UNIVERSITY OF CALIFORNIA
Santa Barbara

Advances in audio coding and networking by effective exploitation of long term correlations

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

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Tejaswi Nanjundaswamy

Committee in Charge:

Professor Kenneth Rose, Chair

Professor Jerry Gibson

Professor John Shynk

Professor Shivkumar Chandrasekaran

March 2013

The Dissertation of
Tejaswi Nanjundaswamy is approved:

Professor Jerry Gibson

Professor John Shynk

Professor Shivkumar Chandrasekaran

Professor Kenneth Rose, Committee Chairperson

March 2013

Advances in audio coding and networking by effective exploitation of long term correlations

by

Tejaswi Nanjundaswamy

To my parents,

S. Nanjundaswamy and B. S. Jayashree

Acknowledgements

I would like to profusely thank my advisor Prof. Kenneth Rose, for providing me the opportunity to work with him in the renowned Signal Compression Lab (SCL) at UCSB. This dissertation would not have been possible without the valuable time and effort he has spent on guiding me through all the traps and tricks of doctoral research work. His enthusiasm for tackling complex research problems has been contagious and there is always so many things to learn from him. He is easily the best advisor anyone can have and he is also the best teacher I have ever had in my life. I would also like to take this opportunity to thank NSF for funding my research through the grant CCF-0917230.

I would like to thank all my committee members Prof. Jerry Gibson, Prof. John Shynk, and Prof. Shivkumar Chandrasekaran for being on my committee, providing valuable feedback during my qualifying exam, and reviewing my dissertation. I would also like to thank all professors of the ECE department for teaching excellent courses that helped me concretize my fundamentals critical for research.

I would like to thank my lab-mates: Vinay, who as my early collaborator played an important role in formative times of my PhD and helped me understand the nitty gritty of conducting research; Emmanuel, collaboration with whom helped me identify some of the fundamental problems this work addresses; Emrah, who

has always been a great critique; and mainly Kumar, who has really been my go-to person to discuss any topic/problem and such discussions have always helped me improve my understanding. I would also like to thank all my current and past lab-mates, Ankur, Pradeep, Emre, Jingning, Rahul, Min-Chi, Renuka, Mustafa, and Yue, for making my stay at SCL greatly enjoyable. I will always be highly grateful to all the office staff at UCSB, especially Valerie De Veyra of ECE department, for having always been helpful and taking up the burden of all the paper work / office work, to let me focus on my studies. I would like to also specifically thank everyone who volunteered to perform listening tests to evaluate subjective quality, which was very critical in validating all my research. I would like to also thank Lauren, who as part of the 2012 RMP program, developed a better subjective quality evaluation tool.

I would like to thank all my friends at UCSB who have been a great company to me during my PhD. Kumar and Karthikeyan, have been my longest roommates ever, have played a significant role in helping me enjoy and learn a lot from all the courses we have taken together, have always given valuable feedback about my research, and have helped me in every other aspect of life. I would to thank all my others friends at UCSB who have always helped me with everything and made my stay extremely enjoyable by being part of all the fun get-togethers, potlucks, parties, treats, and road-trips: Malavika, Sandeep, Karthik, Murali (the Brinjal

curry specialist), Sharath, Sourav, Fitz, Priya, Manasa, Divya, Avantika, Vivek, Vineeth, Dinesh, Aseem, Vinayak, Shashank, Gaurav, Puneet, Deblina.

I would like to thank others from before my time at UCSB, who have played a significant role in my life: Prof. Sumam David and Prof. P. Subbanna Bhat - a lot of credit to where I am and who I am today goes to these two exceptionally good professors and mentors from NITK; Teachers from Marimallappa Junior College, Mysore; Mr. Shivashankar, my Physics teacher at Mysore; Teachers from Seventh Day Adventist High School, Kolhapur; Sukanya Chandramouli and Shantanu Jha, Ittiam; All my ex-colleagues from Ittiam; All my friends from NITK, amongst whom I would like to specially thank Aravind Alagappan (also one of my ex-colleague at Ittiam), who gave that final push of motivation for PhD; And also all my other friends.

Finally, but most importantly I would like to thank my family: my parents, who have always been loving, caring and supportive to me, and have gone to great lengths to provide me everything; my sister, Pooja, who has been my best friend for life; my grandparents, who have been exemplary role models to me all my life; I was also extremely lucky that during my time at UCSB I found my perfect life partner, Priyanka, who has been loving, caring, supportive and a great influence on me; And all my other family members who have always encouraged my endeavors.

Obviously a lot of people have played an important role in shaping my life, hence the long acknowledgement. I also thank everyone who had the patience to read it completely.

Curriculum Vitæ

Tejaswi Nanjundaswamy

Education

2009	Master of Science in Electrical Engineering, University of California, Santa Barbara, USA.
2004	Bachelor of Engineering, National Institute of Technology Karnataka, Surathkal, India

Experience

2009 – 2013	Graduate Student Researcher, Signal Compression Lab, University of California, Santa Barbara.
2004 – 2008	Senior Engineer, Audio Group, Ittiam Systems, Bangalore, India.
2003	Project Intern, Multimedia Codecs Group, Texas Instruments, Bangalore, India.

Publications

- T. Nanjundaswamy and K. Rose, "Cascaded long term prediction for efficient compression of polyphonic audio signals," to be submitted to *IEEE Trans. Audio, Speech, and Language Processing.*

- T. Nanjundaswamy and K. Rose, "Frame loss concealment of polyphonic audio signals based on cascaded long term prediction," to be submitted to *IEEE Trans. Audio, Speech, and Language Processing.*

- T. Nanjundaswamy, V. Melkote, E. Ravelli and K. Rose, "Perceptual distortion-rate optimization of prediction tools in audio coders," to be submitted to *IEEE Trans. Audio, Speech, and Language Processing.*

- E. Ravelli, V. Melkote, T. Nanjundaswamy and K. Rose, "Joint optimization of base and enhancement layers in scalable audio coding," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 4, April 2013.

