window can only be preceded by a LONG or STOP window. The path of minimum cost is found as follows:

Outer Trellis Algorithm

1. *Initialize.* For $1 \leq m \leq 4$, set partial sum $\Upsilon(m) = J_0^*(m)$. Set counter $k = 1$.

2. *Search.* For $1 \leq m \leq 4$, in stage k, find back pointer

$$\Psi_k(m) = \arg \min_{n \in \mathbf{W}_m} \Upsilon(n) \ .$$

3. *Update.* For $1 \leq m \leq 4$, set partial sum $\Upsilon(m) = \Upsilon(\Psi_k(m)) + J_k^*(m)$.

4. *Next Stage.* Increment k. If $k < N$ go to step 2.

5. *BackTrack.* Winning path ends in $w_{N-1}^* = \arg\min_{1 \leq m \leq 4} \Upsilon(m)$. Set $k = N - 1$. While $k \neq 0$, do $\{w_{k-1}^* = \Psi_k(w_k^*),\ k = k - 1\}$.

At each stage, only 4 paths survive and the complexity of this search is linear in N. As is evident, the trellis search naturally incorporates the window switching constraints, hence the name Window Switching Trellis. It is also called the Outer Trellis to differentiate from the Inner Trellis (inset of Fig. 2.4) that will be used to solve (2.17). If the rate $\frac{1}{N} \sum_{k=0}^{N-1} B_k^*(w_k^*)$ associated with the winning path does not satisfy the rate constraint (2.7), λ is adjusted, the minimization of (2.17) redone for each frame and in all window configurations, the outer trellis re-populated, and the above search repeated. When the rate constraint is met the decisions associated with the winning path are the optimal decisions minimizing the overall distortion given by (2.12).

2.3.2 Minimizing maximum overall distortion

Here,

$$\mathcal{D}(\mathcal{X}, \mathcal{P}) = \max_{k=0}^{N-1} D(\mathbf{x}_k, P_k) \tag{2.19}$$

Depending on whether $D(\mathbf{x}_k, P_k)$ is defined according to TNMR (2.2) or MNMR (2.3), the resulting $\mathcal{D}(\mathcal{X}, \mathcal{P})$ will be either MTNMR (2.5) or MMNMR (2.6). A Lagrangian solution is not applicable here due to the min-max nature of the problem. Nevertheless, a trellis-based approach offers an effective means to find the solution. Let parameter γ specify the maximum overall distortion:

$$\begin{aligned} & \mathcal{D}(\mathcal{X}, \mathcal{P}) \;\leq\; \gamma \\ \Rightarrow \qquad & D(\mathbf{x}_k, P_k) \;\leq\; \gamma,\ 0 \leq k \leq N-1 \end{aligned} \tag{2.20}$$

We now find the set of encoding parameters $\mathcal{P}^*$ that minimizes the total rate $\frac{1}{N}\mathcal{B}(\mathcal{X}, \mathcal{P})$ subject to the above distortion constraint, i.e., the cost function to be minimized is,

$$\begin{aligned} \mathcal{J}_M(\mathcal{X}, \mathcal{P}) &= \frac{1}{N}\mathcal{B}(\mathcal{X}, \mathcal{P}) \\ &= \sum_{k=0}^{N-1} J_M(\mathbf{x}_k, P_k) \end{aligned} \tag{2.21}$$

where $J_M(\mathbf{x}_k, P_k) = \frac{1}{N}B(\mathbf{x}_k, P_k)$ is the corresponding cost function for frame k. If the rate thus found exceeds the rate constraint in (2.7), γ can be increased (allow more distortion in each frame) and the minimization repeated. Thus we now iterate over γ, similar to the iteration over λ in Sec. 2.3.1. We can again split the overall minimization into N separate minimizations as below.

$$\min_{\substack{\mathcal{P}\ s.t. \\ \mathcal{D}(\mathcal{X}, \mathcal{P}) \leq \gamma}} \mathcal{J}_M(\mathcal{X}, \mathcal{P}) = \sum_{k=0}^{N-1} \min_{\substack{P\ s.t. \\ D(\mathbf{x}_k, P) \leq \gamma}} J_M(\mathbf{x}_k, P) \tag{2.22}$$

where we have used (2.20). The window switching constraints again forbid independent minimization. Thus the corresponding minimum cost for a frame in window configuration w is defined as,

$$J_k^*(w) = \min_{\substack{S,\,H\ s.t. \\ D(\mathbf{x}_k,\,P)\,\leq\,\gamma}} J_M(\mathbf{x}_k, \{w, S, H\}) \qquad (2.23)$$
$$\forall w \in \{LONG,\ START,\ SHORT,\ STOP\}$$

The above minimization is referred to as **Intra-frame Minimization Problem II** and will be discussed in Sec. 2.3.4 which derives the optimal cost $J_k^*(w)$ and corresponding $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, and $B_k^*(w)$ for populating the Window Switching Trellis. The Outer Trellis Algorithm of Sec. 2.3.1 finds the best path (decisions) through the trellis. The rate can be adjusted by varying γ, repeating the minimization of (2.23), re-populating the trellis, and finding the winning path again.

It should be noted that in Sec. 2.3.1 and Sec. 2.3.2 the best path is decided at the end of the Window Switching Trellis, thereby clearly implementing delayed decisions. Additional delay is due to iterations over λ or γ values, but such delay can be substantially contained by complexity reduction techniques to be discussed later.

2.3.3 Intra-frame minimization problem I

In Sec. 2.3.1 we assumed that the solution to (2.17) is available. The problem is rewritten here in equivalent form: for frame k, in a specific window configura-

tion w, we need to find,

$$\Big\{S_k^*(w), H_k^*(w)\Big\} = \arg\min_{S,H} \frac{1}{N}\Big\{D(\mathbf{x}_k, \{w, S, H\}) + \lambda B(\mathbf{x}_k, \{w, S, H\})\Big\} \quad (2.24)$$

The solution entails a search over all possible combinations of SFs and HCBs, a space whose cardinality is exponential in the number of SFBs. Based on [2] and [3], $S_k^*(w)$ and $H_k^*(w)$ can be obtained in a computationally efficient manner when the frame distortion $D(\mathbf{x}_k, P)$ is defined as TNMR or MNMR calculated over the SFBs. In the former case we specifically write

$$D(\mathbf{x}_k, P) = \sum_{i=0}^{L_k-1} d_i(\mathbf{x}_k, w, s_i) \quad (2.25)$$

This in conjunction with (2.8) and (2.24) and noting that $\mathcal{G}(w_k)$ of (2.8) is independent of S_k and H_k yields,

$$\Big\{S^*(w), H^*(w)\Big\} = \arg\min_{S,H} \sum_{i=0}^{L-1} \Big\{ \begin{array}{r} d_i(w, s_i) + \lambda\Big(\mathcal{Q}(w, s_i, h_i) \\ +\mathcal{E}(s_{i-1}, s_i) + \mathcal{F}(h_{i-1}, h_i)\Big) \end{array} \Big\} \quad (2.26)$$

where the frame index k is implicit and the dependence on the deterministic audio segment $\mathbf{x}_k$ has been omitted to simplify notation. The above minimization can be realized using the Inner Trellis of Fig. 2.4 which has SFBs as stages and states corresponding to combination of SF and HCB values. Thus each state of stage i (SFB i) can be indexed by an ordered pair (u, v) denoting $s_i = u$ and $h_i = v$, associated with distortion $d_i(w, u)$ and quantization bits $\mathcal{Q}(w, u, v)$. A transition from state (u', v') in stage $i-1$ to state (u, v) in stage i is associated with the rate costs $\mathcal{E}(u', u)$ and $\mathcal{F}(v', v)$ to encode (s_i, h_i). A path through this trellis corresponds to SF and HCB sequences S and H, respectively. We seek the path that minimizes the cost in (2.26). We define the cost for a node (u, v) in stage i as

$$\Pi_i(u, v) = d_i(w, u) + \lambda\mathcal{Q}(w, u, v) \quad (2.27)$$

and for transition (u', v') of stage $i-1$ to (u, v) of stage i as

$$\Delta_i\left((u', v') \rightarrow (u, v)\right) = \lambda\left(\mathcal{E}(u', u) + \mathcal{F}(v', v)\right) \tag{2.28}$$

The path of minimum cost is found as follows:

Inner Trellis Algorithm

1. *Initialize.* $\forall(u, v)$ partial cost $\Gamma(u, v) = \Pi_0(u, v) + \Delta_0\left((u', v') \rightarrow (u, v)\right)$ with $u' = 0$ and $v' \neq v$ being forced (Sec. 2.2.2). Set $i = 1$.

2. *Search.* $\forall(u, v)$ of stage i find back pointers

$$\Theta_i(u, v) = \arg \min_{(u', v') \ in \ stage \ i-1} \left\{\Gamma(u', v') + \Delta_i\left((u', v') \rightarrow (u, v)\right)\right\} .$$

3. *Update.* $\forall(u, v)$ update partial cost

$$\Gamma(u, v) = \Gamma(\Theta_i(u, v)) + \Delta_i\left(\Theta_i(u, v) \rightarrow (u, v)\right) + \Pi_i(u, v) .$$

4. *Next Stage.* Increment i. If $i < L$ go to step 2.

5. *Back Track.* Winning path ends in

$$(s^*_{L-1}, h^*_{L-1}) = \arg \min_{(u,v) \ in \ stage \ L-1} \Gamma(u, v)$$

Set $i = L - 1$. While $i \neq 0$, do $\{(s^*_{i-1}, h^*_{i-1}) = \Theta_i(s^*_i, h^*_i),\ i = i - 1\}$.

In step 2 of the above algorithm, only one path into any state survives and thus after each stage there are as many paths as states. Hence the complexity of the above algorithm is linear in the number of SFBs. The algorithm when performed for frame k in window configuration w, gives the best SF and HCB sequence $S^*_k(w)$, $H^*_k(w)$ in (2.24), and corresponding distortion $D^*_k(w)$. The cost and rate

associated with the winning path in the above algorithm, in conjunction with the contribution from $\mathcal{G}(w)$ of (2.8) give $B_k^*(w)$ and $J_k^*(w)$ of (2.17) used in the outer trellis of Sec. 2.3.1.

ATNMR solution: Using the above algorithm in tandem with Sec. 2.3.1 we can now enumerate a *Two-Layered Trellis*-based solution to the ATNMR problem (Sec. 2.2.3):

1. *Initialize.* Select a value of Lagrangian parameter λ.

2. *Inner Trellis.* For each frame k and in each window configuration w, using the Inner Trellis Algorithm and node and transition costs as defined in (2.27) and (2.28), respectively, find $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, $J_k^*(w)$, and $B_k^*(w)$ and populate the outer trellis.

3. *Outer Trellis.* Using the Outer Trellis Algorithm find the best window decisions $w_0^*, \ldots, w_{N-1}^*$ and consequently $P_k^* = (w_k^*, S_k^*(w_k^*), H_k^*(w_k^*))\ \forall k$, overall rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$, and distortion $\mathcal{D}(\mathcal{X}, \mathcal{P}^*)$.

4. *Iterate.* Check rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$ against rate constraint. If satisfied go to step 5, else change λ and go to step 2.

5. *Encode.* Use the optimal parameter set $\mathcal{P}^*$ to encode the audio file.

2.3.4 Intra-frame minimization problem II

We address here the minimization problem in (2.23), i.e., for frame k, in window configuration w_k,

$$\left\{S_k^*(w), H_k^*(w)\right\} = \arg \min_{\substack{S,\,H \\ D(\mathbf{x}_k,\,P)\,\leq\,\gamma}} \frac{1}{N} B(\mathbf{x}_k, \{w, S, H\}) \tag{2.29}$$

As in Sec. 2.3.3, a computationally efficient minimization is possible if the frame distortion $D(\mathbf{x}_k, P)$ is in the form of sum or maximum of SFB distortions. We describe the solution here for the maximum case, i.e.,

$$D(\mathbf{x}_k, P) = \max_{i=0}^{L_k-1} d_i(\mathbf{x}_k, w, s_i) \tag{2.30}$$

Combined with the distortion constraint in (2.29) it implies that

$$d_i(\mathbf{x}_k, w, s_i) \leq \gamma \ , \qquad \forall i \ . \tag{2.31}$$

Using (2.8) and (2.31), we can now rewrite (2.29) as

$$\left\{S^*(w), H^*(w)\right\} = \arg \min_{\substack{S,\,H \\ d_i(w,\,s_i)\,\leq\,\gamma\ \forall i}} \sum_{i=0}^{L-1} \left(\mathcal{Q}(w, s_i, h_i) + \mathcal{E}(s_{i-1}, s_i) + \mathcal{F}(h_{i-1}, h_i)\right) \tag{2.32}$$

where, as usual, we omit index k, the dependence on $\mathbf{x}_k$, and the term $\mathcal{G}(w)$. We use the same inner trellis as in Sec. 2.3.3 to perform the minimization of (2.32) but the node and transition costs (2.27), (2.28) are redefined as,

$$\Pi_i(u, v) = \begin{cases} \mathcal{Q}(w, u, v) & if\ d_i(w, u) \leq \gamma \\ \infty & otherwise \end{cases} \tag{2.33}$$

$$\Delta_i\left((u', v') \rightarrow (u, v)\right) = \mathcal{E}(u', u) + \mathcal{F}(v', v) \tag{2.34}$$

The Inner Trellis Algorithm described in Sec. 2.3.3 can be subsequently used to find $S_k^*(w)$, $H_k^*(w)$ of (2.29), the corresponding distortion $D_k^*(w)$ as well as the

rate cost of the winning path. This, along with $\mathcal{G}(w)$ of (2.8) gives the minimum cost $J_k^*(w)$ of (2.23) and can be used in the outer trellis method of Sec. 2.3.2.

MMNMR solution: We can now solve the MMNMR problem using the above algorithm and the method described in Sec. 2.3.2, in a *Two-Layered Trellis* framework.

1. *Initialize.* Select a value of the maximum distortion parameter γ.

2. *Inner Trellis.* For each frame k and in each window configuration w, using the Inner Trellis Algorithm with node and transition costs of (2.33) and (2.34), find $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, $J_k^*(w)$, and $B_k^*(w)$ and populate the outer trellis.

3. *Outer Trellis.* Using the Outer Trellis Algorithm find the optimal window decisions $w_0^*, \ldots, w_{N-1}^*$ and consequently $P_k^* = (w_k^*, S_k^*(w_k^*), H_k^*(w_k^*))\ \forall k$, overall rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$, and distortion $\mathcal{D}(\mathcal{X}, \mathcal{P}^*)$.

4. *Iterate.* Check rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$ against the rate constraint. If satisfied go to step 5, else change γ suitably and go to step 2.

5. *Encode.* Use decisions $\mathcal{P}^*$ to encode the audio file.

The MTNMR problem, a hybrid of maximum and cumulative distortions, requires the solution of (2.23) but with the frame distortion $D(\mathbf{x}_k, P)$ being the sum (TNMR) of SFB distortions. Therefore (2.23) can be seen as equivalent to finding parameters that minimize the rate $B(\mathbf{x}_k, P)$ given a constraint on a cumulative distortion criterion. This is a dual of the problem where the rate for a frame is fixed and parameters that minimize average (or total) distortion have

to be found [2], [3], [7], [8] and can still be solved using the Lagrangian approach described in Sec. 2.3.3.

MTNMR solution:

1. *Initialize.* Select a value of the maximum distortion parameter γ.

2. *Inner Trellis.* For each frame k and in each window configuration w do the following:

   (a) Select a value of intra-frame Lagrangian parameter λ_{inner}.

   (b) Using the Inner Trellis Algorithm with cost definitions (2.27) and (2.28) and setting $\lambda = \lambda_{inner}$ find $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, $J_k^*(w)$, and $B_k^*(w)$.

   (c) Check $D_k^*(w)$ against γ. If satisfied go to step (d) else change λ_{inner} and go to step (a).

   (d) Populate the corresponding outer trellis node with $S_k^*(w)$, $H_k^*(w)$, $D_k^*(w)$, $J_k^*(w)$, and $B_k^*(w)$.

3. *Outer Trellis.* Using the Outer Trellis Algorithm find the best window decisions $w_0^*, \ldots, w_{N-1}^*$, $P_k^* = (w_k^*, S_k^*(w_k^*), H_k^*(w_k^*))$ $\forall k$, overall rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$, and distortion $\mathcal{D}(\mathcal{X}, \mathcal{P}^*)$.

4. *Iterate.* Check rate $\mathcal{B}(\mathcal{X}, \mathcal{P}^*)$ against the rate constraint. If satisfied go to step 5 else change γ suitably and go to step 2.

5. *Encode.* Use decisions $\mathcal{P}^*$ to encode the audio file.

Note: If γ, the allowed distortion in each frame, is too small, it is possible that no choice of parameter sets S and H achieves it, i.e., the parameter space for the

minimization in (2.23) could be a null set for certain frames in particular window configurations w. In such a case, $D_k^*(w)$ in step 2(c) of above algorithm will not be less than γ for any value of λ_{inner} and, unless fixed, results in an infinite loop. This pathology can be avoided by including an appropriate exit condition in the program. For example, it is easily seen that a low value of λ_{inner} favors decreasing distortion $D_k^*(w)$ at the cost of increasing rate $B_k^*(w)$. So λ_{inner} could be bound to be greater than a minimum value ζ. If the distortion $D_k^*(w) > \gamma$ in step 2(c) even if $\lambda_{inner} = \zeta$, then a forced exit is made from step 2(c) with the cost $J_k^*(w)$ being explicitly set to ∞.

2.3.5 Modifications for SHORT configuration

The SHORT window configuration requires some modifications to the inner trellis design of [2] or [3]. The 8 SHORT windows in the frame must be encoded jointly, i.e., the QC module (the inner trellis) analyzes the SFBs of all 8 windows and jointly determines their SFs and HCBs. Let L_s denote the number of SFBs per SHORT window. The AAC bitstream format dictates that the information regarding the L_s SFBs of the first SHORT window appear first, followed by that of the second and so on. Note that both differential encoding of SFs and run length encoding of HCBs requires the imposition of ordering on the SFBs. The AAC standard allows differential encoding of SFs across SHORT window boundaries within a frame (e.g., the SF of the first SFB in the second SHORT window may be encoded as a difference from that of the last SFB in the first SHORT window), but it restricts run length coding of HCBs from extending beyond the SHORT window boundary. Therefore, the inner trellis has $8L_s$ stages, corresponding to the SFBs of all 8 SHORT windows. Transition costs ((2.28), (2.34)) which

straddle across SFBs of two adjacent SHORT windows are allowed the usual SF contribution of $\mathcal{E}(s_{i-1}, s_i)$ but artificially forced to have a non-zero $\mathcal{F}(h_{i-1}, h_i)$ contribution even if $h_{i-1} = h_i$ (See Sec. 2.2.2).

Additionally, the AAC standard allows 'grouping of SHORT windows' where the encoder can identify consecutive SHORT windows within a frame with similar characteristics and interleave their spectra into a shared set of SFBs [43], [44]. For example, a frame of 8 SHORT windows could be partitioned into three groups of 2, 3 and 3 windows. Windows in the same group share SFs and HCBs for the same SFB. This is accommodated in the inner trellis by using stages as grouped SFBs rather than individual window SFBs.

Since there are 8 windows, 127 groupings are possible and the grouping choice is an additional encoding parameter in the SHORT configuration. But all of these groupings span the same number of audio samples and hence the minimizations in (2.17) and (2.23) can be performed in each grouping configuration to select the optimal grouping, and appropriately populate the SHORT node of the outer trellis.

2.3.6 Complexity reduction

The complexity (or encoding time) can be considerably reduced via memory trade-off. All the above methods require multiple traversals of the audio file, iterating over λ or γ. But the distortion and number of bits associated with a given state of the inner trellis do not depend on the values of these iteration parameters. Thus, concurrent computation of costs for multiple values of λ or γ can eliminate redundant effort. This is akin to maintaining parallel outer and

inner trellises each running at a different value of λ or γ while sharing per state results. If a wide and finely divided range of these iteration parameters is used, the best decisions can be obtained in a single traversal of the audio file. Additionally one could also find the best decisions for a range of encoding rates, if desired. The hybrid nature of the MTNMR problem necessitates additional iterations over the inner parameter λ_{inner} to satisfy a specific distortion constraint γ. The maintenance of parallel trellises as described above helps to reuse such iterations for different values of γ.

2.3.7 Generalization to other codecs

The delayed decisions (beyond the frame) are implemented by the outer Window Switching Trellis. The computational efficiency of the trellis is due to the fact that, in AAC, distortion $D(\mathbf{x}_k, P_k)$ and bit usage $B(\mathbf{x}_k, P_k)$ for frame k are independent of encoding decisions in other frames. This characteristic is shared by many other audio codecs, including Lucent's PAC [81], Dolby's AC-3 [28] and Sony's ATRAC [4]. These codecs analyze audio samples (in the case of ATRAC, sub-band outputs of a very low resolution QMF) in frames and switch between different frame resolutions. As in AAC, the frames are encoded separately and share the available bit resource through heuristic allocation.

Moreover, all the above codecs employ a critical band based analysis within each frame, find quantizers (SF equivalents) for the frequency domain signal using the masking thresholds and, with the exception of AC-3, noiselessly encode the quantized spectra. Therefore an inner trellis scheme with modified node and transition costs can be devised for these codecs.

2.4 Results

We discuss here the experimental setup, including implementation details, and present simulation results. We first list the codecs under comparison.

1. *Reference Model (RM)*: The MPEG-4 Verification Model [90] using only the psychoacoustic model, TLS, bit-reservoir and transient detection based window switching with a restricted set of 8 window grouping choices.

2. *Inner-Trellis-only models RM-TB(T) and RM-TB(M)*: use the same blocks as the RM except that greedy TLS is replaced by the trellis-based parameter selection of [2] and [3]. Modifications for SHORT windows as described in Sec. 2.3.5 are used. RM-TB(T) minimizes TNMR and RM-TB(M) minimizes MNMR within a frame, given a rate constraint. They do not optimize windows and rate distribution across frames.