- T. Nanjundaswamy and K. Rose, "On accommodating pitch variation in long term prediction of speech and vocals in audio coding," Proc. of *133rd AES Convention*, Preprint 8767, Oct 2012.

- K. Viswanatha, E. Akyol, T. Nanjundaswamy and K. Rose, "On common information and encoding of sources that are not successively refinable", Proc. of *IEEE Information Theory Workshop (ITW)*, Sep 2012.

- T. Nanjundaswamy and K. Rose, "Bidirectional cascaded long term prediction for frame loss concealment in polyphonic audio signals," Proc. of *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, March 2012.

- T. Nanjundaswamy and K. Rose, "Perceptually optimized cascaded long term prediction of polyphonic signals for enhanced MPEG-AAC," Proc. of *131st AES Convention*, Preprint 8518, Oct 2011.

- T. Nanjundaswamy and K. Rose, "Cascaded long term prediction for coding polyphonic audio signals," Proc. of *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2011. **(Best Paper Award Finalist)**

- T. Nanjundaswamy, V. Melkote, E. Ravelli and K. Rose, "Perceptual distortion-rate optimization of long term prediction in MPEG AAC," Proc. of *129th AES Convention*, Preprint 8288, Nov 2010. **(Winner of Student Technical Paper Award)**

- E. Ravelli, V. Melkote, T. Nanjundaswamy and K. Rose, "Cross-layer rate-distortion optimization for scalable advanced audio coding," Proc. of *128th AES Convention*, Preprint 8084, May 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. of *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, March 10.

Abstract

Advances in audio coding and networking by effective exploitation of long term correlations

Tejaswi Nanjundaswamy

This dissertation focuses on tackling challenges related to efficient transmission of all varieties of audio signals over networks, which mainly are, compression at delay constraints acceptable for communication applications, and dealing with loss of content due to noisy channels. Efficiently exploiting long term correlations is key to address these challenges.

For audio compression, while there are many well known techniques that are effective in exploiting redundancies within a frame, the only solution known for inter-frame redundancy removal is the naive technique of using a simple long term prediction (LTP) filter, which provides a segment of previously reconstructed samples as prediction for the current frame. Although this technique can at least be effective for audio signals with a single stationary periodic component (i.e., monophonic), the typically employed parameter selection based on minimizing the mean squared error as opposed to the perceptual distortion criteria of audio coding, hinders the performance of LTP. This drawback is first addressed by employing a novel two-stage parameter estimation technique which jointly optimizes

LTP parameters along with quantization and coding parameters, while explicitly accounting for the perceptual distortion and rate tradeoffs. However, since most audio signals are polyphonic in nature, containing a mixture of several periodic components, the LTP tool due to its simplistic structure is well known to be ineffective. This major drawback is addressed by employing a sophisticated filter structure of cascading multiple LTP filters, each corresponding to individual periodic component. Also a recursive "divide and conquer" technique is introduced to estimate parameters of all the LTP filters in the cascade. Effectiveness of cascaded LTP for compression is demonstrated in two distinct settings of the ultra low delay Bluetooth Subband Codec and the MPEG Advanced Audio Coding (AAC) standard. In MPEG AAC, we specifically adapt the cascaded LTP parameter estimation to take into account the perceptual distortion criteria, and also propose a low decoder complexity variant. Another shortcoming of the LTP tool used in audio coders is its subpar performance for speech and vocal content, which is well known to be quasi-periodic and involve small variations in pitch period. This drawback is addressed by employing a novel technique of introducing a single parameter of 'geometric' warping in the LTP filter, whereby past periodicity is geometrically warped to provide an adjusted prediction for the current samples. Again the parameter estimation for this modified LTP filter is adapted to take the perceptual distortion criteria into account. Objective and subjective results

for all the settings validate the effectiveness of the proposals on a variety of audio signals.

For dealing with loss of content due to noisy channels, concealment techniques based on LTP filtering are well known and are suitable for audio signals with single periodic component. However, none of the existing techniques are designed to overcome the main challenge due to the polyphonic nature of most music signals. This shortcoming is addressed by employing the cascaded LTP filtering to effectively estimate every periodic component from all the available information. Objective and subjective evaluation results for the proposed approach, in comparison with existing techniques, all incorporated within an MPEG AAC low delay decoder, provide strong evidence for considerable gains across a variety of polyphonic signals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A wide range of multimedia applications such as internet radio and television, online media streaming, gaming, and high fidelity teleconferencing heavily rely on efficient transmission of audio signals over networks. The two main challenges for such transmission is delay constrained compression, and dealing with loss of content due to noisy channels. Constraints on delay means that the algorithms can only operate on small block sizes (or frame lengths). Thus the key to addressing these challenges is efficiently exploiting inter-frame redundancies due to long term correlations. While well known audio coders are effective in eliminating redundancies within a block of data, and the only known inter-frame redundancy removal technique of employing a long term prediction (LTP) filter is too simplistic, as it is suboptimal for the commonly occurring polyphonic audio signals, which contain a mixture of several periodic components, and also suboptimal for speech and vocal content, which is quasi-periodic with small variations in pitch

period. Moreover the typically employed parameter estimation technique is mismatched to the ultimate perceptual distortion criteria of audio coding. Similarly even in loss concealment, none of the existing techniques are designed to overcome the main challenge due to the polyphonic nature of most music signals.