3. *Outer-Trellis-only models L1-AT, L1-MT, and L1-MM*: use the outer trellis to find the window decisions and bit distributions so as to minimize AT-NMR, MTNMR, and MMNMR, respectively. The minimum costs in (2.17) and (2.23) have to be obtained to populate the outer trellis. Since the aim of these models is to isolate the effect of the outer trellis, a complete minimization over all possible SF and HCB sets (S, H in (2.17) and (2.23)), using the inner trellis, is not effected. Instead a modified TLS is used, in each frame and in every window configuration, as follows: TLS starts off at a low value of distortion (NMR) and corresponding high bit-rate. In subsequent iterations the target NMR is increased in fixed steps till the specified bit-rate for the frame is achieved. Thus, if the bit-rate constraint

in the outer loop is set to 0, TLS passes through all of its operational rate-distortion points, each corresponding to one (S, H) pair. The minimization in (2.17) and (2.23) is effected only over this restricted set of (S, H) pairs. Thus the models L1-AT, L1-MT and L1-MM, by not incorporating the inner trellis, optimize pan-frame decisions but not the choice of parameters within a frame.

4. *Two-Layered Trellis-based models L2-AT, L2-MT, and L2-MM*: use the two-layered trellis-based algorithms (i.e., both inner and outer trellis) to minimize ATNMR, MTNMR, and MMNMR distortion measures, respectively, for the entire file.

At this juncture we note that though RM, RM-TB(T), and RM-TB(M) can code different frames with a different number of bits, they are still referred to, in general parlance, as constant bit-rate (CBR) codecs. Since these codecs employ a bit-reservoir they ensure that the bitstream can be decoded in real time with constant delay when transmitted over a constant bit-rate channel. The L1- and L2- approaches (in which cases too the instantaneous bit-rate fluctuates) would on the other hand be referred to as average bit-rate (ABR) codecs as they do not employ a bit-reservoir but are still coded to achieve a target mean bit-rate. In case of these codecs it might be necessary to buffer a larger chunk of the bitstream at the decoder before playback starts.

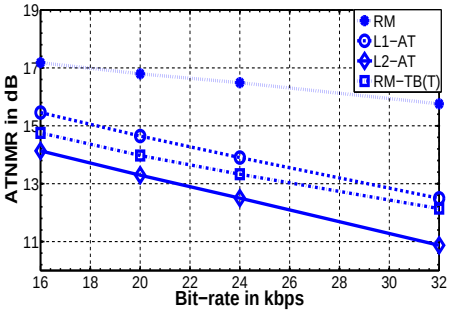
All the trellis-based approaches used the parallelization methods described in Sec. 2.3.6 for computational efficiency. A set of 10 mono, 16-bit PCM audio files sampled at 44.1 kHz, from the EBU-SQAM [89] database were used for the tests. These samples included tonal signals such as the accordion, signals with attacks

such as harpsichord and glockenspiel, speech and general pop music.

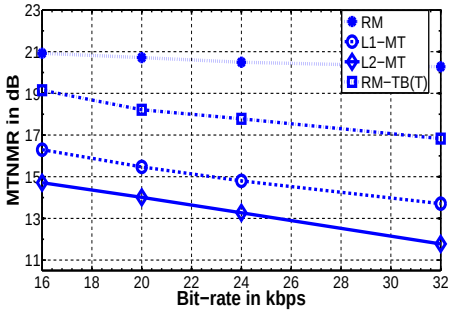
2.4.1 Objective results

Fig. 2.5a compares the gains (reduction in ATNMR) over RM achieved by: optimizing decisions only across frames (L1-AT), only within frames (RM-TB(T)), and optimizing both intra- and inter-frame decisions (L2-AT). The distortion has been averaged over the 10 audio samples. Overall optimization yields the best gains (3-5 dB over RM). Fig. 2.5b compares the performance of the corresponding encoders when the MTNMR measure is optimized. RM shows hardly any decrease in distortion as the bit-rate is increased. This is due to its sub-optimal bit distribution. Most audio samples contain critical frames that require a large number of bits for transparent coding. As the bit-reservoir of RM is inefficient, the maximum distortion (MTNMR) exhibits negligible improvement with increase in average bit-rate. Note that RM-TB(T) also uses the bit-reservoir and hence L1-MT outperforms it by achieving better bit-distribution. This trend in gains is in contrast to the previous case of minimizing average overall distortion (ATNMR). Fig. 2.5c shows the gains when the MMNMR measure is minimized. The two-layered trellis approach (L2-MM) achieves gains of 10-12 dB over RM and about 8 dB over the single-layered trellis approaches, RM-TB(M) and L1-MM, at various bit-rates. As in the MTNMR case, the outer-trellis-only method L1-MM beats RM-TB(M) at low bit-rates thanks to efficient bit distribution across frames. But at higher bit-rates the inner-trellis-only method RM-TB(M) performs better owing to its improved MNMR minimization in each frame, over the sub-optimal TLS of L1-MM. Fig. 2.6 compares window decisions based on transient detection (RM) to that of the Window Switching Trellis (L2-MT), in

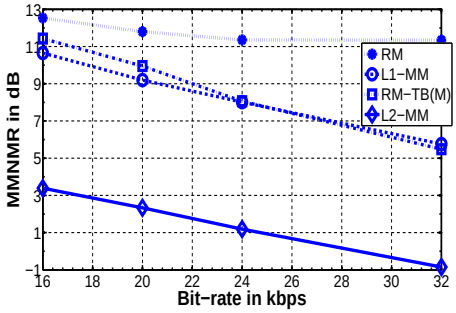
case of the glockenspiel sample. Rate-distortion optimization leads to different window decisions from that of the RM.



(a) Comparison of ATNMR produced by RM, RM-TB(T), L1-AT, and L2-AT at different bit-rates



(b) Comparison of MTNMR produced by RM, RM-TB(T), L1-MT, and L2-MT at different bit-rates



(c) Comparison of MMNMR produced by RM, RM-TB(M), L1-MM, and L2-MM at different bit-rates

Figure 2.5. Comparison of the different encoders based on objective measures

2.4.2 Subjective evaluation

The effect of optimizing encoding decisions on subjective quality depends critically on the ability of the distortion measure to reflect psychoacoustic effects.

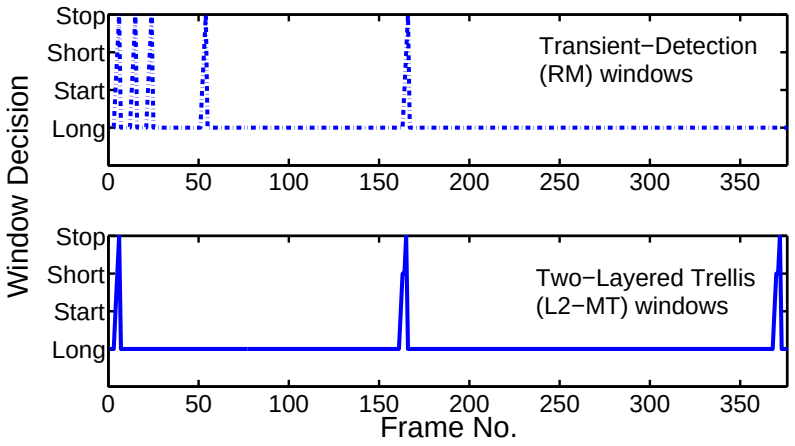


Figure 2.6. Comparison of window decisions made by RM and L2-MT for the glockenspiel sample. Peaks indicate transitions to SHORT configuration.

Subjective tests indicated that minimizing the MTNMR measure improves audio quality. MUSHRA tests [46] were conducted with 20 listeners and 6 audio samples (tenor, harpsichord, accordion, side-drums, male German speech and female English speech) encoded at 16 kbps. Fig. 2.7 shows the results of these tests. The MUSHRA scores have been averaged across samples. The two-layered trellis approach (L2-MT) has the best performance followed by RM-TB(T) and L1-MT. The reference model RM produces the worst quality of audio. Minimizing the MTNMR measure is roughly equivalent to maintaining a constant distortion (TNMR) across frames. The argument for this is as follows: If all the frames do not have the same distortion, then bits used in frames with lesser distortion can be reallocated, thus incrementally increasing distortion in these frames while reducing that in the frame with maximum distortion. This would in effect minimize the overall maximum distortion (MTNMR), but naturally tends to spread the distortion equally over the frames. This uniformity in distortion, which is evident in Fig. 2.3, may explain why MTNMR minimization yields improved

subjective quality, as well as why ATNMR minimization was observed to compromise subjective quality. The MMNMR approach also uses maximum overall distortion. Additionally it considers the maximum distortion amongst SFBs of a frame too. Hence it effectively maintains uniformity in distortion across all frames as well as all frequency bands. Yet the MMNMR approach was observed to accentuate some high frequency artifacts, and subjectively it performed somewhat worse than the MTNMR method.

It should be noted that despite the poorer quality of the ATNMR and MMNMR minimization approaches, these methods should not be dismissed. Since there is no universally precise audio distortion measure, it is possible that future measures benefit from optimization in the ATNMR or MMNMR fashion. Indeed in Chapter 3 we describe a modified NMR metric which when optimized in the MMNMR fashion provides coded audio of excellent quality.

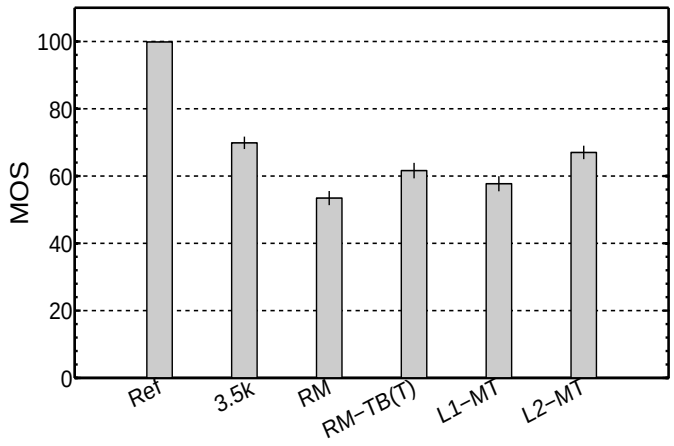


Figure 2.7. Comparison of MUSHRA scores of RM, RM-TB(T), L1-MT, and L2-MT for audio encoded at 16 kbps. 'Ref' represents the original audio and '3.5k' is the low pass anchor.

2.4.3 Complexity

The encoding complexity of all the methods is linear in the number of frames. Therefore we simply compare the average time to encode a frame, normalized by that of RM, to get the relative figures of complexity shown in Table 2.1. Note that the delayed decision part of the proposed approach actually comes from the outer trellis but as the table indicates, using the outer trellis to implement better window switching and bit-distribution (i.e., the L1- approaches) is only about 15 times more complex than RM. A major contribution to the complexity of the L2- approaches is actually the inner trellis. This suggests that sub-optimal intra-frame parameter selection alternatives to the inner trellis could be used to obtain low complexity delayed-decision based algorithms. One could, for example, prune the number of transitions possible from one stage of the inner trellis to the next, as suggested in [3], and thus reduce the number of paths to be compared and hence the complexity.

Encoder	Relative Complexity
RM	1
RM-TB(T,M)	30
L1-(AT,MT,MM)	15
L2-(AT,MM)	450
L2-MT	4500

Table 2.1. Relative figures of complexity of the various encoding methods

Another possibility, in case of the L2-MT approach, is to linearly interpolate between rate-distortion points for a frame with distortion on the logarithmic scale to get an approximate λ_{inner} that satisfies the bit-rate constraint γ, instead of iterating over multiple values of λ_{inner} as demanded by the MTNMR solution. Such linear interpolation was observed to reduce the complexity figure of the

L2-MT approach by a factor of 4 but is sub-optimal (reduction in gains by 0.2 dB).

2.5 Conclusion

In this chapter, we derived a two-layered trellis-based optimization scheme for audio coding while minimizing three different overall distortion measures - ATNMR, MTNMR, and MMNMR. The trellis effectively optimizes all the encoding decisions of the reference encoder by making delayed decisions regarding each frame. The delay and one time encoding complexity do not impact the decoder, and the bitstream generated is standard compatible. Scenarios which involve off-line encoding of audio may substantially benefit from this overall optimization process. Objective and subjective results in the AAC setting support such a delayed-decision based optimization procedure.

Chapter 3

Modifications to the Audio Distortion Metric

In Chapter 2, we developed algorithms for R-D optimized delayed decisions-based compression of audio. Different algorithms in a two-layered trellis framework were proposed, each for a distinct definition - (2.4), (2.5), or (2.6) - of the overall distortion. But all these distortion metrics were just variants of the commonly employed NMR defined as (2.1). The NMR has been the distortion measure of choice in the quantization and coding module since early audio encoders such as the OCF [15], Brandenburg-Johnston hybrid [17], and ASPEC [16] coders, and was eventually integrated into the encoding schemes specified in the informative parts of the MPEG standards [42], [43], [44]. Other objective measurement techniques for audio quality estimation include the auditory spectrum distance [51], perceptual audio quality measure (PAQM) [9], PERCEVAL [71], audible error and error margin of [50], and the more recent PEAQ [45], [83].

These metrics may be psychoacoustically more accurate than NMR, but they are too complicated for calculation in the encoding process, and hence do not render themselves amenable for optimization. Thus, most audio encoder optimization work in the past has relied exclusively on using the NMR, or its derivatives such as ANMR, MNMR, MTNMR, etc., [3], [7], [8], [61], [62], [65], [67], [68]. As noted in Sec. 2.4, although the two-layered trellis results in substantial gains in terms of the objective measures (i.e., in terms of reduction in the distortion metric), commensurate gain in terms of subjective quality depends critically on how well the distortion metric is able to capture coding artifacts. This is evident from the discussion in Sec. 2.4.2. Therefore, we are motivated to analyze deficiencies if any in the NMR metric itself, i.e., in its definition by (2.1), and propose simple modifications to it, so that it is subjectively more accurate. Preliminary results of this work have been presented in [60] and [63].

The primary function of the distortion metric, in the encoder, is to compare different choices of encoding parameters by their effect on the coding quality. This in turn is achieved by comparing the noise due to quantization in different frequency bands, in terms of their true psychoacoustic cost. The masking threshold incorporated in NMR tries to achieve exactly this. In addition, we identify two other requirements to ensure an efficient comparison:

- The distortion in each coding band (SFB in AAC) depends not just on the ratio of the noise spectrum and threshold spectrum (which is what NMR provides) but 'how much' of this ratio is present in each band. This means that, assuming this ratio is a constant across the band, the distortion metric needs to be cognizant of the bandwidth of each SFB on a frequency scale that is relevant to human hearing, i.e., the Bark scale [72, 88]. If all bands

were of equal width on such a scale, NMR as currently defined would be appropriate.

- The true estimate of the distortion due to quantization in different frequency bands should really take into account the effects, if any, of decoder operations too on the quantization noise spectrum: what the listener hears is what is decoded and reconstructed. As we see later in Sec. 3.2 the windowing and overlap-add operations at the decoder has a non-trivial effect on the noise spectrum.

Consider the comparison of window decisions, that is targeted by the two-layered trellis in Chapter 2. A major difference between the LONG and the SHORT windows is that the same frequency range is divided into a larger number of SFBs in the LONG configuration (49 SFBs at 44.1kHz sampling rate) than in the SHORT configuration (14 SFBs at 44.1kHz sampling rate). Thus the widths of the SFBs on the Bark scale (henceforth referred to as 'Bark width') are different in the two modes. Hence frame distortion measures such as the ANMR and MNMR which give equal weights to the NMRs of all SFBs irrespective of their width fail to yield an effective comparison of the different window configurations. Further as we shall see in Sec. 3.1, even within the same window configuration, SFBs are of different Bark widths. Therefore this chapter suggests a Bark scale correction to the existing NMR distortion measure. If the Bark widths do not change across bands, the new measure degenerates back to the standard NMR comparison between bands. The new measure called NMR with Bark Correction (NMR-BC) is utilized in the TLS of the MPEG VM, and the two-layered trellis algorithm of Chapter 2. Listening tests indicate that there is a strong preference for audio encoded in light of NMR-BC than when the usual NMR metric is used,

and that the right distortion metric is critical to the efficacy of delayed decisions based R-D optimal audio coding.

In Sec. 3.2 of this chapter we consider the effect of decoder operations on the distortion. Consider the AAC decoder. The quantized coefficients are reconstructed, an inverse MDCT applied, and frames overlap added. The overlap additions results in time-domain noise components from adjacent frames combining, and it is natural to expect that the distortion for a frame as calculated after the overlap-add operation will be different from prior to it. In [24], this effect is rightly demonstrated but in an audio encoder based on discrete Fourier transform (DFT) of 50% overlapped frames (also a perfect reconstruction filterbank akin to MDCT). But in the case of MDCT-based coders we can show that the overlap error components from neighboring frames are orthogonal to the MDCT basis vectors of the current frame. Thus distortion metrics based solely on MDCT domain error do not capture overlap-add effects. Note that NMR as defined by (2.1) of Chapter 2 falls in this category. The error orthogonal to the MDCT bases can be analyzed using the modified discrete sine transform (MDST). Such analysis reveals that in addition to the overlap contributions, the orthogonal error has a component from quantization in the current frame itself due to the non-rectangular window used. In other words, the decoder based windowing leads to a spreading of quantization noise from the MDCT domain to the MDST domain. Since the human ear is sensitive to the magnitude of noise at any frequency rather than its projections only on cosine or sine bases, a modified distortion measure is proposed that accounts for the MDST domain error. The fact that the windows used in these transforms are heavy centered and taper at the ends leads to the MDST domain error being dominated by the effect of

decoder based windowing rather than overlap-add. Thus a simplified version of the distortion metric which accounts only for the window effect is implemented in the TLS for quantization and coding parameters of the MPEG VM AAC encoder. Subjective tests indicate a preference for audio encoded in light of this modification rather than the usual NMR metric. Experiments are performed using both window choices, sine and Kaiser Bessel derived (KBD), available in the AAC standard. The advantages of one over the other with respect to the new metric are discussed.

3.1 Distortion Modification based on Bark Bandwidths

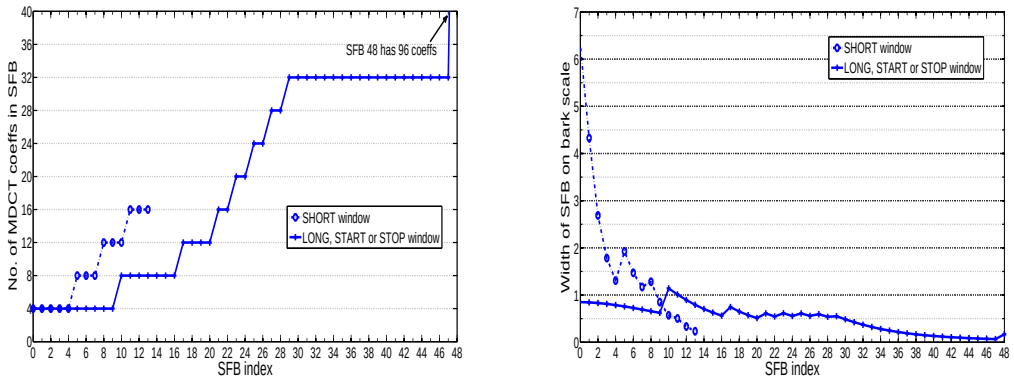
3.1.1 Preliminaries

Bark widths of SFBs

Experiments in psychoacoustics have led to the perception of the 'Critical Bandwidth' [72], [88] which leads to a non-linear mapping of the normal frequency scale. This transformed scaled is known as the Bark scale. The following function [88] is often used to convert frequency f on the hertz scale to the corresponding value $Z_b(f)$ on the Bark scale.

$$Z_b(f) = 13\ arctan(0.00076f) + \ 3.5\ arctan\left[\left(\frac{f}{7500}\right)^2\right]\ \ Bark \qquad (3.1)$$

The idea is that the frequency ranges corresponding to equal widths on the Bark scale have equal perceptual importance. A frequency band corresponding to 1



(a) SFB widths in number of MDCT coefficients (b) SFB widths on the Bark Scale

Figure 3.1. SFB widths for different window configurations at 44.1kHz sampling frequency. The same frequency range is covered by 14 SFBs in the SHORT configuration and 49 in the other modes.

Bark is known as the 'Critical Band' around the center frequency of that band, i.e., each critical band is 1 Bark wide. The non-linear mapping has the effect that a critical band corresponds to a smaller frequency band at lower center frequencies on the Hertz scale than at higher ones.

In AAC, 1024 MDCT coefficients are generated for every LONG, START or STOP frame. These coefficients, covering the range from 0Hz to half the sampling frequency, are grouped into SFBs. The lower indexed SFBs have fewer MDCT coefficients than the higher ones reflecting the non-linear Bark scale. This is indicated by the bold line in Fig. 3.1a which shows the number of coefficients in each SFB of a LONG, START or STOP window at a sampling rate of 44.1kHz (i.e., audio bandwidth of 22.05kHz). Despite this unequal number of coefficients in each SFB the 'Bark widths' of the SFBs are not really the same as is evident in Fig. 3.1b.

On the other hand consider the SHORT configuration. In this case 128 MDCT coefficients are generated for each SHORT frame and these are grouped into 14 SFBs covering the same 22.05kHz audio bandwidth. The number of coefficients per SFB and the Bark widths are shown by the dotted lines in Fig. 3.1a and Fig. 3.1b. It is clear from the figures that there are differences in the Bark widths of SFBs, both in the same window configuration and between different configurations.

Noise-to-mask ratio

The NMR, as originally employed in [18], consisted of calculating the difference (error) between original and coded waveforms frame-wise using an FFT, computing the error energy in each band of a frame, and dividing by the corresponding masking threshold. A table of widths of these analysis bands at 44.1kHz and 48kHz sampling frequencies is provided in [18]. The bands, although not exactly 1 Bark wide, were of equal Bark widths. Later encoders such as the MPEG VM, directly calculate the quantization error in each coding band (SFB) in the transform (MDCT) domain. The psychoacoustic model provides corresponding masking thresholds and the NMR is defined as in (2.1) of Chapter 2. But as noted above not all the SFBs in this case have equal Bark widths.

Ideally, the masking threshold in an SFB determines the maximum amount of quantization noise in that SFB that is imperceptible to the human auditory system. But in low bit-rate audio coding invariably the quantization noise exceeds the masking thresholds in many SFBs, in many frames, and is audible. Since the NMR is a ratio, examples can be contrived where the noise powers in two SFBs maybe different but due to proportional masking thresholds, the NMR is the

same, thus indicating that the listener would perceive the same distortion (or noise), which is counter-intuitive.