This dissertation focusses on addressing all these shortcomings by employing novel sophisticated filter structures suitable for a wide variety of audio signals, and the parameters of such filters are estimated by, taking into account the perceptual distortion criteria for audio compression, and utilizing all the available information for loss concealment. We first provide background information about audio coding and the LTP in Chapter 2, and then we propose solutions to address different aspects of the challenges in the following chapters. An overview of these chapters is provided in the following sections.

1.1 Perceptual distortion-rate optimization of prediction tools in audio coders

Most current audio coders, including the MPEG Advanced Audio Coding (AAC) standard [1], employ a modified discrete cosine transform (MDCT) whose decorrelating properties eliminate redundancies within a block of data. As audio content typically consists of naturally occurring periodic signals, there is still

potential for exploiting redundancies across frames, especially in the case of the short frame MDCT adopted in the low delay (LD) AAC mode [2]. The LTP tool [3] was proposed to close this gap. LTP was specifically targeted at signals with a single periodic component (i.e., monophonic), as it exploits repetition in the waveform by providing a segment of previously reconstructed samples, scaled appropriately, as prediction for the current frame. Typically, time domain waveform matching techniques are employed to find the past segment position (called "lag"), and the gain factor, such that the prediction minimizes the mean squared error (MSE). MPEG AAC further allows LTP to provide flags to selectively enable prediction in a subset of frequency bands, which is determined based on the prediction MSE. This technique can at least be effective for audio signals with a single periodic component that is stationary for relatively long durations. But the parameter estimation approach has significant shortcomings: A mean squared error-based choice of the LTP lag and gain is clearly suboptimal relative to the ultimate objective of minimizing a perceptually relevant weighted mean squared error criterion. Further, the lag and gain are determined by the temporal prediction residue, which ignores the potential spectral impacts of eventually switching off prediction in a subset of the frequency bands.

Motivated by the above observations, we propose in Chapter 3 a rate-distortion (RD) optimization method to jointly select the LTP parameters, and the quan-

tization and coding parameters of the core AAC encoder. This is achieved via a novel two-stage parameter estimation technique, where in the first stage, a set of S LTP parameters with the least mean squared prediction error is retained, and then in the next stage, all the S "survivors" are perceptual distortion-rate evaluated to select the final parameters which minimize the perceptual distortion for the given rate. This approach is evaluated with trellis based optimal perceptual distortion-rate evaluation [4, 5], and the low complexity suboptimal two-loop search based perceptual distortion-rate evaluation [1, 6], with objective and subjective results providing evidence for substantial gains in audio signals with single periodic component. We reuse the fundamental concept introduced in this section through out this dissertation whenever the perceptual distortion criteria has to be taken into account.

1.2 Cascaded long term prediction for efficient compression of polyphonic audio signals

The vast majority of speech and audio content consists of naturally occurring sounds which are periodic in nature. Examples include voiced parts of speech, music from string and wind instruments, etc. An audio signal with only one periodic component (i.e., a monophonic signal) obviously exhibits waveform repe-

tition, which is exploited by the LTP tool to improve compression efficiency. The tool essentially identifies a "similar" previous segment and scales it as the prediction for the current frame. However, most audio signals are polyphonic in nature, containing a mixture of several periodic components, which includes as common examples, vocals with background music, orchestra, and chorus. Note that a single instrument may also produce multiple periodic components, as is the case for the piano or the guitar. While such polyphonic signals are themselves periodic with overall period equaling the least common multiple of the individual component periods, the signal rarely remains sufficiently stationary over the extended period, rendering the LTP tool ineffective for most audio signals.

We propose to address this major drawback in Chapter 4 by employing a novel technique of exploiting the correlation of each periodic component with its immediate past, using a cascade of LTP filters, each corresponding to individual periodic component. We also introduce a recursive "divide and conquer" technique to estimate parameters of all the LTP filters in the cascade. Effectiveness of cascaded LTP for compression is demonstrated in two distinct settings of the ultra low delay Bluetooth Subband Codec (SBC) [7, 8] and the MPEG AAC standard. In MPEG AAC, the cascaded LTP parameter estimation is adapted to take into account the perceptual distortion criteria via the two-stage method introduced in Chapter 3, wherein an initial set of parameters is estimated backward adaptively to

minimize the mean squared prediction error, followed by a refinement stage where parameters are adjusted to minimize the perceptual distortion. A low decoder complexity variant, which employs forward adaptive parameter estimation, is also proposed for MPEG AAC. Objective and subjective results for all the settings validate the effectiveness of the proposal on a variety of polyphonic signals.

1.3 Accommodating pitch variations in long term prediction

The LTP tool is obviously designed for periodic signals that are stationary over relatively long durations. However, amongst the commonly occurring periodic signals in audio content, the class of voiced speech and vocals in music is well known to be quasi-stationary and is characterized by small variations in pitch period. These small changes in pitch period accumulate over the length of the frame, and substantially compromise the LTP tool effectiveness. This performance degradation relative to other stationary periodic signals of musical instruments has been extensively documented in prior LTP related research.

In Chapter 5 we address this drawback by employing a novel technique of introducing a single parameter of 'geometric' warping in the LTP filter, whereby past periodicity is geometrically warped to provide an adjusted prediction for the

current samples. The parameters of this modified LTP filter is estimated using three stages, where an unwarped LTP filter is first estimated to minimize the mean squared prediction error; then filter parameters are complemented with the warping parameter, and re-estimated within a small neighboring search space to retain the set of S best LTP parameters; and finally, a perceptual distortion-rate procedure is used to select from the S candidates, the parameter set that minimizes the perceptual distortion. Objective and subjective evaluations substantiate the proposed technique's effectiveness.

1.4 Frame loss concealment of polyphonic audio signals

Audio transmission over networks enables a wide range of applications such as multimedia streaming, online radio and high-definition teleconferencing. These applications are often plagued by the problem of unreliable networking conditions, which leads to intermittent loss of data. Frame loss concealment (FLC) forms a crucial tool amongst the various strategies used to mitigate this issue. The FLC objective is to exploit all available information to approximate the lost frame while maintaining smooth transition with neighboring frames. While, techniques suitable for audio signals with single periodic component are well known, none of

them have been designed to overcome the main challenge due to the polyphonic nature of most music signals.

This shortcoming is addressed in Chapter 6 by employing the cascaded LTP filtering (introduced in Chapter 4) to effectively estimate every periodic component from all the available information. In the first phase, a cascaded filter is designed from available past samples and is used to predict across the lost frame(s). Available future reconstructed samples allow refinement of the filter parameters to minimize the squared prediction error across such samples. In the second phase a prediction is similarly performed in reverse from future samples. Finally the lost frame is interpolated as a weighted average of forward and backward predicted samples. Objective and subjective evaluation results for the proposed approach, in comparison with existing techniques, all incorporated within an MPEG AAC low delay decoder, provide strong evidence for considerable gains across a variety of polyphonic signals.

Chapter 2

Background

This chapter provides background information on the perceptual audio coding standard of MPEG AAC in LD mode, the long term prediction technique, how LTP has been integrated in the MPEG AAC standard and the ultra low delay Bluetooth SBC. Note that the notations introduced in this chapter are carried over to other chapters.

2.1 MPEG AAC

MPEG AAC is a transform based perceptual audio coder. The AAC encoder segments the audio signal into 50% overlapped frames of $2K$ samples each ($K = 512$ in the LD mode), with frame n composed of the samples $x[m]$, $nK \leq m < (n+2)K$. These samples are transformed via MDCT to produce K transform coefficients, denoted by $c_n[k]$, $0 \leq k < K$. The transform coefficients are grouped into L frequency bands (known as scale-factor bands or SFBs) such that all the

coefficients in a band are quantized using the same scaled version of the generic AAC quantizer. For each SFB l, the scaling factor (SF), denoted by $s_n[l]$, controls the quantization noise level. The quantized coefficients (denoted by $\hat{c}_n[k]$) in an SFB are then Huffman coded using one of the finite set of Huffman codebooks (HCBs) specified by the standard, and the choice is indicated by the HCB index $h_n[l]$. We denote by $\mathbf{p}_n = (\mathbf{s}_n, \mathbf{h}_n)$ the encoding parameters for frame n, with $\mathbf{s}_n = \{s_n[0], \ldots, s_n[L-1]\}$ and $\mathbf{h}_n = \{h_n[0], \ldots, h_n[L-1]\}$. Given a target rate for the frame, the SFs and HCBs are selected to minimize the perceptual distortion. The distortion is based on the noise-to-mask ratio (NMR), calculated for each SFB as the ratio of quantization noise energy in the band to a noise masking threshold provided by a psychoacoustic model

$$d_{(n,l)}(s_n[l]) = \frac{\sum_{k \in \text{SFB } l} (c_n[k] - \hat{c}_n[k])^2}{\mu_n[l]}, \tag{2.1}$$

where $\mu_n[l]$ is the masking threshold in SFB l of frame n. The overall per-frame distortion $\mathcal{D}_n(\mathbf{p}_n)$ may then be calculated by averaging or maximizing over SFBs. In this work we define this distortion as the maximum NMR (MNMR)

$$\mathcal{D}_n(\mathbf{p}_n) = \max_{0 \le l < L} d_{(n,l)}(s_n[l]). \tag{2.2}$$

Since the standard only dictates the bitstream syntax and the decoder part of the codec, numerous techniques to optimize the encoder parameters have been proposed (e.g., [1, 4, 6, 9]). Specifically, the MPEG AAC verification model (publicly

available as informative part of the MPEG standard) optimizes the encoder parameters via a low-complexity technique known as the two-loop search (TLS) [1,6]. An inner loop finds the best SF for each SFB to satisfy a target distortion criterion for the band. The outer loop then determines the set of HCBs that minimize the number of bits needed to encode the quantized coefficients and the side information. If the resulting bit rate exceeds the rate constraint for the frame, the target distortion in the inner loop is increased and the two loops are repeated.

The bit-stream consists of quantized data and the side information, which includes, per SFB, one SF (that is differentially encoded across SFBs), and one HCB index (which is run-length encoded across SFBs). For simplicity, except for the LTP tool, we do not consider optional tools available in the MPEG framework, such as the bit reservoir, window shape switching, temporal noise shaping, etc.

2.2 Long Term Prediction

Transform and subband coders efficiently exploit correlations within a frame, but the frame size is often limited by the delay constraints of an application. This motivates inter-frame prediction, especially for low delay coders, to remove redundancies across frames, which otherwise would have been captured by a long block transform. One technique for exploiting long term correlations has been

well known since the advent of predictive coding for speech [10], and is called pitch prediction, which is used in the quasi-periodic voiced segments of speech. The pitch predictor is also referred to as long term prediction filter, pitch filter, or adaptive codebook for a code-excited linear predictor. The generic structure of such a filter is given as

$$H(z) = 1 - \sum_{k=0}^{T-1} \beta_k z^{-N+k}, \tag{2.3}$$

where N corresponds to the pitch period, T is the number of filter taps, and β_k are the filter coefficients. This filter and its role in efficient coding of voiced segments in speech, have been extensively studied. A thorough review and analysis of various structures for pitch prediction filters is available in [11]. Backward adaptive parameter estimation was proposed in [12] for low-delay speech coding, but forward adaption was found to be advantageous in [13]. Different techniques to efficiently transmit the filter information were proposed in [14] and [15]. The idea of using more than one filter taps (i.e., $T > 1$ in equation (2.3)) was originally conceived to approximate fractional delay [16], but has been found to have broader impact in [17]. Techniques for reducing complexity of parameter estimation have been studied in [18] and [19]. For a review of speech coding work in modeling periodicity, see [20].