Experiments with synthetic audio

The following experiment was conducted to test if the same NMR actually meant that listeners would perceive the same noise level. Consider a pure tone at frequency $f_c = 400$Hz.

Sample A: Noise with a constant power spectral density (psd) centered at f_c, with width $b_A = 0.4$ Bark and level σ_A^2 is superimposed with the tone. At 400Hz, 0.4 Bark approximately corresponds to a 40Hz bandwidth

Sample B: Noise with psd centered at f_c and with width $b_B = 0.8$ Bark (approximately 80Hz bandwidth) and level σ_B^2 is superimposed with the tone.

The integral of the 'spreading function' (see [72] for a description) in a region of 0.8 Bark around the tonal masker is about 1.4 times that in a region of 0.4 Bark, implying that the masking threshold in B is 1.4 times that in A. The same NMR can be maintained in both cases if the noise powers are in the proportion,

$$\frac{2\sigma_B^2}{\sigma_A^2} = 1.4 \tag{3.2}$$

where the factor of 2 comes from the noise bandwidth difference. The above scaling was used to have approporiate psd levels. The tone power was higher than the noise power by about 14dB in A and 12.5dB in B so that noise-masking-tone (NMT) effects [72] can be neglected and clearly the tone was the masker. Test subjects were played Samples A and B in random order, and blindly, and were asked to select one of the following options.

- The two signals have the same noise level.

- The two signals have different noise levels with an identification of which had more noise.

86% of the listeners identified the sample corresponding to B as having higher noise power while 14% opined that there was no difference. No listeners suggested that A had a higher noise level. This suggests that if noise is unmasked (i.e., audible) NMR wrongly indicates the same perceived noise level. The difference between the above two cases stems from the fact that the noise psd has different supports and hence the masking thresholds and noise powers were different. Thus a distortion measure which accounts for differences in Bark widths of SFBs would be attractive.

3.1.2 Noise-to-mask ratio with bark correction

We suggest here a correction to the NMR measure to explicitly account for Bark width differences. Specifically, the NMR of each coding band is scaled by its Bark width. This new measure is referred to as NMR with Bark Correction (NMR-BC). Following the notation in (2.1), NMR-BC d_i^{bc} for SFB i is given by,

$$\begin{aligned} d_i^{bc} &= \mu_i e_i \times \frac{b_i \ Bark}{1 \ Bark} \\ &= b_i \mu_i e_i \end{aligned} \tag{3.3}$$

where b_i is the Bark width of SFB i. As indicated by (3.3), NMR-BC can be thought of as apportioning the NMR according to the Bark width.

Note that the experiment in Sec. 3.1.1 is not the only way the same NMR can be maintained with the perceived noise being different. For example, a louder version of Sample A would have proportionately scaled noise and masker powers

and hence same NMR as A. But noise would of course be perceived louder in that case. More complex measures like the PAQM [9] do account for loudness levels but are not easy to be used in optimization procedures. Here we have limited our attention to distortion modification to include Bark width differences.

Experiment with synthetic audio extended

We use the same setting of the experiment in Sec. 3.1.1. In addition to samples A and B we have,

Sample C: Noise with psd centered at f_c, width $b_C = 0.4$ Bark and level $\sigma_C^2 = 2\sigma_A^2$ is superimposed with the tone.

C differs from A in only that the noise psd level is higher in C. Thus A and B have same NMR while B and C have same NMR-BC. Listeners were asked to identify the two closest in terms of perceived noise level amongst the three samples. Again tests were blind and random. The listeners additionally had the option of stating that they were unable to decide. 66% of the listeners opined that B and C had similar noise loudness, 22% were unable to decide and 12% chose A and B as similar. None chose A and C as similar. Since B and C had the same NMR-BC, this test offers an indication that NMR-BC might be a good measure to use when comparing signals with noise spread across different Bark widths. More tone plus noise experiments are essential to conclusively state so but the above experiments inspired the use of this measure for coding real audio in an AAC setting.

Incorporation of NMR-BC into the encoder

Both distortion metrics NMR (2.1) and NMR-BC (3.3) are defined per SFB per frame. Therefore incorporating NMR-BC into the encoder is just a substitution in place of NMR, in the TLS of MPEG VM (Sec. 2.1), or in the two-layered trellis approaches of Chapter 2. In the latter case we restrict our experiments here to the MMNMR trellis of Sec. 2.3.4. When using NMR-BC as the per band distortion, the corresponding overall distortion metric will be referred to as MMNMR-BC. We note here that minimizing the MNMR (or MNMR-BC) among SFBs is very close to what the TLS aims for, i.e., maintaining the same NMR (or NMR-BC) in each frame. The argument for this is as follows. If all the SFBs do not have the same distortion, then bits used in SFBs with lesser distortion can be reallocated to incrementally increase distortion in them while bringing down that in the SFB with the maximum distortion. This would in effect minimize the overall maximum distortion or the MNMR. This course eventually leads to almost the same distortion in each SFB. In Sec. 3.1.3 the MPEG VM encoder whose TLS employs NMR will be referred to as RM; when it employs NMR-BC it is referred to as RM-BC. In case of the two-layered trellis approach we refer to the corresponding codecs as L2-MM, and L2-MM-BC, respectively.

3.1.3 Results

TLS based optimization

The TLS based encoders RM and RM-BC described in Sec. 2.1 were used to encode audio sampled at 44.1kHz. Both these encoders used transient detection

based window switching and the bit-reservoir to allocate bits to frames. Five such coded audio samples (accordion, orchestra, male german speech, glockenspiel and tenor [89]) were used for testing. Listeners were presented with:

O: the original uncoded sample

A: sample encoded using RM at 48kbps

B: sample encoded using RM-BC at 48kbps

The tests were blind with A and B randomly ordered and the listeners able to near instantaneously shift between playing the two samples as in MUSHRA testing [46]. Since both A and B are coded at moderately high bit-rates the artifacts are expected to be low in either and thus the original was provided to the listeners to help identify these artifacts. Listeners could select either sample as the better quality audio or declare them to be equally good. Table. 3.1 shows the results of these tests. Clearly RM-BC performs considerably better on three of the five audio samples and somewhat better on the other two.

Audio sample	RM-BC	RM	No Preference
accordion	66.7%	0%	33.3%
glockenspiel	66.7%	13.3%	20%
german speech	100%	0%	0%
orchestra	40%	33.3%	26.7%
tenor	40%	26.3%	33.3%

Table 3.1. Subjective comparison tests of RM and RM-BC: The figures are the percentage of listeners who preferred audio encoded using corresponding method.

Two-layered trellis based optimization

The TLS, window decision and bit-reservoir of the MPEG VM were replaced by the two-layered trellis approach described previously. The same audio samples as in the last section were encoded at 20kbps using these two codecs and listening tests conducted similarly i.e.

O: the original uncoded sample

A: sample encoded using L2-MM at 20kbps

B: sample encoded using L2-MM-BC at 20kbps

A and B in this case being encoded at low bit-rates are either duller or have artifacts when compared to O. The listeners were asked to give their preference between A and B. Again the original helped listeners to identify artifacts. The results of these tests are shown in Table. 3.2. Four out of the five audio samples were better encoded better by L2-MM-BC while one sample shows only minor improvement.

Audio sample	L2-MM-BC	L2-MM	No Preference
accordion	60%	26.7%	13.3%
glockenspiel	80%	6.7%	13.3%
german speech	66.7%	20%	13.3%
orchestra	40%	33.3%	26.7%
tenor	60%	20%	20%

Table 3.2. Subjective comparison tests of L2-MM and L2-MM-BC: The figures are the percentage of listeners who preferred audio encoded using corresponding method.

Comparison of objective metrics

We describe here the gains of the two-layered trellis approach (L2-MM-BC) when compared to RM-BC in terms of the distortion metric (MMNMR-BC) at different bit-rates. We also include corresponding single-layered trellis algorithms (see description in Sec. 2.4) RM-TB(M)-BC, and L1-MM-BC, that respectively optimize only intra-frame, or only inter-frame decisions. The performance of these four encoders in terms of the concerned distortion metric is shown in Fig. 3.2. When compared to the TLS based approach each of the single layered trellis approaches fare better by about 5-8 dB at different bit-rates and the two-layered trellis approach provides an additional 10dB gain.

Fig. 3.3 shows the window decisions due to transient detection based window switching (RM), two-layered trellis based approach using the NMR (L2-MM) measure and when using NMR-BC (L2-MM-BC). L2-MM doesn't yield a good comparison between LONG and SHORT windows. It under-estimates the distortion in SHORT windows and thus heavily favors their presence. But such distortion is actually audible as disturbing noise in the audio sample. This problem is rectified by L2-MM-BC thus justifying the motivation for NMR-BC. Note also that the L2-MM-BC approach decides windows differently from the transient detection based scheme. Even in places where they seem to match, the switching decisions are actually off by a frame, which is not visible due to the low resolution of the graph.

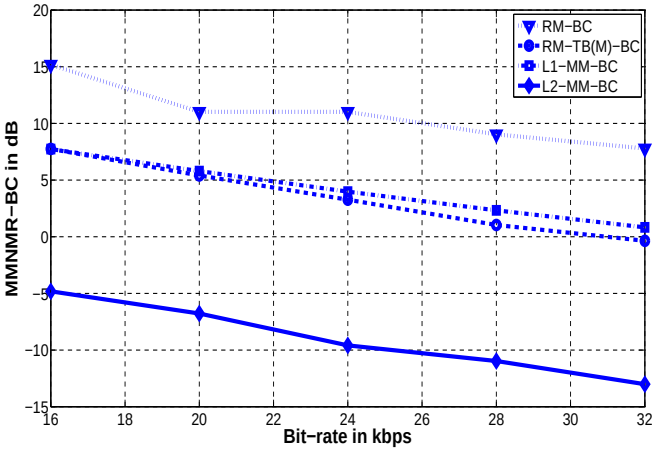


Figure 3.2. Distortion at various bit-rates due to encoding using RM-BC, single layered trellis approaches (RM-TB(M)-BC and L1-MM-BC), and the two-layered trellis approach L2-MM-BC

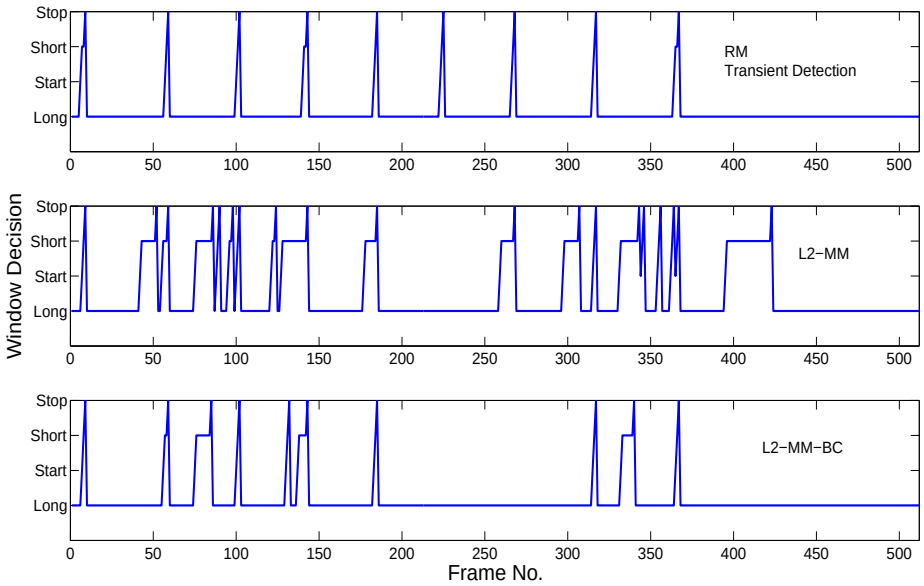
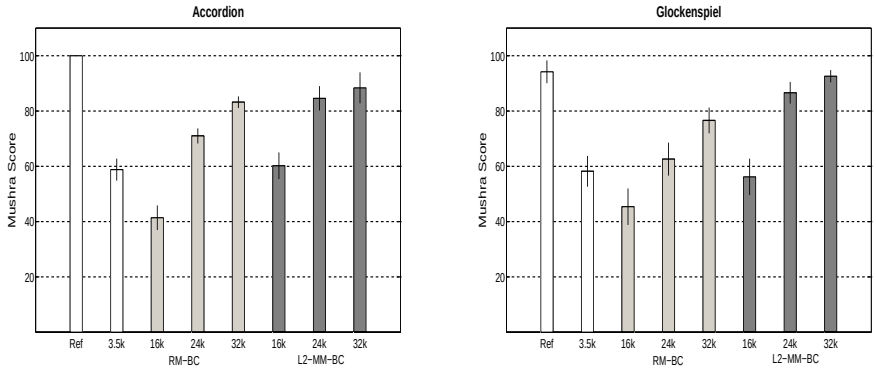
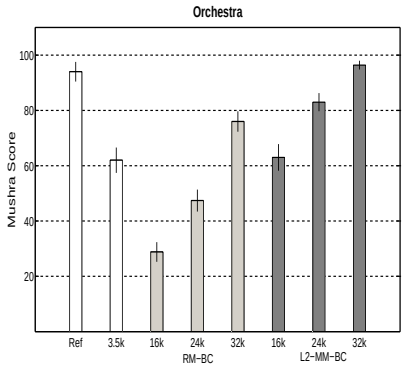


Figure 3.3. Window decisions due to transient detection (RM) and due to using the Window Switching Trellis (L2-MM and L2-MM-BC): A jump in the graph indicates a switch from 'LONG' to 'SHORT' configurations.



(a)

(b)



(c)

Figure 3.4. MUSHRA tests comparing TLS based and two-layered trellis based encoders when minimizing NMR-BC: Quality of audio encoded at 16, 24 and 32kbps is shown. Ref is the hidden original and 3.5k is the low pass anchor.

Subjective improvements due to the two-layered trellis

MUSHRA Tests were conducted with 3 audio samples (accordion, orchestra and glockenspiel) encoded at 16, 24 and 32 kbps using RM-BC and L2-MM-BC. Each test had a reference original, hidden original and 3.5k low pass anchor. The aim was to identify the gains provided by the two-layered trellis when compared to the TLS based approach. Additionally, the experiments indicate the smoothness

of degradation of quality with bit-rate decrements. The results are shown in Fig. 3.4. The superior performance of the two-layered trellis approach is evident. Unlike the RM-BC, the quality for L2-MM-BC degrades rather smoothly. In 2 of the 3 cases, the very good quality of audio coded using L2-MM-BC at 32kbps led to the hidden original being identified wrongly.

3.2 Distortion Modification to account for Decoder-end Operations

We now consider the effect of decoder end operations on the perceived distortion in coded audio.

3.2.1 Problem setting

Audio coding methods such as AAC convert overlapped frames of audio to the frequency domain using a suitable transform which in many cases (including AAC) is the MDCT [55], [73], [79]. As already described, the transform coefficients are grouped into psychoacoustically relevant partitions, quantized and entropy coded. The quantization and coding parameters are chosen so that a distortion measure such as the NMR is minimized subject to a bit-rate constraint. At the decoder the frame's quantized coefficients are inverse transformed and overlap-added with neighboring frames to reconstruct the time domain audio signal. This is illustrated in Fig. 3.5. Each vector $\underline{x}_k$ denotes a 'frame shift' of audio samples. Frame k, composed of $\underline{x}_k$ and $\underline{x}_{k+1}$, is used to obtain the vector of transform coefficients $\underline{X}_k$. This when quantized yields $\hat{\underline{X}}_k$ which is entropy

coded losslessly and hence received intact at the decoder. The reconstruction $\underline{\hat{x}}_k$ is obtained by the overlap-add of the inverse transforms $\underline{z}_{k-1}$ and $\underline{z}_k$ of $\underline{\hat{X}}_{k-1}$ and $\underline{\hat{X}}_k$, respectively. Prior to the transformation at the encoder and post inverse transformation at the decoder, the frames are multiplied by a suitable window choice to avoid blocking effects. This operation can in fact be embedded in the transform (and its inverse) as is the case with MDCT (see Sec. 3.2.2) and is implicit in the corresponding stages of Fig. 3.5.

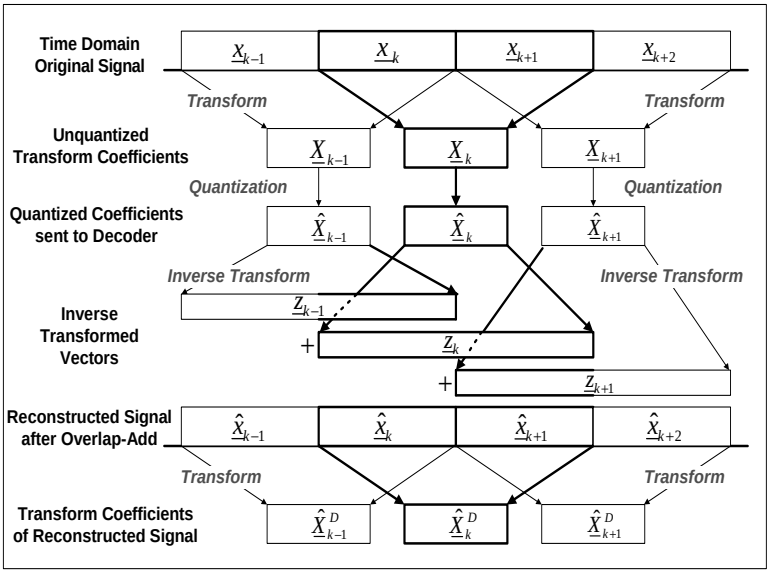


Figure 3.5. Signal analysis in audio coding. The frequency domain reconstructed signal is added here to illustrate the discussion.

Note that the reconstructed frame k comprising of $\underline{\hat{x}}_k$ and $\underline{\hat{x}}_{k+1}$ has error contributions due to quantization of not just $\underline{X}_k$ but also $\underline{X}_{k-1}$ and $\underline{X}_{k+1}$. But current encoders (including the ones in Chapter 2) calculate distortion for each frame individually, i.e., using a metric of the form $D(\underline{X}_k, \underline{\hat{X}}_k)$ which ignores the effect of any decoder based operation such as overlap-add. Thus it is instructive to see if analysis (in the frequency domain) of the decoded time domain signal can capture these effects. To this end, consider applying the same transform

and framing as in the encoder to the reconstructed time domain signal. The resulting transform coefficients are shown as $\underline{\hat{X}}_k^D$ in Fig. 3.5. The same metric as before could be used to define the "end-to-end" distortion $D(\underline{X}_k, \underline{\hat{X}}_k^D)$. It will be observed later that in the case of lapped orthogonal transforms (LOTs) [55], [56], to which class the MDCT belongs, $\underline{\hat{X}}_k^D = \underline{\hat{X}}_k$ and hence $D(\underline{X}_k, \underline{\hat{X}}_k) = D(\underline{X}_k, \underline{\hat{X}}_k^D)$. This is not true for other well known transforms that ensure perfect reconstruction, including the discrete Fourier transform (DFT). The latter fact is rightly demonstrated in [24], where the authors using an audio encoder based on DFT of 50% overlapped frames show that $D(\underline{X}_k, \underline{\hat{X}}_k) \neq D(\underline{X}_k, \underline{\hat{X}}_k^D)$. In the discussion to follow we analyze what causes this difference between the MDCT and DFT coders, and if really, as implied by above arguments, decoder operations have no effect on the perceived distortion in an MDCT-based coding scheme.

3.2.2 Preliminaries

We introduce here some notation as well as relevant background information on MDCT with reference to the schematic in Fig. 3.5. Segment $\underline{x}_k$ of the original signal and corresponding reconstruction $\underline{\hat{x}}_k$ are column vectors of M audio samples. The k^{th} original and reconstructed frames of length $2M$ are, respectively,

$$\mathbf{x}_k = \begin{bmatrix} \underline{x}_k \\ \\ \underline{x}_{k+1} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{x}}_k = \begin{bmatrix} \underline{\hat{x}}_k \\ \\ \underline{\hat{x}}_{k+1} \end{bmatrix} \tag{3.4}$$

Thus frames are 50% overlapped. MDCT of $2M$ audio samples yields M coefficients and the $M \times 2M$ forward MDCT matrix is,

$$P = CH \tag{3.5}$$

$$\text{with} \qquad H = \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ 0 & h(1) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & h(2M-1) \end{bmatrix}_{2M \times 2M} \tag{3.6}$$

$$\text{and} \quad C = \left[\sqrt{\frac{2}{M}} cos\Big[\frac{\pi}{M} \Big(m + \frac{1}{2}\Big)\Big(n + \frac{M+1}{2}\Big)\Big] \right]_{\substack{M \times 2M \\ 0 \le m \le M-1,\ 0 \le n \le 2M-1}} \tag{3.7}$$

m and n in C are row and column indices, respectively. $h(n)$, a window of length $2M$, satisfies the constraints

$$h(2M-1-n) = h(n) \quad \text{and} \quad h^2(n) + h^2(n+M) = 1 \tag{3.8}$$

The inverse MDCT (IMDCT) matrix is P^T and obtained by transposition. Information about window prototypes and the use of MDCT in audio coding can be found in [79]. We alternatively write P as,

$$P = [P_A \ P_B] \tag{3.9}$$

where P_A and P_B are $M \times M$ sub-matrices. Applying MDCT to the original signal one obtains

$$\underline{X}_k = P\mathbf{x}_k = P_A \underline{x}_k + P_B \underline{x}_{k+1} \tag{3.10}$$

We will also consider MDCT of the reconstructed signal:

$$\underline{\hat{X}}_k^D = P\hat{\mathbf{x}}_k = P_A \underline{\hat{x}}_k + P_B \underline{\hat{x}}_{k+1} \tag{3.11}$$

The vector $\underline{X}_k$ is quantized to $\hat{\underline{X}}_k$ and the quantization error is,

$$\underline{E}_k = \underline{X}_k - \hat{\underline{X}}_k \tag{3.12}$$

The vectors $\underline{z}_k$ in Fig. 3.5 are obtained by IMDCT,

$$\underline{z}_k = P^T \hat{\underline{X}}_k = \begin{bmatrix} P_A^T \\ P_B^T \end{bmatrix} \hat{\underline{X}}_k \tag{3.13}$$

Since the MDCT belongs to the class of LOTs it satisfies the following conditions [55],

$$PP^T = P_A P_A^T + P_B P_B^T = \mathbf{I} \tag{3.14}$$

$$\text{and} \qquad P \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} P^T = \mathbf{0} \tag{3.15}$$

$$\Rightarrow \; P_A P_B^T = \mathbf{0} = P_B P_A^T \tag{3.16}$$

where $\mathbf{0}$ and $\mathbf{I}$ are each $M \times M$ in dimension. The above conditions enable perfect reconstruction and time domain aliasing cancellation properties that are characteristic of LOTs.