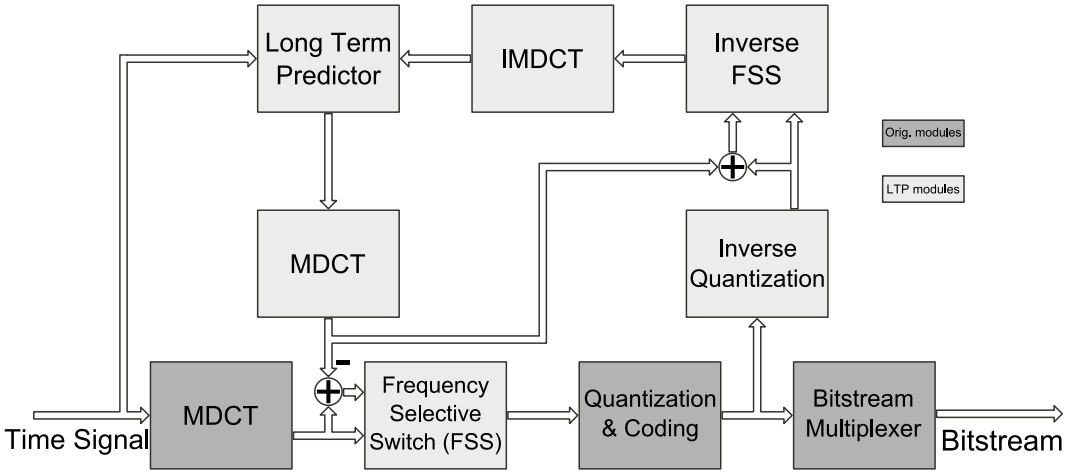

Figure 2.1: Block Diagram of an MPEG AAC coder with LTP.

2.3 Long Term Prediction tool in MPEG AAC

Long term prediction is prevalent in speech coding techniques, and has also been proposed as an optional tool for the audio coding standard of MPEG AAC. This tool was specifically targeted at the LD mode, to compensate for loss in compression performance due to the short frame size of MDCT. A representative block diagram of the LTP tool in an MPEG AAC coder is shown in Fig. 2.1. This section builds on the notation introduced for the MPEG AAC standard in Section 2.1 and describes the LTP parameter selection technique specified in the publicly available informative/non-mandatory part of the MPEG standard. Let the source samples of frame n be $x[m]$, $nK \leq m < (n+2)K$, and let $\hat{x}[m]$ be the sequence of previously reconstructed samples obtained by decoding up to frame $n-1$. Note that the samples $\hat{x}[m]$, $nK \leq m < (n+1)K$, are only partially reconstructed,

due to the inverse MDCT requirement of overlap and add with a portion of the current frame. The LTP tool predicts the current frame from an equally long past segment in $\hat{x}[m]$, the beginning of which (relative to the first sample in frame n) is indicated by the LTP lag, $\mathbf{L}_n$. This lag takes value in $\{K, \ldots, 3K-1\}$, and it is possible that a portion of the $2K$ length prediction segment contains partially reconstructed samples. This segment is subsequently scaled by gain $\mathbf{G}_n$, which is selected from a set of 8 values. Thus the LTP analysis filter is of the form

$$H_{\mathrm{LTP}}(z) = 1 - \mathbf{G}_n z^{-\mathbf{L}_n}, \tag{2.4}$$

and the prediction of the current frame is denoted as

$$\tilde{x}_n[m] = \mathbf{G}_n \hat{x}[m + nK - \mathbf{L}_n],\ 0 \leq m < 2K\ . \tag{2.5}$$

These LTP lag and gains are selected such that they minimize the mean squared prediction error cost:

$$\varepsilon = \sum_{m=0}^{2K-1} (x[m + nK] - \tilde{x}_n[m])^2. \tag{2.6}$$

For a given $\mathbf{L}_n$, $\mathbf{G}_n$ is optimized by setting the partial derivatives of ε with respect to $\mathbf{G}_n$ to 0, leading to the following choice of lag and gain parameters

$$\mathbf{L}_n = \underset{\mathbf{L}\in[K,3K)}{\arg\max} \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-\mathbf{L}]}{\sqrt{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-\mathbf{L}]}} \tag{2.7}$$

$$\mathbf{G}_n = \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-\mathbf{L}_n]}{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-\mathbf{L}_n]} \tag{2.8}$$

This gain factor is subsequently quantized.

Next, the predicted frame of samples is transformed via MDCT to produce K transform coefficients denoted $\tilde{c}_n[k]$, $0 \leq k < K$. The per transform coefficient prediction residue is $e_n[k] = c_n[k] - \tilde{c}_n[k]$. The standard further provides the flexibility to selectively enable LTP in different SFBs and the choice is indicated by a per-SFB bit flag $f_n[l]$. This flag is set whenever the prediction residue energy is lower than the signal energy in the band,

$$f_n[l] = \begin{cases} 1, & \text{if } \sum_{k\in\text{SFB } l} e_n^2[k] < \sum_{k\in\text{SFB } l} c_n^2[k] \\ 0, & \text{otherwise.} \end{cases} \tag{2.9}$$

A global flag $\mathbf{F}_n$ enables/disables LTP on a per-frame basis, contingent on the coding gain provided by this tool. This flag is set based on a heuristic estimate

of the bit savings due to LTP, given by

$$B_n = \frac{1}{6}\sum_{l=0}^{L-1} 10\log_{10}\left[\frac{\sum\limits_{k\in \text{SFB } l} c_n^2[k]}{\min\left(\sum\limits_{k\in \text{SFB } l} c_n^2[k], \sum\limits_{k\in \text{SFB } l} e_n^2[k]\right)}\right] K_l \tag{2.10}$$

where K_l is the number of coefficients in the SFB l. The above estimate assumes the "rule of thumb" of 1 bit savings for every 6 dB of prediction gain. The global flag is set as

$$\mathbf{F}_n = \begin{cases} 1, & \text{if } B_n > \text{LTP side information rate} \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

The output for quantization $\forall k \in$ SFB l is denoted as

$$q_n[k] = \begin{cases} e_n[k], & \text{if } f_n[l] = 1 \text{ and } \mathbf{F}_n = 1, \\ c_n[k], & \text{otherwise} \end{cases} \tag{2.12}$$

These coefficients $q_n[k]$ are quantized to $\hat{q}_n[k]$ and coded via the technique described in Section 2.1. The augmented set of encoding parameters of frame n are $\mathbf{p}_n = (\mathbf{s}_n, \mathbf{h}_n, \mathbf{f}_n, \mathbf{U}_n)$, with $\mathbf{f}_n = \{f_n[0], \ldots, f_n[L-1]\}$, and $\mathbf{U}_n = \{\mathbf{L}_n,\ \mathbf{G}_n,\ \mathbf{F}_n\}$. The NMR in each SFB is now calculated as,

$$d_{(n,l)}(s_n[l], f_n[l], \mathbf{U}_n) = \frac{\sum\limits_{k\in \text{SFB } l} \left(q_n[k] - \hat{q}_n[k]\right)^2}{\mu_n[l]}. \tag{2.13}$$

Note the additional dependency of NMR on LTP parameters, as it is evident from (2.5) and (2.12) that the LTP parameters influence the values of the coefficients

in the SFB. The MNMR is calculated as,

$$\mathcal{D}_n(\mathbf{p}_n) = \max_{0 \le l < L} d_{(n,l)}(s_n[l], f_n[l], \mathbf{U}_n). \tag{2.14}$$

2.4 Bluetooth SBC

The Bluetooth Sub-band Codec (SBC) [7,8] employs a simple ultra-low-delay compression technique for use in short range wireless audio transmission. The SBC encoder blocks the audio signal into frames of BK samples, where samples of frame n are denoted $x[m]$, $nBK \le m < (n+1)BK$. The frame is analyzed into $B \in \{4 \text{ or } 8\}$ subbands with $K \in \{4, 8, 12 \text{ or } 16\}$ samples in each subband, denoted $c_n[b,k],\ 0 \le b < B,\ 0 \le k < K$. The analysis filter bank is similar to the one in MPEG Layer 1-3 [21], but has a filter order of $10B$, with history requirement of $9B$ samples, while analyzing B samples of input at a time. The block of K samples in each sub-band is then quantized adaptively to minimize the quantization MSE. The effective scalefactor $s_n[b],\ 0 \le b < B$ for each subband is sent to the decoder as side information. Note that the FIR filter used in the analysis filter bank introduces a delay of $(9B+1)/2$ samples. The decoder receives the quantization step sizes and the quantized data in the bitstream. The subband data is dequantized and input to the synthesis filter bank (similar to the one used

in MPEG Layer 1-3) to generate the reconstructed output signal. The analysis and synthesis filter banks together introduce a delay of $(9B + 1)$ samples.

Chapter 3

Perceptual distortion-rate optimization of prediction tools in audio coders

3.1 Introduction

The MPEG AAC standard described in Section 2.1 along with the LTP tool described in Section 2.3 should at least be effective in exploiting inter-frame redundancies for audio signals with a single periodic component that is stationary for relatively long durations. But the performance of the LTP tool even for such signals is critically hampered by the typically employed parameter estimation. The objective of a perceptual audio encoder is, generally, to find encoding parameters that minimize a perceptually relevant distortion criterion under a rate constraint. Clearly, the typical procedure employed for LTP parameter value selection as described in Section 2.3, that follows minimization of a mean squared

prediction error criterion, cannot guarantee parameter choices that minimize a perceptually relevant criterion. Further, the determination of the lag and gain parameters by (2.7) and (2.8) ignores the fact that LTP does not uniformly impact the whole spectrum due to the per-band prediction switch determined by (2.9). Additionally, decoupling the selection of LTP parameter values from the subsequent quantization and coding parameter value selection, as is the case in current encoders, can lead to sub-optimal performance: the values selected for the LTP parameters influences what is coded in the quantization and coding process.

Motivated by the above observations, we propose in this chapter a rate-distortion (RD) optimization method to jointly select the LTP parameters, and the quantization and coding parameters (SFs and HCBs). An exhaustive search to find the optimal set is computationally prohibitive. The LTP lag has a range as large as the frame length, and the gain takes value in a quantized set of eight levels. Each lag-gain combination corresponds to a different prediction, and is further combined with a set of per band parameter values: LTP flags, SFs, and HCBs. Thus, the proposed approach reduces the search space by identifying a small set of "prediction survivors" (corresponding to LTP lag-gain pairs), which are then fully evaluated in terms of their RD performance.

The optimal per-band LTP flags, SFs and HCBs are now determined for each survivor via an efficient trellis-based search. This forms an extension to the previ-

ously introduced trellis-based search in [4, 5], with the addition of per-band LTP flags to the quantization and coding parameters. The trellis stages correspond to scalefactor bands, and nodes/states within a stage correspond to choice of parameter values for the band. The best path through the trellis, which minimizes an RD cost function, is determined by dynamic programming. The per-survivor minimum costs are then compared to choose the optimal LTP lag and gain. A low complexity variant of the proposed approach forgoes the trellis, but retains the multiple prediction survivors. This approach instead subsumes in it a greedy two loop search [1, 6] for SF and HCB selection. Objective and subjective evaluations demonstrate the gains achieved by using the proposed approaches. The results of this work have appeared in [22]. Note that the proposed approaches in this chapter are modifications to the encoder only and thus the bit-stream generated is standard compliant. Also note that while emphasis in this chapter is on the Low Delay mode of AAC, the proposed approach itself is easily extended to other modes and other coders.