The reconstruction segments $\hat{\underline{x}}_k$ and $\hat{\underline{x}}_{k-1}$ are formed by overlap-add of corresponding IMDCT vectors:

$$\hat{\underline{x}}_k = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \underline{z}_{k-1} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \underline{z}_k = P_B^T \hat{\underline{X}}_{k-1} + P_A^T \hat{\underline{X}}_k \tag{3.17}$$

$$\hat{\underline{x}}_{k+1} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \underline{z}_k + \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \underline{z}_{k+1} = P_B^T \hat{\underline{X}}_k + P_A^T \hat{\underline{X}}_{k+1} \tag{3.18}$$

where $\mathbf{0}$ and $\mathbf{I}$ are of dimensions $M \times M$. Substituting into (3.11) we obtain

$$\hat{\underline{X}}_k^D = P_A P_B^T \hat{\underline{X}}_{k-1} + (P_A P_A^T + P_B P_B^T) \hat{\underline{X}}_k + P_B P_A^T \hat{\underline{X}}_{k+1} \tag{3.19}$$

$$\text{and by (3.14), (3.16)} \qquad \hat{\underline{X}}_k^D = \hat{\underline{X}}_k \tag{3.20}$$

which subsequently leads to,

$$D(\underline{X}, \hat{\underline{X}}_k) = D(\underline{X}, \hat{\underline{X}}_k^D) \tag{3.21}$$

Thus a metric such as NMR defined as quantization noise in the MDCT coefficients divided by the masking thresholds, is not altered by decoder based operations such as overlap-add and hence is deficient in its ability to capture corresponding psychoacoustic effects. The derivation of (3.21) has not explicitly used the MDCT kernel but the more general LOT properties (3.14) and (3.15). Hence (3.21) holds true for other LOTs also. Note that, as evidenced by the system of [24], (3.21) is not valid for all perfect reconstruction systems employing overlapped transforms.

3.2.3 Distortion in the MDCT and MDST domains

We now analyze the time domain error in a reconstructed frame. From (3.5), taking the MDCT of frame $\mathbf{x}_k$ implies applying the cosine based transform C to the 'windowed' frame $H\mathbf{x}_k$. The time domain reconstruction error in the k^{th} frame is $\mathbf{x}_k - \hat{\mathbf{x}}_k$. The 'windowed' error is

$$\mathbf{e}_k = H[\mathbf{x}_k - \hat{\mathbf{x}}_k] = H \begin{bmatrix} \underline{x}_k - \hat{\underline{x}}_k \\ \underline{x}_{k+1} - \hat{\underline{x}}_{k+1} \end{bmatrix} \tag{3.22}$$

By the perfect reconstruction property, absent quantization, IMDCT followed by overlap-add yields back the original samples:

$$\underline{x}_k = P_B^T \underline{X}_{k-1} + P_A^T \underline{X}_k \tag{3.23}$$

$$\underline{x}_{k+1} = P_B^T \underline{X}_k + P_A^T \underline{X}_{k+1} \tag{3.24}$$

Substituting (3.17), (3.18) and the above in (3.22) and using (3.12) we have,

$$\mathbf{e}_k = H \begin{bmatrix} P_B^T \underline{E}_{k-1} + P_A^T \underline{E}_k \\ P_B^T \underline{E}_k + P_A^T \underline{E}_{k+1} \end{bmatrix} \tag{3.25}$$

$$(3.5) \Rightarrow \; C\mathbf{e}_k = P \begin{bmatrix} P_B^T \underline{E}_{k-1} + P_A^T \underline{E}_k \\ P_B^T \underline{E}_k + P_A^T \underline{E}_{k+1} \end{bmatrix} \tag{3.26}$$

$$(3.9), (3.14), (3.16) \Rightarrow \; C\mathbf{e}_k = \mathbf{0}\underline{E}_{k-1} + \mathbf{I}\underline{E}_k + \mathbf{0}\underline{E}_{k+1} \tag{3.27}$$

This indicates that the cosine basis vectors (rows of C) are orthogonal to error components in $\mathbf{e}_k$ that result from the overlap of $\hat{\mathbf{x}}_k$ with neighboring frames. On the other hand these components can be captured using a basis set that is orthogonal to the row space of C. The sine transform S given by

$$S = \left[\sqrt{\frac{2}{M}} sin\Big[\frac{\pi}{M} \Big(m + \frac{1}{2} \Big) \Big(n + \frac{M+1}{2} \Big) \Big] \right]_{M \times 2M} \tag{3.28}$$

is one possible orthogonal basis set, i.e., $SC^T = \mathbf{0}$. Note that both C and S are of rank M and together form a 'complete basis' for the $2M$ dimensional space. By straightforward manipulations, it can be shown that

$$C^T C + S^T S = 2\mathbf{I} \tag{3.29}$$

$$\Rightarrow \; \mathbf{e}_k^T \mathbf{e}_k = \frac{1}{2} \Big[(C\mathbf{e}_k)^T (C\mathbf{e}_k) + (S\mathbf{e}_k)^T (S\mathbf{e}_k) \Big] \tag{3.30}$$

Thus the time domain error in a windowed frame can be completely analyzed using both cosine and sine transforms. Define $\underline{\mathcal{E}}_k = S\mathbf{e}_k$. By (3.25) ,

$$\underline{\mathcal{E}}_k = SH \begin{bmatrix} P_B^T \\ \mathbf{0} \end{bmatrix} \underline{E}_{k-1} + SH \begin{bmatrix} P_A^T \\ P_B^T \end{bmatrix} \underline{E}_k + SH \begin{bmatrix} \mathbf{0} \\ P_A^T \end{bmatrix} \underline{E}_{k+1} \tag{3.31}$$

$$= P_S \begin{bmatrix} P_B^T \\ \mathbf{0} \end{bmatrix} \underline{E}_{k-1} + P_S P^T \underline{E}_k + P_S \begin{bmatrix} \mathbf{0} \\ P_A^T \end{bmatrix} \underline{E}_{k+1} \tag{3.32}$$

where paralleling the treatment of MDCT we define the MDST matrix as

$$P_S = SH \tag{3.33}$$

The error $\underline{\mathcal{E}}_k$ will be referred to as the MDST domain error, as it is the MDST of the actual (not windowed) time domain error $\mathbf{x}_k - \hat{\mathbf{x}}_k$. Note that despite $SC^T = \mathbf{0}$,

$$P_S P^T = SH^2C^T \neq \mathbf{0} \tag{3.34}$$

for windows not satisfying $H^2 = \mathbf{I}$. A rigorous proof of the prior statement is left out for conciseness. It can specifically be verified for the sine and KBD windows specified by the AAC standard [43]. Thus, by (3.32), in addition to quantization error contributions from neighboring frames, part of the MDST domain error for a frame, i.e., $P_S P^T \underline{E}_k$ results from quantizing the MDCT coefficients of the concerned frame itself. In other words, the non-rectangular window used in these transforms results in 'spreading' the MDCT quantization error into the MDST domain.

As mentioned previously the metric of choice in the AAC encoder is the NMR (2.1), which can be defined in more detail for SFB i of frame k as

$$NMR_{k,i} = \mu_{k,i} \sum_{j \in SFB\ i} \underline{E}_k^2(j) \tag{3.35}$$

Here $\underline{E}_k(j)$ is the j^{th} element of $\underline{E}_k$ and $\mu_{k,i}$ is the reciprocal of the masking threshold for the i^{th} SFB of frame k, provided by a psychoacoustic model. It is well known that the human ear is sensitive to the spectral magnitude rather than any one individual orthogonal component (sine or cosine). Thus a distortion metric that accounts for the magnitude of error in different frequency bins, rather than its projection only in the MDCT domain, yields a better comparison of the effects of quantization in different coding bands. Therefore we propose an

enhanced distortion measure, NMR$^+$, which, in addition to the MDCT error, accounts for the error $\underline{\mathcal{E}}_k$ (3.32) present in the MDST domain. Specifically,

$$NMR^+_{k,i} = \mu'_{k,i} \sum_{j \in SFB\ i} [\underline{E}^2_k(j) + \underline{\mathcal{E}}^2_k(j)] \tag{3.36}$$

It follows from (3.32) that NMR$^+$ depends on the MDCT errors of neighboring frames and hence cannot be incorporated into an encoder that analyzes each frame separately, e.g., the MPEG VM. Note that the masking thresholds employed in (3.35) and (3.36) are not the same. Usually the psychoacoustic model performs an FFT of the windowed frame and finds thresholds in different bands. The FFT thresholds are eventually scaled to reflect the energy in the MDCT domain. In the case of (3.36) this threshold should additionally account for MDST domain energy.

As suggested by (3.34), the error $\underline{E}_k$ propogates to the MDST domain through H^2 (or $h^2(n)$) which is plotted in Fig. 3.6 for sine and KBD windows. The KBD window provides reduced overlap. Under the assumption that all M elements of $\underline{E}_{k-1}$, $\underline{E}_k$ and $\underline{E}_{k+1}$ are independent random variables with equal variance, (3.32) can be used to calculate the variance of elements in $\underline{\mathcal{E}}_k$ (the MDST domain error) for any specific window choice. The MDST domain error turns out to have the same variance as $\underline{E}_k$ suggesting that the orthogonal domain error is as important to account for as the MDCT domain error. In case of the sine window the errors $\underline{E}_{k-1}$ and $\underline{E}_{k+1}$ can be shown to contribute 25% each to the MDST domain error of the k^{th} frame while the remaining 50% is due to $\underline{E}_k$. For the KBD window only 15% of the MDST domain error is due to each of the neighboring frames and 70% due to MDCT quantization in the current frame. Therefore we approximate

$\underline{\mathcal{E}}_k$ by $\hat{\underline{\mathcal{E}}}_k = P_S P^T \underline{E}_k$ and NMR$^+$ by,

$$NMR^+_{k,i} \approx \frac{\sum_{j \in SFB\ i} [\underline{E}^2_k(j) + \hat{\underline{\mathcal{E}}}^2_k(j)]}{T'_i} \tag{3.37}$$

This simplified NMR$^+$ accounts for most of $\mathbf{e}^T_k \mathbf{e}_k$ in (3.30), especially in the case of the KBD window.

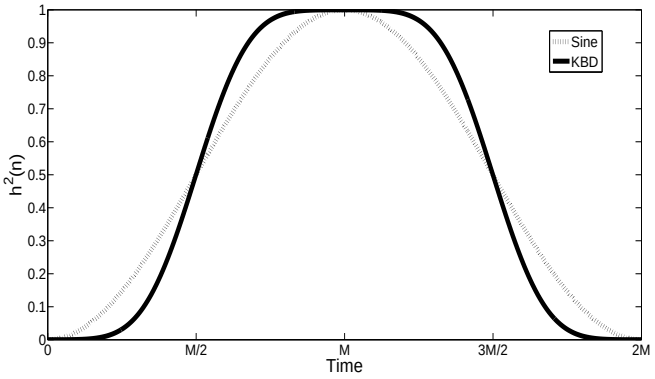


Figure 3.6. Comparison of the squares of sine and KBD windows. The KBD window results in reduced overlap error due to faster tapering.

Since this approximate NMR$^+$ depends only on the MDCT error in the current frame itself, it can be incorporated in an encoder such as the MPEG VM by simple substitution of the usual NMR. Whenever the SF (and hence the MDCT error $\underline{E}_k$) for an SFB is altered, $\hat{\underline{\mathcal{E}}}_k = P_S P^T \underline{E}_k$ is re-computed and the NMR$^+$ value updated.

Multiplication with the $M \times M$ matrix $P_S P^T$ is performed efficiently by recognizing the fact that, for good window choices such as sine and KBD, this matrix has its most dominant elements close to the principal diagonal. This band-like structure of $P_S P^T$ is the result of critically located spectral zeroes in the case of the sine window and very good anti-aliasing (side lobe reduction) properties

in the case of KBD. Therefore for any j, $\hat{\underline{\mathcal{E}}}_k(j)$ is constructed from elements of $\underline{E}_k$ with indices in a very small neighborhood of j. When the sine window is used it can be shown that $\hat{\underline{\mathcal{E}}}_k(j)$ depends exactly on $\underline{E}_k(j+1)$ and $\underline{E}_k(j-1)$. In case of the KBD window 4 to 6 $\underline{E}_k$ coefficients are sufficient to calculate each $\hat{\underline{\mathcal{E}}}_k(j)$. Thus the M multiplications (and additions) to calculate each $\hat{\underline{\mathcal{E}}}_k(j)$ can be reduced to a modest number. Efficient computation of MDST coefficients from MDCT coefficients has been used previously, for example in [21] to estimate the power spectrum of the frame. Since the sine and cosine bases in P_S and P are uniformly spaced in frequency, most of the rows of $P_S P^T$ (except a few at the top and bottom ends) are shifted repetitions of each other enabling efficient storage of the matrix.

3.2.4 Experiments

In experiments we have employed the MPEG VM. The (approximate) modified metric NMR$^+$ can be used in lieu of the NMR in the inner loop of the TLS in the MPEG VM. The two implementations are respectively termed VM-NMR and VM-NMR$^+$. The encoders were constrained to work only in the 'LONG' window mode of AAC (i.e., M was fixed at 1024). 5 audio files each at sampling rate 44.1kHz were encoded at a bit-rate of 48kbps by both methods and with both window choices, sine and KBD. Blind listening tests in the A-B style were conducted with 15 subjects, with access to the original audio file and randomly ordered samples encoded by the two methods when using the same window. They could switch near instantaneously between any of these 3 files. Since the choice of bit-rate is relatively high, the original helps listeners to identify artifacts in either coded sample. They could pick one as preferred or state that they were

unable to decide. The results of the tests are given in Table 3.3. Considerable subjective gains of using the new measure are seen with either window choice. Only in the case of the accordion piece there was no clear preference.

Audio Sample	sine			KBD		
	VM-NMR$^+$	VM-NMR	No Pref	VM-NMR$^+$	VM-NMR	No Pref
Harpsichord	58.33	0	41.66	75	0	25
Organ	50	0	50	66.67	0	33.33
Accordion	25	25	50	41.67	33.33	25
Male German speech	91.67	8.33	0	100	0	0
Female English speech	91.67	8.33	0	91.67	8.33	0

Table 3.3. Subjective comparison tests of VM-NMR and VM-NMR$^+$ with both sine and KBD windows: figures indicate the percentage of listeners who preferred audio encoded using corresponding method.

We note that we are yet to employ the NMR$^+$ in our two-layered trellis. This entails certain difficulties as the distortion in a band now depends on neighboring bands also, due to the MDST domain spreading, and the inner trellis algorithm needs modifications to account for this spreading.

3.2.5 Generalization to other LOT based codecs

We consider here audio coding with generic LOT matrices of dimensions $M \times 2M$. Additionally, let us suppose the forward LOT matrix P of dimensions $M \times 2M$ is decomposable into the form $C'H$ as in (3.5), with the rows of C' being orthogonal basis vectors spanning an M dimensional sub-space of the $2M$ dimensional space. Using a matrix S' with rows as orthogonal basis vectors of the complementary M dimensional sub-space, similar to the definition of the MDST,

we could now define corresponding P_S and hence proceed to a time domain error analysis similar to (3.32). Thus the use of a distortion measure similar to NMR$^+$ is conceivable even in such generic encoders, although perceptual considerations may need to be revisited in light of the actual choice of transform.

3.3 Conclusion

Two modifications to the widely accepted NMR audio distortion measure have been proposed in this chapter.

One stems from the need to better compare distortion between coding bands of different widths on the Bark scale, and involves scaling the usual NMR in each SFB by an appropriate scaling factor based on the Bark width of the SFB. This in turn leads to a better comparison between different window choices, which were a focus of the optimization by delayed decisions described in Chapter 2. Synthetic experiments support the choice of this correction factor to NMR. When the improved distortion measure (NMR-BC) is used in place of NMR in the MPEG verification model, or in the MMNMR minimizing two-layered trellis approach of Chapter 2 a preference is indicated for audio encoded in light of the new measure. The new measure seems to bring out a better comparison between window choices and thus avoid artifacts notably due to superfluous SHORT window selection.

The second modification stems from the fact that distortion metrics for audio coding based solely in the MDCT domain of a frame are invariant to necessary windowing and overlap-add operations at the decoder. An analysis of the time domain error of a frame reveals that the corresponding error components are orthogonal to the MDCT basis vectors. An enhanced distortion measure NMR$^+$

is suggested that incorporates these components via MDST domain analysis. Subjective tests, using a simplified version of this metric accounting only for the windowing effects, evidence a preference for audio encoded by employing this modification. The improved metric captures the magnitude of the frequency domain error rather than its projection onto the cosine basis vectors of MDCT.

Chapter 4

Delayed Decoding of Predictively Encoded Sources

In Chapter 2 we considered the application of encoding delay to optimize decisions in an audio encoder. Chapter 3 extrapolated on this research and proposed modifications to the audio distortion metric itself, so that the encoding decisions are optimal even in a subjective sense. We now shift gears and focus on the decoder end of the compression chain. We consider the application of delay at the decoder, to collect coded information about future frames/samples, and exploit correlations (if any) of the current frame with such information, to optimize its reconstruction. Since the existence of such correlations is a pre-requisite to apply this technique, the AAC setting of Chapter 2, where such correlation between frames is almost depleted by the use of transform coding, is not appropriate. Hence we consider here the scenario of predictive coding systems, where a correlation model is explicitly assumed. We thus propose in

this chapter optimal delayed decoders for predictively encoded sources.

Predictive coding is widely employed for various signal compression applications including the H.264 standard for motion compensated video coding [85], the G.726 standard for speech coding via adaptive differential pulse code modulation (ADPCM) [47], continuously variable slope delta modulation used in the hands-free profile of Bluetooth devices [1], etc. When applied to a sequence of correlated signal samples, redundancy in the current sample is removed by predicting it from past coded samples, and encoding only the innovation, or prediction residual. For simplicity we use "temporal" terminology, but the proposed ideas are equally applicable to spatial or other types of correlation. The development of predictive coding schemes frequently assumes an AR model of the source [6], [12], [20], [25], [27], [37], [38], [40], [41], [77]. Consider, for now, a first-order AR (or Markov) model. The source consists of a zero-mean stationary sequence $\{x_n\}$ of real-valued random variables with,

$$x_n = \rho x_{n-1} + z_n \ . \tag{4.1}$$

The random variables $\{z_n\}$ are independent and identically distributed (i.i.d), with pdf $p_Z(z)$, and are referred to as the innovations of the process. The correlation coefficient of adjacent samples is ρ. We consider predictive coding of this source using a DPCM scheme [27], [32], [37], [38], [41]. For example, motion compensated video coding effectively performs inter-frame (temporal) prediction of spatial transform coefficients, and can be modeled as DPCM. The DPCM encoder generates a prediction $\tilde{x}_n$, based on prior reconstructions, and subtracts it from the current sample x_n to generate the prediction error e_n, which is quantized using a scalar quantizer $\mathcal{Q}$. This quantizer is specified by the mappings $f_{\mathcal{Q}}(x) : \mathbb{R} \to \mathbb{I}$, and $g_{\mathcal{Q}}(x) : \mathbb{I} \to \mathbb{R}$, where $\mathbb{R}$ is the real line, and $\mathbb{I}$, a countable

index set. The quantization index $i_n = f_{\mathcal{Q}}(e_n)$ is entropy coded and sent to the decoder, which generates $\hat{x}_n = \tilde{x}_n + g_{\mathcal{Q}}(i_n)$. At high rate, $\hat{x}_{n-1} \approx x_{n-1}$, and the prediction

$$\tilde{x}_n = \rho \hat{x}_{n-1} \tag{4.2}$$

is optimal. Even at low bit-rates this form of the predictor is commonly employed.

In an AR source model (such as (4.1)), the present sample x_n is not only correlated with the past, but also with the future, i.e., with $\{x_l\}_{l>n}$. At high rate, the prediction error $e_n \approx z_n \; \forall n$ and hence the indices $\{i_n\}$ are approximately i.i.d. In this case future indices $\{i_l\}_{l>n}$ provide no additional information on x_n. In practice, bit-rates are limited and such approximations do not hold, in which case these future indices do contain information on x_n, which could potentially be exploited at the decoder to improve reconstruction. Obviously, this entails decoding delay. Prior research that exploits this fact includes the interpolative DPCM (IDPCM) [77], and smoothed DPCM (SDPCM) [20] approaches, both of which *smooth* (i.e., filter) the regular DPCM outputs $\{\hat{x}_n\}$ with a suitable *non-causal* post-processor to generate refined estimates. The basic paradigm of these schemes is summarized in Fig. 4.1a. While more details are provided in Sec. 4.1, suffice it to say that the design of the post-processor in either scheme is heuristic and depends on assumptions, for instance about the quantizer resolution, which preclude performance guarantees relative to regular (zero-delay) DPCM. A more significant shortcoming is that by merely filtering $\{\hat{x}_n\}$ (see Fig. 4.1a), while disregarding the indices $\{i_n\}$, as well as knowledge about the exact function of modules such as the quantizer and predictor, these methods under-utilize the information available at the decoder. The following simple argument illustrates this suboptimality: The decoder knows $(\tilde{x}_n, i_n, \mathcal{Q})$ that determine the effective

quantization interval $\mathcal{I}_n = \{x \in \mathbb{R} : f_{\mathcal{Q}}(x - \tilde{x}_n) = i_n\}$ in which x_n *must* lie, but simple smoothing of $\{\hat{x}_n\}$ may produce a reconstruction that lies outside $\mathcal{I}_n$.