This chapter is structured as follows: The proposed perceptual distortion-rate optimization is described in Section 3.2. Results are presented in Section 3.3, and the chapter concludes in Section 3.4.

3.2 Proposed perceptual distortion-rate optimization of LTP parameters

We propose here an approach for LTP parameter value selection that explicitly minimizes a perceptual distortion criterion under a rate constraint. We first formally state the problem addressed by the proposed method, and then provide the proposed algorithms.

3.2.1 Problem statement

The objective of the proposed approach is to find the optimal parameters $\mathbf{p}_n^*$ that minimize the distortion defined in (2.14) under a rate-constraint for the frame, i.e.,

$$\begin{aligned} \mathbf{p}_n^* = \quad & \arg\min_{\mathbf{p}_n} \mathcal{D}_n(\mathbf{p}_n) \\ & \text{s .t. } \mathcal{R}_n(\mathbf{p}_n) \leq \mathcal{R}_t \end{aligned} \tag{3.1}$$

where $\mathcal{R}_t$ denotes the target rate for the frame. Note that by definition of $\mathbf{p}_n$ the above implies a jointly optimized selection of LTP parameters, SFs, and HCBs for a frame.

3.2.2 Proposed trellis-based solution for optimal parameter value selection

In [4, 5], a trellis-based approach for optimal selection of SFs and HCBs for a frame was proposed from our lab, as an alternative to the sub-optimal TLS described in Section 2.1. This method was developed in the framework of an AAC encoder with no LTP, i.e., this tool was disabled. In other words, the method solved the problem in (3.1) but with minimization only over the choice of SFs and HCBs. The trellis consisted of stages corresponding to SFBs, with states in each stage corresponding to different possible SF-HCB pairs. Each state was associated with distortion and rate costs corresponding to quantizing and coding of the coefficients in that SFB. Further, transition between states were associated with rate costs required to differentially encode SFs and runlength encode the HCBs. A path through the trellis comprised of a selection of SFs and HCBs for the whole frame. A combination of the Lagrangian technique and dynamic programming was pursued to find the optimal path through the trellis that minimizes a perceptually relevant distortion criterion under a rate constraint. This path then corresponded to the optimal set of SFs and HCBs. More details of this trellis approach can be found in [4,5]. We note that in [9] a mixed integer linear

programming-based approach was proposed to the same problem of quantization and coding parameter value selection.

The proposed approach here for optimally selecting LTP parameter values builds upon the trellis-based approach of [4, 5]. Note that for any fixed choice of the LTP parameters $\mathbf{U}_n$ and $\mathbf{f}_n$, (2.12) completely determines the spectral coefficients to be coded into the bit stream, and the necessary quantization and coding process simply involves selection of SFs and HCBs for a frame. Thus, for a fixed choice of LTP parameters the same trellis as in [4, 5] can be employed to find the optimal SFs and HCBs. Simply put, the proposed approach endeavors to find the optimal set of SFs and HCBs for each combination of LTP parameters, i.e., the SFs and HCBs that minimize the MNMR. The optimal LTP parameters is then that combination with the least overall cost. But as described in Section 2.3 the LTP lag can be one of $2K$ different values while the gain has a range of 8 discrete levels. Further, the per-SFB flag $f_n[l]$ could be turned off or on. A brute force search through all $8 \times 2K \times 2^L$ combinations of these parameters is computationally prohibitive. We thus follow a two pronged strategy to reduce the complexity:

- First, the per-SFB LTP flags are absorbed into the trellis that determines the SFs and HCBs. Each state of the trellis is further split into two separate states, one corresponding to a 0 value for the LTP flag of the SFB, and

another where the flag is set. Thus every path in this modified trellis now corresponds to a specific choice of the parameters $\mathbf{s}_n$, $\mathbf{h}_n$, and $\mathbf{f}_n$. The same dynamic programming approach of [4, 5] is then employed to find the best path through this modified trellis.

- Rather than evaluating all possible combinations of LTP lag and gain parameters, we reduce the search space by identifying a small set of "prediction survivors", each corresponding to a LTP lag-gain pair. Every survivor is then individually examined in terms of its perceptual distortion-rate performance. Note that the current approach determines these parameters prior to the quantization and coding process via (2.7) and (2.8), and thus in comparison to the proposed approach simply retains a single survivor.

We now elaborate further on the selection of prediction survivors. Note that the lag values which maximize the cross-correlation in (2.7), between the current frame and the reconstructed samples, minimize the mean squared prediction error. We evaluate the following cross-correlation $\forall \mathbf{L} \in \{K, \ldots, 3K-1\}$,

$$R_n[\mathbf{L}] = \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-\mathbf{L}]}{\sqrt{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-\mathbf{L}]}}. \tag{3.2}$$

The lags corresponding to the highest P cross-correlations are retained. The unquantized gain value for each lag survivor is evaluated similar to (2.8):

$$\mathbf{G}_n[\mathbf{L}] = \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-\mathbf{L}]}{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-\mathbf{L}]} \tag{3.3}$$

Among the 8 possible quantized gain values the Q closest values to $\mathbf{G}_n[\mathbf{L}]$ are determined. The lag and gain parameters thus obtained correspond to PQ prediction survivors. Although this pruning of the search space is still based on the mean squared error criterion, retaining a larger number of survivors than just one results in a high probability of the presence of perceptually meaningful survivors in the shortlist.