In contradistinction, we propose an ET approach to delayed decoding that *optimally* combines all the information (i.e., indices $\{i_l\}_{l \leq n+L}$ or equivalently the intervals $\{\mathcal{I}_l\}_{l \leq n+L}$) available at the decoder, for a given delay (or 'look-ahead') L, in a recursively calculated conditional pdf, to guarantee the best reconstruction of the current sample x_n. Fig. 4.1b contrasts the proposed scheme with prior work. Although derived in the framework of first order AR sources, the proposed technique is easily generalized to higher order processes. While the applicability of this optimal delayed decoder is not limited to any particular form of the predictor, in case of the commonly employed 'matched' predictor (e.g., (4.2) for the source in (4.1)) it motivates an approximate decoder whose reconstruction is of the form $\tilde{x}_n + c(\{i_l\}_{n-L' \leq l \leq n+L})$, and is implementable as a time-invariant codebook over the 'window' of indices $\{i_l\}_{n-L' \leq l \leq n+L}$, which has the obvious benefit of low-complexity decoding even compared to linear smoothing. This codebook approach is asymptotically (in the memory L') optimal, and in the case of first order AR processes, is observed in experiments to provide near-optimal performance even with $L' = 0$. Simulation results demonstrate that both (the optimal and codebook) methods substantially outperform IDPCM and SDPCM. The increased storage necessitated by the delayed decoding codebook is considerably mitigated by selective merging of codebook entries, that conditionally maps different index values to the same reconstruction. The codebook approach naturally paves the way to a training-set based design, that is particularly attractive in the case of higher order processes, where the recursive evaluation of conditional pdfs (employed in the optimal method) becomes overly complex due to high dimen-

(a) IDPCM [77] or SDPCM [20] (b) Proposed optimal delayed decoder

Figure 4.1. Prior approaches merely smooth the regular DPCM reconstructions; the Optimal Delayed Decoder exploits all available information

sion. Corresponding results with second and third order processes demonstrate the efficacy of this approach.

Preliminary results of this work have been reported in [64] and in more detail in [59]. We note that the ET approach in [75] to (zero-delay) scalable predictive coding, that efficiently uses base-layer quantization interval information for improved enhancement layer prediction, was an early inspiration for the current work. Related prior research includes [41] that derives asymptotic bounds on delayed decoding gains for a first order gaussian AR process, assuming that the quantization noise is white, gaussian, and uncorrelated with the source, and [54] that provides an information theoretic analysis under the same setting. An optimal delta modulation (zero-delay) system that recursively estimated the pdf of the current sample conditioned on past outputs has been described in [25], while an iterative optimization of a delta modulator, where the prediction is conditioned on a finite set of past indices, has been proposed in [12]. The general framework in [32] for alphabet constrained compression, that extends earlier work by Fine [29], subsumes delayed decoding as a special case, although no practical design algorithm was proposed. Specific to a gaussian AR source, and under certain assumptions about quantization effects, improved DPCM performance was demonstrated in [37] by modifying the encoder to include a rate-distortion opti-

mized non-causal pre-filter, and a corresponding modified predictor. Extension to include a decoder post-filter was presented in [38], although it is noted that most of the gains are due to the encoder side pre-filtering. We emphasize that the focus here is on a zero-delay encoder, with latency allowed at the decoder.

Finally we note that predictive coding is particularly attractive for applications where the framing delay or complexity of competing transform-based approaches are undesirable/unacceptable: for instance, low-delay audio coding [76], audio/speech compression for bluetooth headsets [1], low power image sensors [53], etc. The delayed decoding approaches presented here employ a delay of a few samples (3-4), as opposed to an entire frame, and are thus attractive for these low latency scenarios. In applications such as the image sensor, although the encoder may be simple, the decoder may be endowed with considerable computational capabilities, which can be well exploited by delayed decoding.

The rest of this chapter is organized as follows. Sec. 4.1 describes the prior work [20] and [77] in more detail. The optimal delayed decoder is derived in Sec. 4.2, followed by its low complexity approximation, the codebook method in Sec. 4.3. Results for first order sources, a method for codebook size reduction, and generalizations to higher order processes are provided in Sec. 4.4, Sec. 4.5, and Sec. 4.6 respectively. Finally Sec. 4.7 describes an example encoder modification that employs for prediction, reconstructions of past samples obtained via the proposed delayed decoder.

4.1 Preliminaries

IDPCM [77] and SDPCM [20] adopt a smoothing approach to exploit future information at the decoder. Either approach designs a non-causal filter, and applies it to the "regular", zero-delay, DPCM reconstructions. We briefly describe here the design of this non-causal filter by both approaches, and highlight its limitations.

4.1.1 Interpolative DPCM

IDPCM [77] determines the set of coefficients b_l, $-L' \leq l \leq L$, $l \neq 0$, that minimize $E[(x_n - \sum_{l=-L', l\neq 0}^{L} b_l x_{n+l})^2]$, to obtain the smoothed estimate of x_n:

$$\hat{x}_n^{idpcm} = \sum_{l=-L'}^{-1} b_l \hat{x}_{n+l}^{idpcm} + \sum_{l=1}^{L} b_l \hat{x}_{n+l} \tag{4.3}$$

where $\hat{x}_n$ denotes the regular DPCM reconstruction. Note that (4.3) implies smoothing of $\hat{x}_n$ using the non-causal IIR filter specified by the coefficients b_l. It is shown in [77] that b_l are determined by the auto-correlation matrix, irrespective of the bit-rate, or process distributions. Specifically for the first order process (4.1), $b_{-1} = b_1 = \frac{\rho}{1+\rho^2}$, and for $l \notin \{-1, 1\}$, $b_l = 0$, i.e., the look-ahead L is automatically restricted to 1. In general, the maximum possible look-ahead L in IDPCM (and look-back L') is restricted to the process order, although as will become evident in Sec. 4.4 the potential gains in looking further ahead can be substantial. Similar to examples in [77], henceforth in this chapter we assume L' of the IDPCM smoother is fixed to its maximum value, i.e., the process order. IDPCM as described here is purely a decoder enhancement. A second method (applicable to processes with order greater than 1), that modifies the encoder to

introduce the smoothed IDPCM reconstructions back into the prediction loop, is also proposed in [77]. More on this in Sec. 4.7.

4.1.2 Smoothed DPCM

In SDPCM [20] a Kalman filtering-based fixed-lag smoother [66] is used. The AR process provides the 'plant' model which, in case of the first order AR process (4.1) and for a fixed-lag (i.e., look-ahead) L, is:

$$\underbrace{\begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-L} \end{bmatrix}}_{\mathbf{x}_n} = \underbrace{\begin{bmatrix} \rho & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Psi} \mathbf{x}_{n-1} + \begin{bmatrix} z_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{4.4}$$

The above is easily generalized to higher order processes. The quantization process provides the 'observation' model:

$$\hat{x}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \mathbf{x}_n + w_n, \tag{4.5}$$

where w_n is the DPCM quantization noise, assumed to be white. Kalman filtering is now used to obtain the best estimate $\hat{\mathbf{x}}(n|n)$ of $\mathbf{x}_n$, given observations up to time n, and the smoothed estimate

$$\hat{x}_n^{sdpcm} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \hat{\mathbf{x}}(n+L|n+L) \tag{4.6}$$

Note that invariably w_n is correlated with x_n, especially at low bit-rates, and since $\{x_n\}$ itself is a temporally correlated sequence, the assumption that quantization errors $\{w_n\}$ are white is invalid. Although the Kalman filter is generally adaptive, for the time-invariant model given by (4.4) and (4.5), it can be shown that it has

a 'steady state' [66] when $|\rho| < 1$, implying a time-invariant smoother. This is in general the case when the driving process is stationary, and the quantizer does not change with time. SDPCM, like IDPCM, ignores the process distribution. But unlike IDPCM, it does incorporate knowledge of the bit-rate through the observation (or quantization) noise variance employed in the Kalman recursion [66].

As illustrated by the simple argument in the introductory part of this chapter, both SDPCM and IDPCM, by merely smoothing the regular DPCM reconstructions, disregard a substantial amount of information available to the decoder in the form of quantization indices $\{i_n\}$, which is efficiently utilized by the proposed optimal delayed decoder discussed in the following section.

4.2 Optimal Delayed Decoder

In this section we formulate the optimal delayed decoder for the first order AR source (4.1) encoded via DPCM. Extension to higher order processes is discussed in Sec. 4.6. While the general derivation assumes no particular form of the predictor, a computationally efficient simplification is also presented for the special case when the predictor is matched to the source.

4.2.1 Arbitrary predictor: the general case

Henceforth, with respect to any sequence $\{a_n\}$, the notation $\{a_l\}_m^k$, $\{a_l\}_m$, and $\{a_l\}^k$ denote, respectively, the truncated sequences $\{a_l : m \leq l \leq k\}$, $\{a_l : l \geq m\}$, and $\{a_l : l \leq k\}$. We do not alter the DPCM encoder in any way. Thus, the in-

dices $\{i_l\}^n$ known at the decoder, determine the effective quantization intervals $\{\mathcal{I}_l\}^n$ exactly, $\{i_l\}^n \Leftrightarrow \{\mathcal{I}_l\}^n$. Let mean squared error (MSE) be the distortion criterion. Then the *optimal estimate* $\hat{x}_n^*$, for a fixed look-ahead L, is given by

$$\hat{x}_n^* = E[x_n|\{i_l\}^{n+L}] = E[x_n|\{\mathcal{I}_l\}^{n+L}] \ , \tag{4.7}$$

where expectation is over the pdf $p(x_n|\{\mathcal{I}_l\}^{n+L})$ conditioned on *all information available* at the decoder. Thus, $\hat{x}_n^*$ can be obtained if this density is known.

We use the streamlined notation $p(\cdot)$ to denote any pdf or probability, and add a subscript whenever the interpretation is not evident from the arguments. Note that the above conditional pdf automatically limits the optimal estimate to the interval $\mathcal{I}_n$. We now write

$$p(x_n|\{\mathcal{I}_l\}^{n+L}) = \frac{p(x_n|\{\mathcal{I}_l\}^n)p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)}{\int p(x_n|\{\mathcal{I}_l\}^n)p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)dx_n} \ . \tag{4.8}$$

Unless otherwise indicated, integrals are over $\mathbb{R}$. The above equality follows from Bayes' rule, and the Markov property of the process (4.1): given x_n, the probability of events $\{\mathcal{I}_l\}_{n+1}^{n+L}$, is independent of any other information from the past (i.e., $\{\mathcal{I}_l\}^n$). Note that $p(x_n|\{\mathcal{I}_l\}^n)$ is the pdf of x_n conditioned on all information up to the current instant n. An optimal *zero-delay* decoder estimates x_n simply as the mean of this pdf (henceforth referred to as the *zero-delay pdf*). The optimal *delayed* decoder weighs the zero-delay pdf with $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$ representing the conditional probability, given x_n, of the known future outcomes. Hence the composite pdf $p(x_n|\{\mathcal{I}_l\}^{n+L})$ of (4.8) incorporates all known information up to the fixed delay L. The estimate of x_n is then $\hat{x}_n^*$ of (4.7). We next provide recursion formulas to calculate $p(x_n|\{\mathcal{I}_l\}^n)$ and $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$.

The zero-delay pdf at time $n-1$, denoted $p(x_{n-1}|\{\mathcal{I}_l\}^{n-1})$, subsumes in it all information received at the decoder up to that time. The zero-delay pdf at time

n, or $p(x_n|\{\mathcal{I}_l\}^n)$, is recursively obtained as follows. The first step is to find the pdf of x_n conditioned on information available up to time $n-1$:

$$\begin{aligned} p(x_n|\{\mathcal{I}_l\}^{n-1}) &= \int p(x_{n-1}, x_n|\{\mathcal{I}_l\}^{n-1})dx_{n-1} \\ &= \int p(x_{n-1}|\{\mathcal{I}_l\}^{n-1})p(x_n|x_{n-1})dx_{n-1} \\ &= \int p(x_{n-1}|\{\mathcal{I}_l\}^{n-1})p_Z(x_n - \rho x_{n-1})dx_{n-1} \end{aligned} \tag{4.9}$$

The second equality is due to the Markov property of the process (4.1). Employing the fact that z_n is independent of x_{n-1} and using (4.1) yields (4.9). Now, at time n, the index i_n becomes available, that provides the additional information that x_n lies in the interval $\mathcal{I}_n$. This information is absorbed into the above pdf via appropriate conditioning to obtain:

$$p(x_n|\{\mathcal{I}_l\}^n) = \begin{cases} \frac{p(x_n|\{\mathcal{I}_l\}^{n-1})}{\int_{\mathcal{I}_n} p(x_n|\{\mathcal{I}_l\}^{n-1})dx_n} & x_n \in \mathcal{I}_n \\ 0 & else \end{cases} \tag{4.10}$$

A close inspection of (4.9) shows that it is in effect a convolution. In practice, discretized versions of the densities are used, with this convolution efficiently implemented using an FFT, combined with an interpolation/re-sampling operation between updates, for the required axial scaling. This recursive update of the zero-delay pdf is similar to the recursion employed in [25] to obtain the optimal binary quantizer in a delta modulator.

Next, the probability of future events is derived via the backward recursion enumerated below. Say, at time $n+1$ we have the probability $p(\{\mathcal{I}_l\}_{n+2}|x_{n+1})$ of all future outcomes given x_{n+1}. The following recursive update provides the

corresponding probability, $p(\{\mathcal{I}_l\}_{n+1}|x_n)$, at time n.

$$\begin{aligned} p(\{\mathcal{I}_l\}_{n+1}|x_n) &= \int_{\mathcal{I}_{n+1}} p(\{\mathcal{I}_l\}_{n+2}, x_{n+1}|x_n)dx_{n+1} \\ &= \int_{\mathcal{I}_{n+1}} p(\{\mathcal{I}_l\}_{n+2}|x_{n+1}, x_n)p(x_{n+1}|x_n)dx_{n+1} \\ &= \int_{\mathcal{I}_{n+1}} p(\{\mathcal{I}_l\}_{n+2}|x_{n+1})p_Z(x_{n+1} - \rho x_n)dx_{n+1}. \end{aligned} \tag{4.11}$$

In other words, the above recursion is a 'one sample retreat' procedure: each use of this procedure implies a step backwards in time ($n+1$ to n), which thus appends a new event ($\mathcal{I}_{n+1}$) to the existing list of future events ($\{\mathcal{I}_l\}_{n+2}$), and by (4.11) accommodates it appropriately into the probability calculation. But for a given look-ahead L, the future information available to reconstruct x_n is limited to $\{\mathcal{I}_l\}_{n+1}^{n+L}$. The only knowledge about samples x_l, $l > n+L$, is that they are on the real line. Hence, effectively $\mathcal{I}_l = \mathbb{R}$, $l > n+L$ (obviously with probability one). Thus we initialize $p(\{\mathcal{I}_l\}_{n+L+1}|x_{n+L}) = 1$ and employ (4.11) L times to 'retreat' from time $n+L$ to n, and obtain the probability $p(\{\mathcal{I}_l\}_{n+1}|x_n) = p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$ of the known future outcomes given x_n. Application of a suitable indicator function converts (4.11) to a convolution, which is then efficiently implemented via FFT.

The optimal delayed decoder, for a look-ahead L, is thus:

Optimal Delayed Decoder

At time $n+L$

1. Decode (as in regular DPCM) index i_{n+L}, and use $\tilde{x}_{n+L}$ to obtain $\hat{x}_{n+L}$, and the interval $\mathcal{I}_{n+L}$.

2. Update pdf $p(x_{n-1}|\{\mathcal{I}_l\}^{n-1})$ to $p(x_n|\{\mathcal{I}_l\}^n)$, that combines all information available up to time n.

3. Obtain probability $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$ as a function in x_n, that combines information about all available future outcomes (relative to time n).

4. Use (4.7) and (4.8) to obtain the optimal estimate $\hat{x}_n^*$.

Note that $p(x_n|\{\mathcal{I}_l\}^n)$ needs to be suitably initialized, say at $n = 0$. If $|\rho| < 1$, the effect of this initialization decays with time.

4.2.2 The matched predictor: a special case

While Step 2 of the optimal delayed decoder is a one step update ($n-1$ to n), Step 3 is an L step recursion ($n+L$ to n), that is repeated every instant n. The latter could entail considerable computation for large look-aheads. For the special case of the matched predictor (4.2), and with respect to the assumed time invariant DPCM scheme, a simplification can be effected. With look-ahead $L = 1$, the substitution $x_{n+1} = e + \tilde{x}_{n+1}$ in (4.11) (where $\tilde{x}_{n+1}$ is given by (4.2)) yields

$$p(\mathcal{I}_{n+1}|x_n) = \int_{\mathcal{I}_{\mathcal{Q}}(i_{n+1})} 1 \cdot p_Z(e - \rho(x_n - \hat{x}_n))de \ . \tag{4.12}$$

Here $\mathcal{I}_{\mathcal{Q}}(i) = \{x \in \mathbb{R} : f_{\mathcal{Q}}(x) = i\}$ are *time-invariant* intervals characteristic of the quantizer. Define the function $\Lambda(x, i) : \mathbb{R} \times \mathbb{I} \to \mathbb{R}$ as:

$$\Lambda(x, i) = \int_{\mathcal{I}_{\mathcal{Q}}(i)} p_Z(e - \rho x)de \ . \tag{4.13}$$

Note that the function $\Lambda(x, i)$ is time invariant, and completely determined by the quantizer, and innovation pdf. Hence, $p(\mathcal{I}_{n+1}|x_n)$ is the above function evaluated at $(x_n - \hat{x}_n, i_{n+1})$. We now make the following claim:

Claim 1: In case of the matched predictor (4.2), a time invariant quantizer $\mathcal{Q}$,

and the stationary process (4.1), $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$ can be obtained by evaluation of a time invariant function of the form $\Lambda(x, \{i_l\}_0^{L-1})$, at $(x_n - \hat{x}_n, \{i_l\}_{n+1}^{n+L})$, $\forall L > 0$.

Proof sketch: Already shown for $L = 1$. Complete by induction on L.

Thus the L-step recursion in Step 3 of the optimal delayed decoder can *equivalently* be simplified by constructing a codebook containing the functions $\Lambda(x, \{i_l\}_0^{L-1})$, i.e., each element of the codebook is a function (of a fixed shape along the real line) indexed by $\{i_l\}_0^{L-1}$. Once the indices $\{i_l\}_{n+1}^{n+L}$ are collected, the corresponding function $\Lambda(x, \{i_l\}_{n+1}^{n+L})$ is read, and shifted by $\hat{x}_n$, to obtain $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$.

4.3 Codebook-based Delayed Decoder

Although the optimal estimate (4.7) is conditioned on all known information, $\{i_l\}^{n+L}$, the optimal delayed decoder of Sec. 4.2 needed, at each instant n, the exact knowledge of indices $\{i_l\}_n^{n+L}$ only: information about the past is embedded in the zero-delay pdf, and the prediction $\tilde{x}_n$, which are updated every instant. Hypothetically, as an alternative to these recursive calculations, one could envision a *time-invariant* codebook-based decoder that continuously collects indices, maps index sequences to reconstructions given by (4.7) stored in a look-up table, and simply reads out the optimal estimate. While such an approach would considerably simplify the computational requirements of optimal delayed decoding, the growing length of the sequence $\{i_l\}^{n+L}$ and associated growth in memory requirements render strict optimality infeasible. With fixed storage constraints, only a finite window of indices $\{i_l\}_{n-L'}^{n+L}$ is available every instant. However, the decoder still has access to $\tilde{x}_n$ which, by virtue of the prediction loop, is usually a

function of *all* past indices $\{i_l\}^{n-1}$, and hence contains some additional information that is not present in the window of indices $\{i_l\}_{n-L'}^{n+L}$. The optimal estimate with finite index memory is thus:

$$\hat{x}_n^*(L') = E[x_n|\tilde{x}_n, \{i_l\}_{n-L'}^{n+L}] \ . \tag{4.14}$$

Note that $\tilde{x}_n$ is a real number and not an index, and a look-up table whose entries implement (4.14) cannot be constructed. However, AR processes do not contain periodic components, and inter-sample correlation generally decreases with increasing separation ($|\rho| < 1$ ensures this for (4.1)). Predictors are either closely matched to the source or at least correspond to a stable system, and thus the current sample's prediction is influenced only minimally by indices that occurred much earlier in the sequence (easily verified for the matched predictor (4.2)). These two factors imply that, in practice, a long window of indices $\{i_l\}_{n-L'}^{n+L}$ should subsume within it almost all the information in $\tilde{x}_n$ and suffice to approximate (4.14) closely [1]. Therefore, the following approximate estimate of x_n can be stored in a codebook:

$$\hat{x}_n^*(L') \approx E[x_n|\{i_l\}_{n-L'}^{n+L}] \ . \tag{4.15}$$

Stationarity of the process ensures time invariance of the codebook. A caveat is that, depending on the degree of correlation, a good approximation may require large L', and hence a gigantic codebook (whose size grows as $K^{L'}$, where K is the number of cells in the quantizer). Bello et al. [12] employ such a codebook in their design of a zero-delay ($L = 0$) delta modulation system. However, for

[1] Although, absent quantization, temporally decreasing correlation implies x_n is nearly independent of x_{n-m}, $m \gg 0$, when quantization is present the interval $\mathcal{I}_n$ in which x_n lies, and the index i_n, are influenced by the prediction $\tilde{x}_n$, and through it linked to past events. So stability of the prediction loop is *also* a requirement to ensure that the inherent system memory is limited.

the case of the matched predictor, the insight provided by the optimal delayed decoder of Sec. 4.2 enables a very good approximation that converts (4.14) to a look-up table-based estimate without recourse to a large L'.

Specifically, we show that in case of the matched predictor, the optimal estimate in (4.7) is of the form

$$\hat{x}_n^* = \tilde{x}_n + c'\Big(p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1}), \{i_l\}_{n-L'}^{n+L}\Big) \tag{4.16}$$

where $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$ is the *pdf of the prediction error* at time $n-L'$ conditioned on the indices $\{i_l\}^{n-L'-1}$ (i.e., all past information relative to time $n-L'$). While the proof of the above equation is deferred to the Appendix A.1, its import is the following: given the density $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$, any other effect of the indices $\{i_l\}^{n-L'-1}$ on the optimal estimate is completely subsumed in $\tilde{x}_n$; the value of $c'(\cdot)$ is then solely determined by the values of the remaining indices $\{i_l\}_{n-L'}^{n+L}$. Thus, if the prediction error pdf $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$ is approximated by a *time-invariant density* (for instance by assuming that it is simply $p_Z(e)$, which it indeed asymptotically approaches at high bit-rates), then the term $c'(\cdot)$ is simply a time-invariant term of the form $c(\{i_l\}_0^{L+L'})$, i.e., its values can be stored in a codebook indexed by $\{i_l\}_0^{L+L'}$. Thus, the optimal finite memory estimate in (4.14) can be approximated as:

$$\hat{x}_n^*(L') \approx \tilde{x}_n + c(\{i_l\}_{n-L'}^{n+L}) \ . \tag{4.17}$$

At every instant n the indices $\{i_l\}_{n-L'}^{n+L}$ are collected, the term $c(\{i_l\}_{n-L'}^{n+L})$ read from the codebook, and added to $\tilde{x}_n$.

To understand (4.16) better, consider the optimal delayed decoder of Sec. 4.2. At every instant n information from the past is incorporated into the optimal estimate in two different ways: (a) via the pdf $p(x_n|\{\mathcal{I}_l\}^{n-1})$ in (4.9), and

(b) via the intervals $\{\mathcal{I}_l\}_n^{n+L}$ in (4.10) or (4.11) whose limits depend not just on the indices $\{i_l\}_n^{n+L}$, but also on the prediction $\tilde{x}_n$ (and hence on past indices). Since our focus in this section is a codebook-based approximation in lieu of recursive pdf calculations, let us say we simply approximate $p(x_n|\{\mathcal{I}_l\}^{n-1})$, that is recursively obtained in the optimal decoder, by some fixed density $p_A(x_n)$. Given this density, the required estimate is obtained by incorporating the interval information $\{\mathcal{I}_l\}_n^{n+L}$ via (4.10) and (4.11). Note that although $p(x_n|\{\mathcal{I}_l\}^{n-1})$ is itself approximated, the intervals $\{\mathcal{I}_l\}_n^{n+L}$ and hence the indices $\{i_l\}_n^{n+L}$ are still optimally utilized. Hypothetically, if these intervals had been solely determined by indices $\{i_l\}_n^{n+L}$, for any combination $\{i_l\}_0^L$ the corresponding intervals would be time-invariant. Hence (4.10) and (4.11) would correspond to time-invariant calculations, and a codebook could store the appropriate reconstructions for each combination $\{i_l\}_0^L$. But in reality the intervals $\{\mathcal{I}_l\}_n^{n+L}$ also depend on past indices through the prediction, and hence a time-invariant approach is feasible only if the information embedded in these intervals due to past indices can be somehow separated/decoupled from that due to the indices $\{i_l\}_n^{n+L}$. The latter could then be utilized to build a codebook. In case of the matched predictor, (4.16) indicates that this is exactly achievable. In fact it demonstrates this in a more general setting where a codebook over the indices $\{i_l\}_{n-L'}^{n+L}$ is the objective.

In (4.16) the effect of past indices $\{i_l\}^{n-L'-1}$ on the limits of the intervals $\{\mathcal{I}_l\}_{n-L'}^{n+L}$ is completely subsumed in the term $\tilde{x}_n$. The remainder of the information that determines these intervals, i.e., $\{i_l\}_{n-L'}^{n+L}$, is incorporated *optimally* into the estimate via $c'(\cdot)$. Note that this term depends only indirectly on $\{i_l\}^{n-L'-1}$ through the density $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$. Therefore, approximating this density converts $c'(\cdot)$ to the look-up table $c(\cdot)$ without recourse to any loss of optimality

in utilizing $\{i_l\}_{n-L'}^{n+L}$, i.e., the sub-optimality is solely due to the approximation of the past via $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$. *Asymptotically, as L' increases, the set of optimally incorporated indices $\{i_l\}_{n-L'}^{n+L}$ closely fits $\{i_l\}^{n+L}$, thus $c(\cdot)$ tends to $c'(\cdot)$, and the RHS of* (4.17) *is indeed the optimal estimate in* (4.16).

The formulation (4.16) separates out $\tilde{x}_n$, the component of the optimal estimate that is highly correlated in time and depends on a long index history, from $c'(\cdot)$, which is an innovation-like component that is relatively insensitive to the past. This insensitivity of $c'(\cdot)$ to the past is itself a result of isolating in it only that information in $\{\mathcal{I}_l\}_{n-L'}^{n+L}$ that comes from the indices $\{i_l\}_{n-L'}^{n+L}$. These factors result in a much smaller L' (and hence smaller look-up table $c(\cdot)$) compared to a codebook that implements (4.15).

Note that independent of the predictor structure,

$$\hat{x}_n^* = \tilde{x}_n + E[e_n|\{i_l\}^{n+L}], \tag{4.18}$$

since $\tilde{x}_n$ is always a deterministic function of past indices, and

$$\hat{x}_n^*(L') = \tilde{x}_n + E[e_n|\tilde{x}_n, \{i_l\}_{n-L'}^{n+L}]. \tag{4.19}$$

However, whether or not the effect of the indices $\{i_l\}_{n-L'}^{n+L}$ is separable from the past, as is the case in (4.16), depends on the predictor. In other words, whether the term $E[e_n|\tilde{x}_n, \{i_l\}_{n-L'}^{n+L}]$ in the above equation is well approximated by a time-invariant $c(\{i_l\}_{n-L'}^{n+L})$ for *small* L', is predictor dependent.

4.3.1 Design of the codebook: known density information

The codebook entries $c(\{i_l\}_{n-L'}^{n+L})$ in (4.17) require a suitable approximation of $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$ by a time-invariant pdf. A good approximation is simply

$$p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1}) \approx p_E(e) \tag{4.20}$$

where $p_E(\cdot)$ is the *stationary marginal prediction error density*. While a detailed exposition can be obtained in [6], [27], [40] we only briefly describe this pdf here. Consider the issue of predictive quantizer design [27]. If the quantizer thresholds (i.e., $\mathcal{I}_{\mathcal{Q}}(\cdot)$, or equivalently $f_{\mathcal{Q}}(\cdot)$) are fixed, the reconstructions $g_{\mathcal{Q}}(i_n)$ are obtained as

$$g_{\mathcal{Q}}(i_n) = E[e_n|i_n] = \int e_n p(e_n|i_n) de_n = \frac{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_n)} e_n p(e_n) de_n}{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_n)} p(e_n) de_n} \ ,$$

where $p(e_n)$ is the marginal (unconditional) prediction error density *at time* n. But e_n (and thus $p(e_n)$) is itself dependent on $\{g_{\mathcal{Q}}(i_l)\}^{n-1}$, through the prediction $\tilde{x}_n$. Thus, to obtain a time invariant $g_{\mathcal{Q}}(\cdot)$ a recursive optimization needs to be performed, which at convergence yields a corresponding stationary marginal prediction error density $p_E(e)$.

With approximation (4.20), $c'(p_E(\cdot), \{i_l\}_{n-L'}^{n+L})$ in (4.16) depends only on the indices $\{i_l\}_{n-L'}^{n+L}$, and is unaffected by $\tilde{x}_n$. Hence assume $\tilde{x}_{n-L'} = 0$, and obtain the interval sequence $\{\mathcal{I}_l\}_{n-L'}^{n+L}$, and prediction $\tilde{x}_n$ that correspond to indices $\{i_l\}_{n-L'}^{n+L}$, by recursive application of (4.2). Now (4.20) implies

$$p(x_{n-L'}|\{\mathcal{I}_l\}^{n-L'-1}) \approx p_E(x_{n-L'} - \tilde{x}_{n-L'}) = p_E(x_{n-L'}) \ . \tag{4.21}$$

Given the above initialization the forward recursion of the optimal delayed decoder can be used to obtain $p(x_n|\{\mathcal{I}_l\}^n)$, the backward recursion for $p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n)$, and through (4.7), (4.8), and (4.16), we can obtain $c'(p_E(\cdot), \{i_l\}_{n-L'}^{n+L}) = c(\{i_l\}_{n-L'}^{n+L})$.

It should be noted that the derivation of $p_E(e)$ via the recursive approach in [27] as well as usage of the forward and backward recursions in the above codebook design technique require explicit knowledge of the innovation pdf.

4.3.2 Design of the codebook: training-set method

Comparing (4.16) and (4.18) we see that $c(\{i_l\}_{n-L'}^{n+L})$ is in fact an estimate of the prediction error at time n, given a window of neighboring indices, and is easily obtained by employing a training-set of prediction errors. Such a strategy is particularly useful when the innovation pdf is not explicitly known, and instead a training set of the source is available. The procedure automatically designs the codebook to be concurrent with the underlying unknown process statistics. Alternatively, this approach could also be employed if the method in Sec. 4.3.1 based on the recursive evaluation of formulae becomes too complex:

1. Generate a long source sequence $\{x_n\}_0^N$ according to the given source model (alternatively, a training set of the source might be directly available).

2. Encode the source using the given DPCM encoder. Collect the prediction error and index at each instant, to build $\{e_n\}_0^N$ and $\{i_n\}_0^N$

3. For each combination $\{i'_l\}_0^{L'+L}$, $i'_l \in \mathbb{I}$, set

$$c(\{i'_l\}_0^{L+L'}) = \frac{\sum_k e_{k+L'}}{\sum_k 1}, \ k \text{ s.t. } \{i_{k+l} = i'_l, 0 \leq l \leq L + L'\}$$

It should be noted that in general not all combinations $\{i'_l\}_0^{L'+L}$ occur with the same frequency in the index sequence. For a fixed N, some may not appear at all in the training-set. Such combinations, due to their low probability, only mildly

influence the distortion during operation, and the corresponding codebook entries are just set to the associated zero-delay prediction error reconstruction $g_{\mathcal{Q}}(i'_{L'})$.

4.4 Results for First Order AR Sources

Experiments described in this section focus on first order AR sources that conform to the model (4.1), and include cases where the innovations are drawn from gaussian or Laplace distributions. For instance, the transform coefficients of prediction residual blocks in video coding, are modeled well by the Laplace distribution [10]. The distribution parameters are adjusted to maintain unit source variance.

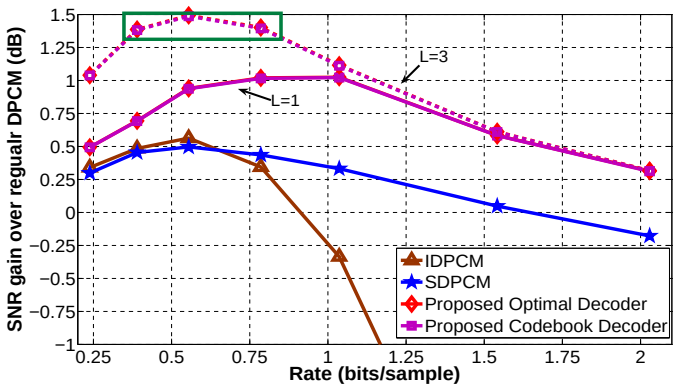


Figure 4.2. Performance comparison of different delayed decoders for a first order gaussian AR process with $\rho = 0.95$. The performance curves of the proposed optimal and codebook-based delayed decoders are almost indistinguishable

We compare the proposed optimal and codebook-based decoders, at look-ahead values $L = 1$ and 3, with IDPCM ($L = 1$ necessarily), and SDPCM at $L = 3$ (which outperforms SDPCM at $L = 1$ or 2). For reasons that will soon become obvious, we have fixed $L' = 0$, i.e., the delayed decoding codebook is

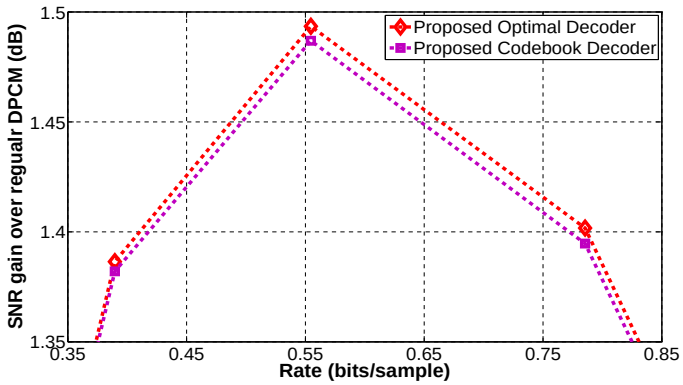


Figure 4.3. Magnification of the boxed region in Fig. 4.2, showing the performance gap between the proposed optimal delayed decoder and its codebook-based approximation.

indexed only by present and future indices, $\{i_l\}_n^{n+L}$. The quantizer $\mathcal{Q}$ is a symmetric uniform threshold quantizer (UTQ), suitably scaled to vary the bit-rate, estimated as the first order entropy of output indices. At every rate, and for every choice of innovation density $p_Z(z)$, while the thresholds of the quantizer are fixed, the reconstructions $g_{\mathcal{Q}}(\cdot)$ are optimized recursively as in [27], thereby also computing $p_E(e)$ as required for the numerical evaluation of the delayed decoding codebook (Sec. 4.3.1). The alternate, training-set based design leads to essentially the same performance. We emphasize that the proposed decoding schemes are independent of the choice of $\mathcal{Q}$, and the UTQ represents a practical case with widespread deployment in signal compression applications. Each point on the graphs (Figs. 4.2 - 4.5) has been obtained by averaging over 20 trials, with a random sequence of 2000 samples each. The figures depict SNR gains over the regular DPCM decoder, versus bit-rate.

The results demonstrate that both proposed decoders substantially outperform SDPCM, and IDPCM. The latter are not always guaranteed to perform

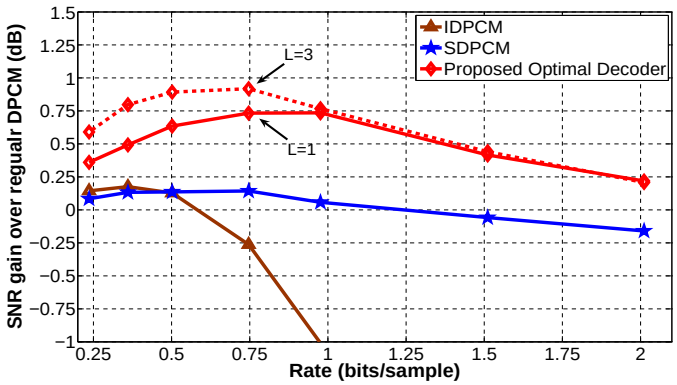


Figure 4.4. Performance comparison of different delayed decoders for a first order gaussian AR process with $\rho = 0.8$.

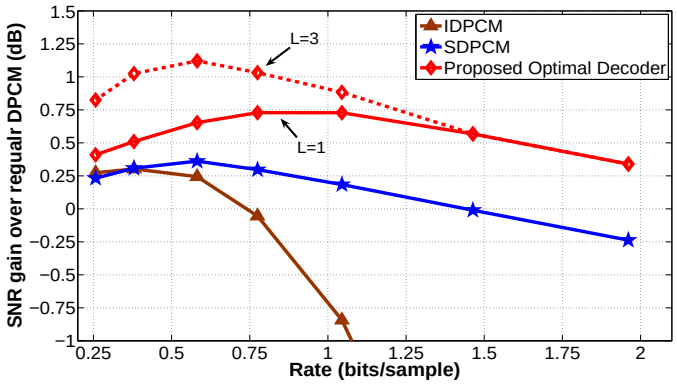


Figure 4.5. Performance comparison of different delayed decoders for a first order AR process with laplacian innovations, and $\rho = 0.95$.

better than regular DPCM. In case of the processes considered, the performance of the codebook-based decoder (with $L' = 0$) is already very close to that of the optimal approach. Fig. 4.3 magnifies the boxed region in Fig. 4.2 to show the performance gap between the two methods. In other words, almost all the past history is captured in the prediction $\tilde{x}_n$, and $c(\{i_l\}_n^{n+L})$ is an estimate of the prediction error that incorporates additional information from current and future indices. Given the observed quality of this codebook approximation, the corre-

sponding curves have been omitted in Figs. 4.4 and 4.5, to avoid clutter. Note that even with only 1 sample decoding delay the proposed approaches provide better performance than SDPCM at a higher delay ($L = 3$). At low bit-rates, as L increases, the gain over regular DPCM due to both proposed schemes increases, i.e., the information $\{i_l\}_{n+1}^{n+L}$ provides on x_n increases with decreasing rate. At high bit-rates, there is diminishing return from increasing L. The poor performance of IDPCM and SDPCM at high rate is attributed to the observation made earlier in this chapter. As the rate increases, the interval $\mathcal{I}_n$ shrinks, and it becomes more likely that the smoothed estimates, $\hat{x}_n^{sdpcm}$ and $\hat{x}_n^{idpcm}$, will fall outside it. The optimal, and codebook-based delayed decoders, by design, account for this interval information. The results for gaussian innovations, at different values of ρ, indicate expectedly that higher correlation offers more to be gained by increasing L.

4.5 Codebook Size Reduction

In the following discussion we employ the notation $\mathcal{I}_{\mathcal{Q}}(i) = \Big[a(i), b(i)\Big)$ to specify the limits of the quantizer interval. The index value $i = 0$ indicates the quantization interval containing the origin, $i > 0$ indicates cells in the positive region, and $i < 0$ cells in the negative region. We assume a symmetric mid-tread quantizer, commonly employed when the source has a symmetric zero-mean pdf (the UTQ in Sec. 4.4 is of this type). Thus, $a(0) = -b(0)$ with the associated reconstruction $g_{\mathcal{Q}}(0) = 0$. Let $\rho \geq 0$ (the case $\rho < 0$ simply leads to a dual of the results that follow).

Claim 2: Let the innovations be laplacian, i.e., $p_Z(z) = \frac{\lambda}{2}\exp(-\lambda|z|)$, the

quantizer be symmetric mid-tread, and the look-ahead $L = 1$. If $i_n = 0$ then the optimal delayed reconstruction $\hat{x}_n^*$ (and hence the codebook-based delayed reconstruction) is sensitive only to the sign of the index i_{n+1} (and not its actual value).

Note: This implies that, rather than storing K reconstructions in the codebook (corresponding to each value of i_{n+1}) when $i_n = 0$, we need only 3 (for each sign of i_{n+1}), thus reducing the storage complexity.

Proof: By (4.12),

$$p(\mathcal{I}_{n+1}|x_n) = \frac{\lambda}{2} \int_{a(i_{n+1})}^{b(i_{n+1})} \exp(-\lambda|e - \rho(x_n - \hat{x}_n)|)de. \tag{4.22}$$

Consider the case $i_{n+1} > 0$. Recall that $i_n = 0$. This implies that, in (4.22), $\rho(x_n - \hat{x}_n) < \rho b(0) < a(i_{n+1}) \leq e$, for $x_n \in \mathcal{I}_n$. Thus $\forall i_{n+1} > 0$

$$p(\mathcal{I}_{n+1}|x_n) = \alpha(i_{n+1}) \exp(\lambda\rho(x_n - \hat{x}_n)),\ x_n \in \mathcal{I}_n \tag{4.23}$$

where $\alpha(i_{n+1})$ captures all the dependence on the value of the index i_{n+1} due to the integration in (4.22). After substituting (4.23) in (4.8), we can rewrite (4.7) as

$$\hat{x}_n^* = \frac{\int\limits_{\mathcal{I}_n} x_n p(x_n|\{\mathcal{I}_l\}^n)\alpha(i_{n+1}) \exp(\lambda\rho(x_n - \hat{x}_n))dx_n}{\int\limits_{\mathcal{I}_n} p(x_n|\{\mathcal{I}_l\}^n)\alpha(i_{n+1}) \exp(\lambda\rho(x_n - \hat{x}_n))dx_n} \tag{4.24}$$

$\forall i_{n+1} > 0$. Note that $\alpha(i_{n+1})$ cancels out, which eliminates all dependence on i_{n+1}. Thus, when $i_n = 0$, $\hat{x}_n^*$ is the same for all positive i_{n+1}. The codebook reconstruction too shares this property, as it differs from the optimal estimate only in the approximation for the past, i.e., only in its approximation for $p(x_n|\{\mathcal{I}_l\}^n)$ in (4.24). Following similar arguments, given $i_n = 0$ we can show that if $i_{n+1} < 0$ then $p(\mathcal{I}_{n+1}|x_n)$ is of the form

$$p(\mathcal{I}_{n+1}|x_n) = \beta(i_{n+1}) \exp(-\lambda\rho(x_n - \hat{x}_n)),\ x_n \in \mathcal{I}_n. \tag{4.25}$$

Here $\beta(i_{n+1})$ captures all dependence on i_{n+1}, similarly as $\alpha(i_{n+1})$. Again by substitution in (4.8), we can show that $\hat{x}_n^*$ of (4.7) is independent of the actual value of the index i_{n+1}. In summary, when $i_n = 0$, the one sample delayed reconstruction is dependent only on the sign of the index i_{n+1}. As an aside, note that a non-causal filter that *linearly combines* reconstructions (i.e., neglecting index information) can never provide this optimal reconstruction that is *conditionally unaltered* by future index values.

Claim 2 can be suitably extended to the case $i_n \neq 0$, and further to $L > 1$, although (for an exact representation of the optimal or codebook estimates) such extensions may require more involved conditions than the sign of future indices. Nevertheless, as we shall see in the results of this section, there is minimal loss in performance if each future index in $\{i_l\}_{n+1}^{n+L}$ is simply coarsely quantized to the cases > 0, < 0, and $= 0$. Without formal arguments, we extend the same logic to the past indices $\{i_l\}_{n-L'}^{n-1}$, although this has no bearing on the experiments in Sec. 4.4 with first order sources, where $L' = 0$ was nearly optimal. It does find use in case of higher order sources (Sec. 4.6). Ordinarily the codebook $c(\{i_l\}_{n-L'}^{n+L})$ has $K^{L'+L+1}$ reconstructions which could still be quite large if the number of quantizer cells K is big, even though L and L' may be modest. With the proposed mapping of indices to just their signs, the delayed decoding codebook size is now reduced considerably, to $3^{L'+L}K$.

The arguments and derivations of index-mapping so far assumed laplacian innovations. Nevertheless we propose employing this technique for other distributions as well. We specifically consider the performance of this technique on a first order gaussian source (with $\rho = 0.95$). The deliberate mismatch (from the assumed laplacian) results in a performance loss due to codebook size reduction.

This source is covered by Fig. 4.2 in Sec. 4.4. For the higher four rates marked on the figure, Table. 4.1 compares the original codebook size, and the reduced size after index-mapping, at look-ahead values $L = 1$ and 3. Also provided is the loss in performance due to the reduced size codebook which, as evidenced, is negligible despite the mismatch. For laplacian innovations this loss is even smaller.

Note that we assume here that K is finite, although the UTQ is theoretically an infinite cell quantizer, and covers the entire real line with cells of equal width. In experiments we use only a finite number of cells including overload cells that extend to infinity at each end side. Given the innovation pdf we ensure that the probability of overload, i.e., that an index lying outside the quantizer range, is negligible. The quantizer range is rate-independent, while the step-size, and hence the value of K are rate-dependent. The three lower rates in Fig. 4.2 have $K = 3$, in which case there is no codebook size reduction, and are therefore excluded from Table. 4.1. It is evident from Fig. 4.2 that at 1.5 or 2 bits/sample most of the delayed decoding gains are obtained with 1 sample delay, in which case a larger codebook (with $L > 1$) provides no tangible advantage.

4.6 Generalization to Higher Order Sources

The first order AR source (4.1), is generalized to order M as

$$x_n = \sum_{i=1}^{M} a_i x_{n-i} + z_n \ . \tag{4.26}$$

In (4.1) we employed the notation ρ in place of a_1 to explicitly indicate that $a_1 = \rho$ was in fact the correlation coefficient of the first order AR process. However, in

Rate (bits/sample)	0.79	1.03	1.52	2.02
K	5	5	7	11
$L = 1$, $L' = 0$				
Codebook size (original)	25	25	49	121
Codebook size (reduced)	15	15	21	33
Performance loss (dB)	0.0000	0.0008	0.0215	0.0301
$L = 3$, $L' = 0$				
Codebook size (original)	625	625	2401	14641
Codebook size (reduced)	135	135	189	297
Performance loss (dB)	0.0000	0.0002	0.0192	0.0303

Table 4.1. Comparison of codebook sizes and performance loss for gaussian AR source with $\rho = 0.95$, when the index-mapping technique of Sec. 4.5 is applied. Although derived under the assumption of laplacian innovations this technique works well for the (mismatched) gaussian case too.

the general Mth order case the inter-sample correlations are dependent on, but not the same, as the coefficients a_i in (4.26). DPCM is implemented similarly to the first order case, except that the matched predictor of order M:

$$\tilde{x}_n = \sum_{i=1}^{M} a_i \hat{x}_{n-i} \tag{4.27}$$

This Mth order process of random scalars in $\mathbb{R}$, can be equivalently viewed as a first order process of random vectors in $\mathbb{R}^M$, by a formulation similar to the Kalman filtering plant model of (4.4), albeit with a different Ψ. The current structure of the recursions in Sec. 4.2 for the optimal delayed decoder still hold, but the intervals $\mathcal{I}_n$ are replaced by corresponding M-dimensional segments $\underline{\mathcal{I}}_n$, and integrals, auxiliary functions in the algorithms, and densities are all defined in the vector space $\mathbb{R}^M$. Needless to say, the increased dimensionality renders the optimal delayed decoder cumbersome, in particular, due to the M-dimensional convolutions now implicit in (4.9) and (4.11). But we note that the derivation in Sec. 4.3 of the codebook-based delayed decoder still stands, with the matched

predictor now defined as (4.27), and it retains its low computational complexity. The reconstructions are still given by (4.17), and asymptotically this is optimal.

We present here the application of the codebook-based decoder to second and third order sources. Again, the DPCM encoder employed a matched predictor (here, (4.27)). Similar to the first order case (Sec. 4.4) the UTQ is used, with scaling to achieve the required bit-rate. But the optimization of the reconstructions $g_{\mathcal{Q}}(\cdot)$ by the iterative method in [27], that alternates between optimizing the quantizer given the prediction error density, and estimating the prediction error density given the quantizer, is highly complex for process orders > 1. Hence we employ a different predictive quantizer design strategy: training-set based closed loop optimization [22]. Given a training-set of the source $\{x_n\}_0^N$, and an initial quantizer $\mathcal{Q}_0$, the DPCM encoder is run to generate the prediction errors $\{e_n\}_0^N$. Given this prediction error training sequence a new reconstruction for each quantizer cell is calculated as the mean of the prediction errors that lie within the cell (the UTQ partitions are fixed and need not be optimized). This results in a new quantizer $\mathcal{Q}_1$, that replaces $\mathcal{Q}_0$ in the encoder. These two steps are alternated till the the quantizer designs of consecutive iterations converge. In certain predictive coding scenarios such a closed loop optimization can fail to converge, and alternatives have been proposed [52]. But with respect to the simple sources and fixed quantizer partitions considered here such problems were not encountered. With the encoder fixed, the codebook for the delayed decoder is designed by the training-set based method of Sec. 4.3.2, which now offers a distinct advantage over the alternate, formula-based approach in Sec. 4.3.1 due to the higher source order. Extension of IDPCM and SDPCM to Mth-order sources can be found in the respective references.

Fig. 4.6 and Fig. 4.7 compare the performance of the competing schemes on a second order source with $(a_1, a_2) = (1.5, -0.6)$, and laplacian innovations, and a third order source with $(a_1, a_2, a_3) = (1.526, -0.773, 0.101)$, and gaussian innovations, respectively. The coefficients of the third order source correspond to Law's 3-tap filter [20], [77]. The first two autocorrelation coefficients (i.e., with lag 1 and 2) for the second order source are 0.9375 and 0.8063, and in the third order case the first three coefficients are 0.9001, 0.6914, and 0.4604, respectively. In both cases, the parameters of the innovation pdf are adjusted to maintain unit source variance. In the second order case, all competing methods used a 1-sample look-ahead, and in the third order case, $L = 2$. Unlike in Sec. 4.4, the delayed decoding codebook is now indexed by past indices too. In case of both sources it is observed that limiting the window to just one past index, i.e., $L' = 1$, results in the codebook-based delayed decoder performing slightly worse than SDPCM at the lowest bit-rate considered ($\sim$ 0.25 bits/sample). As the window is increased to encompass more past indices ($L' = 3$), this sub-optimality is overcome. It should be emphasized that the decoding delay of all competing methods is still the same, and the performance curve of the *optimal* delayed decoder with this value of delay is just the envelop of the curves for the codebook decoder with increasing L'. Note that with $L' = 3$, the codebook-based decoder provides substantial gains over both SDPCM and IDPCM, and this is a lower limit on the optimal performance. Increasing L' to 4 increased the gains by about 0.05 dB at the lower bit-rates. Since our experiments with a full-size delayed decoding codebook yielded very small performance improvements over the reduced-size codebook motivated in Sec. 4.5, the results presented in Fig. 4.6 and Fig. 4.7 were actually obtained with the latter, i.e., each past and future

index is mapped to one of the cases, > 0, < 0, and $= 0$, before the codebook is looked up.

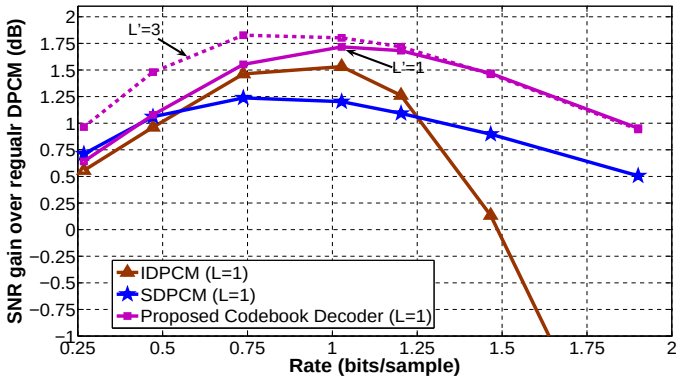


Figure 4.6. Performance comparison of different delayed decoders for a 2nd order AR process with laplacian innovations.

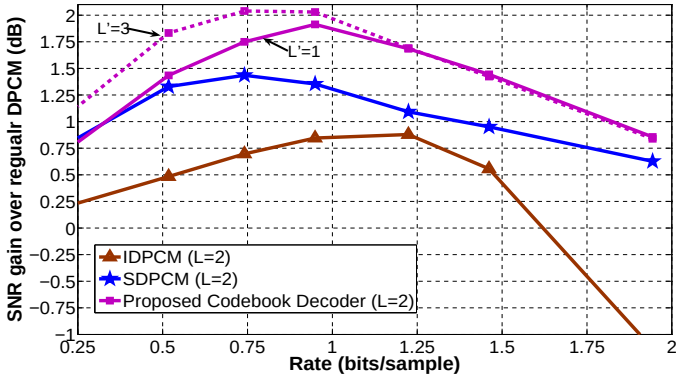


Figure 4.7. Performance comparison of different delayed decoders for a 3rd order gaussian AR process.

The size of the delayed decoding codebook, along with the number of quantizer cells K, at different bit-rates for the second order process with laplacian innovations is enumerated in Table. 4.2. The length of the training-set employed in the codebook design was 50000 for both cases $L' = 1$ and $L' = 3$, although the latter necessitates a larger codebook. As noted in Sec. 4.3.2, some combinations

of the indices $\{i_l\}_{n-L'}^{n+L}$ are extremely infrequent, and due to their unavailability in the limited training set the delayed decoding reconstruction in these cases defaults to the zero-delay reconstruction $g_{\mathcal{Q}}(i_n)$. The 'effective size of the codebook' in Table. 4.2 provides the number of reconstructions of the complementary type (i.e., those that are somewhat frequent), and these are the ones that really provide the observed delayed decoding gains. Note that in most cases the effective size is just a small fraction of the actual size (which in turn has already been reduced compared to the full codebook due to index mapping). In other words, most of the entries in this delayed decoding codebook correspond to events (i.e., index windows $\{i_l\}_{n-L'}^{n+L}$) that rarely occur in the index sequence. This suggests the potential for further pruning of the codebook. For instance, the decoder during operation could first check by some simple rule whether the index window is one that appears often in the index sequence, or is a rare event. Only in the former case delayed decoding is employed, which now requires a small codebook with entries corresponding to only the few frequent events. It must also be noted that since the quantizer and the process distributions employed in these examples are symmetric, there is an in-built symmetry in the delayed decoding codebook too, which can by itself reduce all the codebook sizes seen so far by at least a factor of two.

Rate (bits/sample)	0.26	0.47	0.74	1.03	1.20	1.46	1.90
K	5	7	9	15	17	25	39
$L = 1$, $L' = 1$							
Codebook size (reduced by index mapping)	45	63	81	135	153	225	351
Effective size of the codebook	12	14	24	30	38	50	74
$L = 1$, $L' = 3$							
Codebook size (reduced by index mapping)	405	567	729	1215	1377	2025	3159
Effective size of the codebook	32	46	102	180	230	316	458

Table 4.2. Comparison of the codebook size at different bit-rates for the second order laplacian source, and the effective size that provides delayed decoding gains

4.7 Encoder Modification to Incorporate Delayed Decoding

A DPCM encoder that incorporates a linear predictor such as (4.2) or (4.27) embeds within it a local decoder to obtain the reconstruction of prior samples. A natural question is if the improved estimate obtainable by delayed decoding could also be generated by the encoder, and employed for enhanced prediction. In effect, the delayed decoding gain would be fed back into the prediction loop, thus amplifying it and improving DPCM performance further. But the restriction is that the encoder should be zero-delay. Consider incorporating a look-ahead $L = 1$ into the encoder's local decoder. The delayed reconstruction of x_n can be obtained only when i_{n+1} is known. Thus x_{n+1} itself cannot be predicted from the delayed reconstruction of x_n, but if the predictor is second order, then the prediction for x_{n+2} can incorporate in it the 1-sample delayed reconstruction of x_n. In general, an encoder with an Mth-order linear predictor can incorporate an L-sample delayed decoder to reconstruct $M - L$ of the M samples it linearly

combines in the prediction. The specific case of an encoder with a second order predictor (matched to the second order source with laplacian innovations in Sec. 4.6) is discussed here as an example. Naturally, $L = 1$ is the only possibility in this case. The general principle is as follows. The delayed reconstructions are fed back into the prediction loop, via the prediction:

$$\tilde{x}'_n = a_1 \hat{x}_{n-1} + a_2 \hat{x}^d_{n-2} \tag{4.28}$$

where we use the notation $\tilde{x}'_n$ to distinguish from the prediction $\tilde{x}_n$ employed in regular DPCM. The zero-delay reconstruction $\hat{x}_n$ is now given by

$$\hat{x}_n = \tilde{x}'_n + g_{\mathcal{Q}}(i_n) \ . \tag{4.29}$$

With a 1-sample delay, the decoder reconstructs x_n as $\hat{x}^d_n$. This delayed reconstruction is also produced at the encoder's local decoder, and employed for prediction in (4.28). As alluded to in Sec. 4.1.1, this type of encoder modification has already been proposed in [77], where the IDPCM principle was employed to obtain $\hat{x}^d_n$. Specifically, in the example here this would be,

$$\hat{x}^d_n = b_1 \hat{x}_{n+1} + b_{-1} \hat{x}^d_{n-1} + b_{-2} \hat{x}^d_{n-2} \ . \tag{4.30}$$

The non-causal IIR smoothing filter (i.e., the coefficients b_l above) are obtained using the same procedure described in Sec. 4.1.1. Since the process is second order, $b_l = 0$ for $l < -2$. The DPCM scheme with a modified encoder that employs (4.30) as the definition of $\hat{x}^d_n$ will be henceforth referred to as IDPCM-Enc.

Alternatively, we could obtain $\hat{x}^d_n$ via the optimal delayed decoder proposed in this chapter, i.e., setting $\hat{x}^d_n = E[x_n | \{\mathcal{I}_l\}^{n+1}]$. Since this is computationally cumbersome we approximate:

$$\hat{x}^d_n = \tilde{x}'_n + c^*(\{i_l\}^{n+1}_{n-L'}) \tag{4.31}$$

The need for the superscript $*$ in $c^*(\cdot)$ will become obvious shortly. In what follows, we use the abbreviation CDD-Enc to refer to the DPCM scheme with a modified encoder which embeds in it the codebook-based delayed decoder (4.31). The regular codebook-based delayed decoder, with the DPCM encoder unaltered, as discussed in prior sections, will be referred to as CDD. For both CDD and CDD-Enc, $L' = 3$. Since this type of prediction loop modification has not been attempted in the case of SDPCM [20], we exclude it from the discussion in this section. The objective here is to demonstrate that incorporating the proposed delayed-decoding approaches suitably into the prediction loop can result in performance improvements similar to what IDPCM-Enc achieves compared to regular IDPCM.

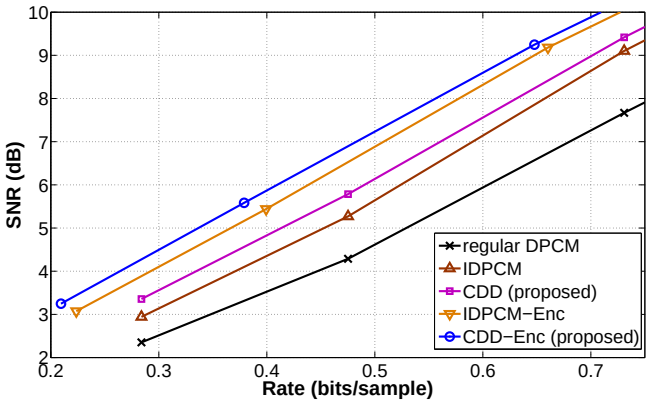
The performance of regular DPCM, IDPCM, CDD, IDPCM-Enc, and CDD-Enc for the second order laplacian source of Sec. 4.6 is compared in Fig. 4.8, in terms of SNR vs bit-rate. Unlike in Sec. 4.4 and Sec. 4.6, the encoders are not the same for all schemes, and hence the same entropy output cannot be ensured. Strictly speaking, for both IDPCM-Enc and CDD-Enc, the modified predictor (4.28) results in a new prediction error density, and hence the quantizer reconstructions $g_{\mathcal{Q}}(\cdot)$ need to be obtained anew for both types of encoders separately. But IDPCM-Enc as implemented in [77] employed no such optimization, and used the same quantizer as the regular DPCM encoder. In our case, this is just the quantizer designed in Sec. 4.6 for the specific source considered here. In order to ensure a fair comparison this quantizer has been used at the encoder in all the schemes, including CDD-Enc. Note that the design of the delayed decoding codebook as well depends on the prediction error statistics. In the case of CDD, the encoder itself was unaltered, and the prediction error statistics were thus solely

determined by the choice of quantizer and predictor coefficients. Thus, given the encoder and a training set of the source, the design procedure in Sec. 4.3.2 could be followed to obtain the appropriate delayed decoding codebook, $c(\{i_l\}_0^3)$. But in the case of CDD-Enc the prediction error statistics are themselves dependent on the delayed decoding codebook, as the delayed reconstructions are fed back into the prediction loop. Therefore, although the quantizer itself is unaltered, we follow a closed loop optimization procedure to obtain the delayed decoding codebook for CDD-Enc, i.e., the training-set based codebook design procedure of Sec. 4.3.2 is slightly modified. We start off with an initial guess for the codebook: the same as $c(\{i_l\}_0^3)$ employed in CDD. Now the modified DPCM encoder, with this delayed decoding codebook embedded in it, is run to obtain a new set of prediction errors, and the codebook recalculated. We repeat this procedure till the codebooks of consecutive iterations converge. Since the eventually obtained codebook is different from $c(\{i_l\}_0^3)$, employed in CDD, we referred to it as $c^*(\{i_l\}_0^3)$ in (4.31). The performance gap between CDD-Enc and CDD, is almost the same as between IDPCM-Enc and IDPCM: about 1-dB at rates 0.25-0.75 bits/sample. Note that IDPCM-Enc retains the poor performance of IDPCM at high-rates. CDD-Enc performs better than IDPCM-Enc by about 0.5dB at rates of about 0.25-0.5 bits/sample, and is substantially better at rates close to 1.25 bits/sample.

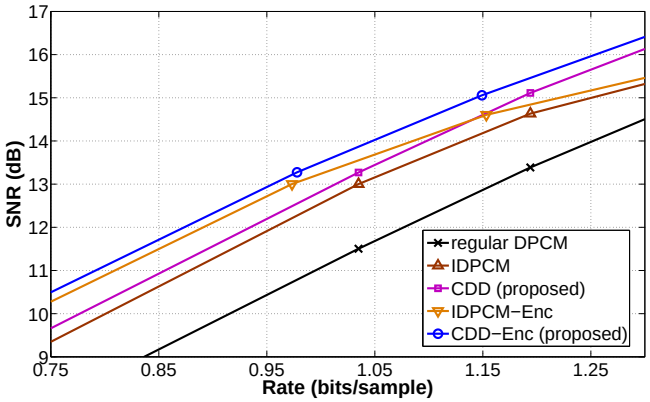
4.8 Conclusion

An optimal delayed decoding algorithm for predictively encoded autoregressive sources is proposed, that obtains the optimal reconstruction of a sample via

an estimation-theoretic approach that recursively calculates the sample's pdf conditioned on all information known to the decoder for a given delay or look-ahead. Irrespective of the bit-rate, or innovation probability density, the algorithm ensures optimal reconstruction. The optimal delayed decoder in turn motivates an approximate codebook-based approach, which is indeed asymptotically (as the codebook-size increases) optimal, and has performance almost indistinguishable from the former even at very modest dimensions. The codebook approach has the obvious advantage of low implementation complexity, and is amenable as well for a training-set based design. Thus, it is particularly useful for delayed decoding of higher order sources, where the recursion-based optimal decoder is rendered cumbersome. Insights into the optimal delayed decoder also motivate a methodology for codebook size reduction, based on an index mapping technique. Simulations with first, second, and third order sources demonstrate the considerable performance gains of both the optimal and codebook-based approaches, over existing smoothing post-filters. Finally, an example encoder modification that incorporates delayed decoding at the local decoder is demonstrated, which further amplifies the delayed decoding gains via feedback in the prediction loop.



(a) Low bit-rate region



(b) Medium bit-rate region

Figure 4.8. Performance comparison of delayed decoding schemes with and without encoder modifications: feedback of delayed decoding gains in the prediction loop considerably improves low bit-rate performance. Source is 2nd order with laplacian innovations.

Chapter 5

Conclusion and Future Directions

This dissertation has primarily focused on optimal decision making at the encoder or decoder in signal compression, via incorporation of delay.

In the case of encoding delay, trellis-based dynamic programming approaches were developed that employed delay for optimal encoding parameter selection in audio coding. Such approaches are of particular benefit to applications that employ off-line compression, to which category many audio coding applications do belong. Significant gains compared to standard methods, that employ a myopic encoding process, were provided by the proposed algoritm. Motivated by the rate-distortion optimization approach followed in this delayed encoding paradigm, we explored certain deficiencies in the audio distortion metric itself, and suggested modifications that substantially improved the subjective quality of the coded audio.

In the context of decoding delay, we proposed the optimal algorithm for decoding predictively encoded sources, when a finite delay is admissible at the decoder.

This estimation-theoretic algorithm recursively calculates the probability density function of each sample, conditioned on all information available at the decoder, and obtains the optimal reconstruction via conditional expectation. Experiments indicated substantial gains over prior work that adopted a smoothing approach to the problem. The optimal delayed decoder in turn motivated a codebook-based decoder that is nearly optimal even with modest codebook size (or memory). The insight into the optimal delayed decoder also motivated approaches for reduction of the size of this codebook without significant performance loss. This delayed decoding approach finds utility in conventional prediction-based applications, such as motion compensated video codecs, as well as in emerging low-delay applications where prediction is expressly preferred over competing transform-based techniques, due to the unacceptable complexity or framing delays of the latter.

5.1 Future Directions

- **Unified speech and audio coding**: An examination of the subjective tests conducted to evaluate the distortion metric modification of Sec. 3.2 indicates substantial improvements in the quality of speech coded by the audio coder. This suggests that that the distortion metric may be the key to obtain a unified audio and speech coder, an area currently of substantial interest in the industry. We conjecture that this MDST-based modification to the distortion metric, that is cognizant of the noise envelop rather than just its projection on the MDCT basis, parallels the approach followed in speech codecs that tries to preserve the spectral envelop in the

LPC coefficients. More concrete work that analyzes the effect of the proposed distortion metric on speech samples is one possible extension to be considered.

- **Encoder optimization for delayed decoding**: In Sec. 4.7 we considered an example encoder modification for improved prediction, based on incorporating delayed decoding locally at the DPCM encoder. But the quantizer at the encoder was not optimized for the true prediction error density, that results from the refined reconstructions obtained by delayed decoding. A possible future direction is to optimize the quantizer, and the delayed decoding codebook jointly, which should considerably improve over the performance demonstrated in Sec. 4.7. A closed loop approach for this optimization might run into instability issues, and we might need to devise an appropriate pseudo-closed loop optimization similar to the one in [52].

- **Adaptive delayed decoding**: While we assumed a time-invariant DPCM encoder throughout Chapter 4, future work could consider the application of the optimal delayed decoder in adaptive predictive coding schemes (such as ADPCM). Note that the optimal delayed decoder (Sec. 4.2.1) assumes no particular form of the predictor or the quantizer. It simply requires the appropriate interval information, and the source model (innovation density) at every instant n, and is thus amenable to adaptation over time of both source statistics as well as the encoder. Thus we can consider application of the delayed decoder to a practical ADPCM-based speech coding setting. An interesting, albeit complicated, research direction would be to arrive at an appropriate adaptation strategy to modify the codewords of the codebook-based delayed decoder in tandem with the existing adaption technique for

the predictor or quantizer in the ADPCM codec. While the discussion of experiments and results in Chapter 4 was limited to generic scalar sources, preliminary results of employing the proposed ET algorithm for delayed video decoding in the H.264 framework have been demonstrated in [39].

Bibliography

[1] *Specification of the Bluetooth System 2.0+ EDR, Core System Package, Part B: Baseband Specification, Section 9: Audio*, 2004.

[2] A. Aggarwal, S. L. Regunathan, and K. Rose. Trellis-based optimization of MPEG-4 Advanced Audio Coding. In *Proc. IEEE Workshop. Speech Coding*, pages 142–144, Sep 2000.

[3] A. Aggarwal, S. L. Regunathan, and K. Rose. A trellis-based optimal parameter values selection for audio coding. *IEEE Trans. Audio, Speech and Language Processing*, 14(2):623–633, Mar 2006.

[4] K. Akagiri, M. Katakura, H. Yamauchi, E. Saito, M. Kohut, M. Nishiguchi, and K. Tsutsui. *Sony systems*, pages 43(1)–43(16). Digital Signal Processing Handbook. V. Madisetti and D. B. Williams Eds. New York: IEEE Press, 1998.

[5] J. B. Anderson and J. B. Bodie. Tree encoding of speech. *IEEE Trans. Information Theory*, IT-21(4):379–387, Jul 1975.

[6] D. S. Arnstein. Quantization error in predictive coders. *IEEE Trans. Communication*, 23(4):423–429, Apr 1975.

[7] C. Bauer. The optimal choice of encoding parameters for MPEG-4 AAC streamed over wireless networks. In *Proc. ACM Workshop. Wireless Multimedia Networking and Performance Modeling*, pages 93–100, Oct 2005.

[8] C. Bauer and M. Vinton. Joint optimization of scale factors and huffman codebooks for MPEG-4 AAC. In *Proc. IEEE Workshop. Multimedia Signal Processing*, pages 111–114, Sep 2004.

[9] J. G. Beerends and J. A. Stemerdink. A perceptual audio quality measure based on a psychoacoustic sound representation. *J. Audio Eng. Soc.*, 40(12):963–978, Dec 1992.

[10] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti. Statistical analysis of the 2D-DCT coefficients of the differential signal for images. *Signal Processing: Image Communication*, 4(6):477–488, May 1992.

[11] R. E. Bellman. *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[12] P. A. Bello, R. N. Lincoln, and H. Gish. Statistical delta modulation. *Proc. IEEE*, 55(3):308–319, Mar 1967.

[13] J. Boehm, S. Kordon, and P. Jax. An experimental audio coder using rate-distortion controlled temporal block switching. In *Proc. 120th AES convention*, May 2006.

[14] K. Brandenburg. Evaluation of quality for audio coding at low bit-rates. In *Proc. 82nd AES Convention*, Mar 1987.

[15] K. Brandenburg. OCF - a new coding algorithm for high quality sound signals. In *Proc. IEEE ICASSP*, volume 12, pages 141–144, May 1987.

[16] K. Brandenburg, J. Herre, J. D. Johnston, Y. Mahieux, and F. E. Schroeder. ASPEC: adaptive spectral entropy coding of high quality music signals. In *Proc. 90th AES Convention*, Feb 1991.

[17] K. Brandenburg and J. Johnston. Second generation perceptual audio coding: the hybrid coder. In *Proc. 88th AES Convention*, Mar 1990.

[18] K. Brandenburg and T. Sporer. NMR and masking flag: Evaluation of quality using perceptual criteria. In *Proc. AES 11th International Conference*, May 1992.

[19] E. Camberlein and P. Philippe. Optimal bit-reservoir control for audio coding. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 251–254, Oct 2005.

[20] W-W. Chang and J. D. Gibson. Smoothed DPCM codes. *IEEE Trans. Communication*, 39(9):1351–1359, Sep 1991.

[21] C. Cheng. Method for estimating magnitude and phase in the MDCT domain. In *Proc. 116th AES convention*, May 2004.

[22] V. Cuperman and A. Gersho. Vector predictive coding of speech at 16kbits/s. *IEEE Trans. Communication*, 33(7):696–685, Jul 1985.

[23] C. R. Davis and M. E. Hellman. On tree coding with a fidelity criterion. *IEEE Trans. Information Theory*, IT-21(4):373–378, Jul 1975.

[24] R. Der, P. Kabal, and W-Y. Chan. Rate-distortion allocation for time-frequency dependent audio coding. In *Proc. IEEE ICASSP*, pages 197–200, Mar 2005.

[25] J. G. Dunham. Optimal discrete-time delta modulation scheme. *IEEE Trans. Communication*, 34(5):510–512, May 1986.

[26] B. Edler. Codierung von audiosignalen mit uberlappender transformation und adaptiven fensterfunktionen. *Frequenz*, 43(9):252–256, Sep 1989.

[27] N. Farvardin and J. W. Modestino. Rate-distortion performance of DPCM schemes for autoregressive sources. *IEEE Trans. Information Theory*, 31(3):402–418, May 1985.

[28] L. D. Fielder, M. Bosi, G. Davidson, M. Davis, C. Todd, and S. Vernon. *AC-2 and AC-3: low-complexity transform-based audio coding*, pages 54–72. Collected Papers on Digital Audio Bit-Rate Reduction. N. Gilchrist and C. Gerwin, Eds. New York: Audio Eng. Soc., 1996.

[29] T. Fine. Properties of an optimum digital system and applications. *IEEE Trans. Information Theory*, 10(4):287–296, Oct 1964.

[30] T. R. Fischer and M. Wang. Entropy-constrained trellis-coded quantization. *IEEE Trans. Information Theory*, 38(2):415–426, Mar 1992.

[31] G. D. Forney, Jr. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, Mar 1973.

[32] J. D. Gibson and T. R. Fischer. Alphabet-constrained data compression. *IEEE Trans. Information Theory*, 28(3):443–457, May 1982.

[33] E. A. Gifford, B. R. Hunt, and M. W. Marcellin. Image-coding using wavelet transforms and entropy-constrained trellis-coded quantization. *IEEE Trans. Image Processing*, 4(8):1061–1069, Aug 1995.

[34] A. Goris and J. D. Gibson. Incremental tree coding of speech. *IEEE Trans. Information Theory*, IT-27(4):511–516, Jul 1981.

[35] R. M. Gray. Sliding-block source coding. *IEEE Trans. Information Theory*, IT-21(4):357–367, Jul 1975.

[36] R. M. Gray. Time-invariant trellis encoding of ergodic discrete time sources with a fidelity criterion. *IEEE Trans. Information Theory*, IT-23(1):71–83, Jan 1977.

[37] O. G. Guleryuz and M. T. Orchard. Rate-distortion based temporal filtering for video compression. In *Proc. IEEE DCC*, pages 122–131, Mar 1996.

[38] O. G. Guleryuz and M. T. Orchard. On the DPCM compression of gaussian autoregressive sources. *IEEE Trans. Information Theory*, 47(3):945–956, Mar 2001.

[39] J. Han, V. Melkote, and K. Rose. Estimation-theoretic delayed decoding of predictively encoded video sequences. In *Proc. IEEE DCC*, Mar 2010.

[40] A. Hayashi. Differential pulse code modulation of stationary gaussian inputs. *IEEE Trans. Communication*, 26(8):1137–1147, Aug 1978.

[41] P. Ishwar and K. Ramachandran. On decoder-latency versus performance tradeoffs in differential predictive coding. In *Proc. IEEE ICIP*, pages 1097–1100, Oct 2004.

[42] ISO/IEC std. ISO/IEC JTC1/SC29 11172-3:1993. *Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*, 1993.

[43] ISO/IEC std. ISO/IEC JTC1/SC29 13818-7:1997. *Information Technology - Generic Coding of Moving Pictures and Associated Audio*, 1997.

[44] ISO/IEC std. ISO/IEC JTC1/SC29 14496-3:2005. *Information Technology - Generic Coding of Moving Pictures and Associated Audio*, 2005.

[45] ITU-R Recommendation, BS 1387-1. *Method for objective measurements of perceived audio quality*, 2001.

[46] ITU-R Recommendation, BS 1534-1. *Method for the subjective assessment of intermediate quality level of coding systems*, 2001.

[47] ITU-T recommendation G.726. *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*, 1990.

[48] F. Jelenik and J. B. Anderson. Instrumentable tree-encoding of information sources. *IEEE Trans. Information Theory*, IT-17(1):118–119, Jan 1971.

[49] J. D. Johnston. Estimation of perceptual entropy using noise masking criterion. In *Proc. IEEE ICASSP*, volume 5, pages 2524–2527, Apr 1984.

[50] R. Kapust. A human ear related objective measurement technique yields audible error and error margin. In *Proc. AES 11th International Conference*, May 1992.

[51] M. Karjaleinen. A new auditory model for the evaluation of sound quality of audio systems. In *Proc. IEEE ICASSP*, volume 10, pages 608–611, Apr 1985.

[52] H. Khalil, K. Rose, and S. L. Regunathan. The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding. *IEEE Trans. Image Processing*, 10(1):15–23, Jan 2001.

[53] W. D. Leon-Salas, S. Balkir, N. Schemm, M. W. Hoffman, and K. Sayood. Predictive coding on-sensor compression. In *Proc. IEEE Int. Sym. Circuits and Systems*, pages 1636–1639, May 2008.

[54] N. Ma and P. Ishwar. The value of frame-delays in sequential coding of correlated sources. In *Proc. IEEE ISIT*, pages 1496–1500, Jun 2007.

[55] H. Malvar. Lapped transforms for efficient transform/subband coding. *IEEE Trans. Acoustics, Speech, Signal Processing*, 38(6):969–978, Jun 1990.

[56] H. S. Malvar and D. H. Staelin. The LOT: transform coding without blocking effects. *IEEE Trans. Acoustics, Speech, Signal Processing*, 37(4):553–559, Apr 1989.

[57] M. W. Marcellin and T. R. Fischer. Trellis-coded quantization of memoryless and Guass-Markov sources. *IEEE Trans. Communications*, 38(1):82–93, Jan 1990.

[58] J. W. Mark. Adaptive trellis encoding of discre te-time sources with a distortion measure. *IEEE Trans. Communications*, COM-25(4):408–417, Apr 1977.

[59] V. Melkote and K. Rose. Optimal delayed decoding in predictive coding systems. *submitted to the IEEE Trans. Signal Processing.*

[60] V. Melkote and K. Rose. An improved distortion measure for audio coding and a corresponding two-layered trellis approach for its optimization. In *Proc. 125th AES convention*, Oct 2007.

[61] V. Melkote and K. Rose. Trellis based approach for joint optimization of window switching decisions and bit resource allocation. In *Proc. 123rd AES convention*, Oct 2007.

[62] V. Melkote and K. Rose. A two-layered trellis approach to audio encoding. In *Proc. IEEE ICASSP*, pages 201–204, Apr 2008.

[63] V. Melkote and K. Rose. A modified distortion metric for audio coding. In *Proc. IEEE ICASSP*, pages 17–20, Apr 2009.

[64] V. Melkote and K. Rose. Optimal delayed decoding of predictively encoded sources. In *Proc. IEEE ICASSP*, Mar 2010.

[65] V. Melkote and K. Rose. Trellis based approaches to rate-distortion optimized audio encoding. *IEEE Trans. Audio, Speech, Lang. Proc.*, 18(2):330–341, Feb 2010.

[66] J.M. Mendel. *Lessons in estimation theory for signal processing, communications, and control*. Prentice Hall PTR, Upper Saddle River, NJ, 1995.

[67] H. Najafzadeh-Alaghandi and P. Kabal. Improving perceptual encoding of narrow-band audio signals at low rates. In *Proc. IEEE ICASSP*, volume 2, pages 913–916, Mar 1999.

[68] H. Najafzadeh-Alaghandi and P. Kabal. Perceptual bit allocation for low-rate coding of narrow-band audio. In *Proc. IEEE ICASSP*, volume 2, pages 893–896, Jun 2000.

[69] O. A. Niamut and R. Heudsens. R-D optimal time segmentations for the time varying MDCT. In *Proc. European Signal Processing Conf.*, pages 1649–1652, Sep 2004.

[70] O. A. Niamut and R. Heudsens. Optimal time segmentation for overlap-add systems with variable amount of window overlap. *IEEE Signal Processing Letters*, 12(10):665–668, Oct 2005.

[71] B. Paillard, P. Mabilleau, S. Morissette, and J. Soumagne. PERCEVAL: Perceptual evaluation of the quality of audio signals. *J. Audio Eng. Soc.*, 40(1):21–31, Feb 1992.

[72] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proc. IEEE*, 88(4):451–513, Apr 2000.

[73] J. P. Princen, A. W Johnson, and A. B. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Proc. IEEE ICASSP*, volume 12, pages 2161–2164, Apr 1987.

[74] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders. *IEEE. Trans. Image Processing*, 3(5):533–545, Sep 1994.

[75] K. Rose and S. L. Regunathan. Toward optimality in scalable predictive coding. *IEEE Trans. Image Processing*, 10(7):965–976, Jul 2001.

[76] G. Schuller and A. Harma. Low delay audio compression using predictive coding. In *Proc. IEEE ICASSP*, volume 2, pages 1853–1856, May 2002.

[77] M. L. Sethia and J. B. Anderson. Interpolative DPCM. *IEEE Trans. Communication*, 32(6):729–736, Jun 1984.

[78] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. In *IRE National Convention Rec.*, volume 4, pages 142–163, Mar 1959.

[79] S. Shlien. The modulated lapped transform, its time-varying forms and its applications to audio coding standards. *IEEE Trans. Speech and Audio Processing*, 5(4):359–366, Jul 1997.

[80] Y. Shoham and A. Gersho. Efficient bit-allocation for an arbitrary set of quantizers. *IEEE Trans. Acoustics, Speech, Signal Processing*, 36(9):1445–1453, Sep 1988.

[81] D. Sinha, J. D. Johnston, S. Dorward, and S. Quackenbush. *The perceptual audio coder (PAC)*, pages 42(1)–42(17). Digital Signal Processing Handbook. V. Madisetti and D. B. Williams, Eds. New York: IEEE Press, 1998.

[82] P. Sriram and M. W. Marcellin. Image-coding using wavelet transforms and entropy-constrained trellis-coded quantization. *IEEE Trans. Image Processing*, 4(6):725–733, Jun 1995.

[83] W. C. Treurniet and G. A. Soulodre. Evaluation of the ITU-R objective audio quality measurement method. *J. Audio Eng. Soc.*, 48(3):164–173, Mar 2000.

[84] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, IT-13(4):260–269, Apr 1967.

[85] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits and Systems for Video Technology*, 13(4):560–576, Jul 2003.

[86] S. G. Wilson and S. Husain. Adaptive-tree encoding of speech at 8000 bits/s with a frequency-weighted error criterion. *IEEE Trans. Communications*, COM-27(1):165–170, Jan 1979.

[87] C-H. Yang and H-M. Hang. Cascaded trellis-based rate-distortion control algorithm for MPEG-4 advanced audio coding. *IEEE Trans. Audio, Speech and Language Processing*, 14(3):998–1007, May 2006.

[88] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. New York: Springer-Verlag, 2 edition, 1999.

[89] `http://www.ebu.ch/en/technical/publications/tech3000_series/tech3253/index.php`.

[90] http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_14496-5_2001_Software_Reference.

[91] http://www.3gpp.org/ftp/Specs/html-info/26410.htm.

Appendix

Appendix A

A.1 Proof of Equation (4.16)

Consider (4.9) that provides the pdf of x_n conditioned on all past information, $\{i_l\}^{n-1}$. This can be re-written as

$$\begin{aligned} p(x_n|\{\mathcal{I}_l\}^{n-1}) &= \frac{\int\limits_{\mathcal{I}_{n-1}} p(x_{n-1}|\{\mathcal{I}_l\}^{n-2}) p_Z(x_n - \rho x_{n-1}) dx_{n-1}}{\int\limits_{\mathcal{I}_{n-1}} p(x_{n-1}|\{\mathcal{I}_l\}^{n-2}) dx_{n-1}} \\ &= \frac{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_{n-1})} p_{E_{n-1}}(e|\{i_l\}^{n-2}) p_Z(x_n - \tilde{x}_n + \rho g_{\mathcal{Q}}(i_{n-1}) - \rho e) de}{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_{n-1})} p_{E_{n-1}}(e|\{i_l\}^{n-2}) de} \end{aligned} \quad \text{(A.1)}$$

The first equality is by application of (4.10) as employed at time $n-1$. Substitution of $x_{n-1} = e + \tilde{x}_{n-1}$, the equivalence $\{i_l\}^n \Leftrightarrow \{\mathcal{I}_l\}^n$, and recognizing that $\tilde{x}_n = \rho g_{\mathcal{Q}}(i_{n-1}) + \rho \tilde{x}_{n-1}$ for the predictor (4.2), yields the second equality. We employ the subscript E_{n-1} to indicate that $p_{E_{n-1}}(e|\{i_l\}^{n-2})$ is the prediction error pdf at time $n-1$ (conditioned on the relative past $\{i_l\}^{n-2}$). Let $\mathbb{P}$ be the set of functions which are valid pdfs in $\mathbb{R}$. Define the functional $\Phi(x, i, p(\cdot)) : \mathbb{R} \times \mathbb{I} \times \mathbb{P} \to \mathbb{R}$ as:

$$\Phi(x, i, p(\cdot)) = \frac{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i)} p(e) p_Z(x + \rho g_{\mathcal{Q}}(i) - \rho e) de}{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i)} p(e) de} \,. \quad \text{(A.2)}$$

Then (A.1) implies that, $p(x_n|\{\mathcal{I}_l\}^{n-1})$, is the above functional evaluated at $(x_n - \tilde{x}_n, i_{n-1}, p_{E_{n-1}}(e|\{i_l\}^{n-2}))$.

Claim 3: In case of the matched predictor (4.2), a time invariant quantizer $\mathcal{Q}$, and the stationary process (4.1), $p(x_n|\{\mathcal{I}_l\}^{n-1})$ can be obtained by the evaluation

of a functional of the form $\Phi(x, \{i_l\}_0^{L'-1}, p(\cdot))$, at $(x_n - \tilde{x}_n, \{i_l\}_{n-L'}^{n-1}, p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1}))$, $\forall L' \geq 0$. (At $L' = 0$, set $\{i_l\}_0^{-1} = \{i_l\}_n^{n-1} = \{\}$)

Proof sketch: The case $L' = 0$ is trivial: the functional is just $\Phi(x, \{\}, p(\cdot)) = p(x)$. It simply maps the (conditional) prediction error pdf at time n to the conditional pdf of x_n, via $p(x_n|\{\mathcal{I}_l\}^{n-1}) = p_{E_n}(x_n - \tilde{x}_n|\{i_l\}^{n-1})$. The case $L' = 1$ is already proved. Now apply induction on L'.

Claim 1 (see Sec. 4.2.2) and Claim 3 along with (4.8) and (4.10) provide the following expression for the optimal estimate of (4.7).

$$\begin{aligned}
\hat{x}_n^* &= \frac{\int\limits_{\mathcal{I}_n} x_n p(x_n|\{\mathcal{I}_l\}^{n-1}) p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n) dx_n}{\int\limits_{\mathcal{I}_n} p(x_n|\{\mathcal{I}_l\}^{n-1}) p(\{\mathcal{I}_l\}_{n+1}^{n+L}|x_n) dx_n} \\
&= \frac{\int\limits_{\mathcal{I}_n} x_n \Phi(x_n - \tilde{x}_n, \{i_l\}_{n-L'}^{n-1}, p_{E_{n-L'}}(\cdot)) \Lambda(x_n - \hat{x}_n, \{i_l\}_{n+1}^{n+L}) dx_n}{\int\limits_{\mathcal{I}_n} \Phi(x_n - \tilde{x}_n, \{i_l\}_{n-L'}^{n-1}, p_{E_{n-L'}}(\cdot)) \Lambda(x_n - \hat{x}_n, \{i_l\}_{n+1}^{n+L}) dx_n}
\end{aligned}$$

Substituting $x_n = e_n + \tilde{x}_n$, and $\hat{x}_n - \tilde{x}_n = g_{\mathcal{Q}}(i_n)$ yields,

$$\begin{aligned}
\hat{x}_n^* &= \tilde{x}_n + \frac{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_n)} e_n \Phi(e_n, \{i_l\}_{n-L'}^{n-1}, p_{E_{n-L'}}(\cdot)) \Lambda(e_n - g_{\mathcal{Q}}(i_n), \{i_l\}_{n+1}^{n+L}) de_n}{\int\limits_{\mathcal{I}_{\mathcal{Q}}(i_n)} \Phi(e_n, \{i_l\}_{n-L'}^{n-1}, p_{E_{n-L'}}(\cdot)) \Lambda(e_n - g_{\mathcal{Q}}(i_n), \{i_l\}_{n+1}^{n+L}) de_n} \\
&\triangleq \tilde{x}_n + c'(p_{E_{n-L'}}(\cdot), \{i_l\}_{n-L'}^{n+L}) \qquad \text{(A.3)}
\end{aligned}$$

where we use the abbreviation $p_{E_{n-L'}}(\cdot)$ in place of $p_{E_{n-L'}}(e|\{i_l\}^{n-L'-1})$.

□