Each of these prediction survivors is now evaluated in terms of its perceptual distortion-rate performance. We assume that the global LTP flag $\mathbf{F}_n$ is set.[1] Thus for any survivor $\mathbf{U}_n$ is completely determined. The minimization in (3.1) is now only over the SFs, HCBs, and per-SFB LTP flags, and the modified trellis-based approach described previously can be employed to find the optimal values of these parameters.

The overall algorithm can be enumerated as follows:

1. P lag survivors are determined based on the correlation criterion in (3.2).

[1]When $\mathbf{F}_n$ is reset LTP is disabled, and (3.1) simply boils down to the SF and HCB selection problem addressed in [4, 5]. This can be treated as a separate case.

2. Q gain survivors are determined for each lag survivor.

3. The optimal set of parameters $\mathbf{s}_n$, $\mathbf{h}_n$, and $\mathbf{f}_n$, and corresponding distortion, for each of the PQ prediction survivors, are found via the modified trellis-based algorithm.

4. The optimal SFs and HCBs, and the corresponding distortion achieved, for the case when LTP is disabled, are also found, via the original trellis-based algorithm to help decide the global flag $\mathbf{F}_n$.

5. Amongst the $PQ + 1$ cases, the parameters that result in the minimum distortion are employed in encoding the frame.

Naturally, increasing the number of prediction survivors improves the RD performance, thus providing a trade-off between complexity and quality.

3.2.3 A low complexity variant of the proposed approach

Although the trellis-based approach employs dynamic programming, it can still be substantially complex, and we thus propose a simplified TLS-based solution as well. This algorithm is summarized below:

1. PQ number of prediction survivors are found as described before.

2. For each survivor the per-band LTP flags are decided as in current encoders via (2.9).

3. Given the LTP parameters of each survivor, the corresponding SFs and HCBs are determined via TLS, and the associated distortion calculated.

4. TLS is also employed for the case when LTP is disabled (to help decide the $\mathbf{F}_n$ flag).

5. Amongst the $PQ+1$ cases, the parameters that correspond to the minimum distortion are employed in encoding the frame.

Note that although this approach employs the TLS, it still retains a number of prediction survivors and thus addresses two major drawbacks of existing encoders: a one-shot selection of LTP lag and gain indices based on minimizing a mean squared error criteria, and heuristic estimation of the prediction coding gain.

3.3 Results

In this section, we evaluate the proposed approaches for LTP parameter value selection against the existing approach, in the framework of the AAC-LD standard. It is emphasized that the proposed modifications in this chapter only apply to the

encoder, and thus generate a standard compatible bitstream. Six different versions of the AAC-LD encoder are compared:

1. MPEG reference AAC-LD encoder with LTP disabled: This approach employs TLS for SF and HCB selection, and is henceforth referred to as "nopred-TLS".

2. MPEG reference with LTP enabled: also employs TLS, determines LTP parameters via heuristic equations of Section 2.3, and will be referred to as "stdLTP-TLS".

3. AAC-LD encoder with trellis-based quantization and coding parameters selection and LTP disabled: TLS is replaced by the trellis-based search for SFs and HCBs, but is otherwise similar to nopred-TLS. This coder is further referred to as "nopred-Trel".

4. AAC-LD encoder with trellis-based quantization and coding parameters selection and standard LTP enabled: Employs trellis for SF and HCB selection only, and heuristics of Section 2.3 are used to determine the LTP parameters. This coder is henceforth referred to as "stdLTP-Trel".

5. Proposed jointly optimized selection of LTP and quantization and coding parameters via trellis and multiple survivor retainment: the approach described in Section 3.2.2. This is referred to as "propLTP-Trel".

6. Proposed two-loop search-based low-complexity coder: This is the proposed approach described in Section 3.2.3, and is herein referred to as "propLTP-TLS".

All coders employ a simple psychoacoustic model based on the MPEG reference software. In the case of propLTP-Trel and propLTP-TLS, 120 prediction survivors are retained, with $P = 20$ and $Q = 6$.

The coders were evaluated using a subset of the standard MPEG and EBU SQAM database. We select only a 10 seconds part of each audio file (which are single channel at 48/44.1kHz) to reduce computation and evaluation times. This results in the following dataset:

- Single instrument/Monophonic: harpsichord, accordion, and flute
- Polyphonic with dominant instrument: mfv
- Complex polyphonic: haffner
- Speech signal: mgerman

3.3.1 Objective evaluation results

For a thorough objective evaluation, all the aforementioned coders were evaluated at bit-rates in the range of 20 to 45 kbps. The distortion (MNMR) was

calculated for each frame, and then averaged across frames to arrive at a single distortion value for each file. The average MNMR (AMNMR) achieved at different bit-rates for each sample in the dataset is shown in Fig. 3.1.

The sub-optimality of the current approach for LTP parameter selection is underscored by the fact that stdLTP-TLS almost always performs worse than nopred-TLS, and by up to 3 dB at certain rates. This observation clearly explains the poor deployment of LTP, and indicates a need for improved optimization techniques for selection of LTP parameter values. This also implies that any subjective evaluation (Section 3.3.2) of the proposed codecs need only include nopred-TLS for comparison (stdLTP-TLS anyway has worse or similar performance).

Next, we observe that the proposed approach propLTP-Trel provides substantial gains (in the range of 5 to 9 dB at different bit-rates) over the standard method, nopred-TLS. In order to extract the gains specifically due to the proposed LTP modifications we compare propLTP-Trel against nopred-Trel and stdLTP-Trel. We must emphasize that both latter coders are not standard approaches, and employ the trellis algorithm for SF and HCB selection that was previously proposed from our lab. Note that nopred-Trel and stdLTP-Trel are very close in their RD performance, mirroring the comparison between nopred-TLS and stdLTP-TLS. This reinforces the argument that the existing approach for selection of LTP parameter values is seriously flawed. The encoder with the proposed LTP

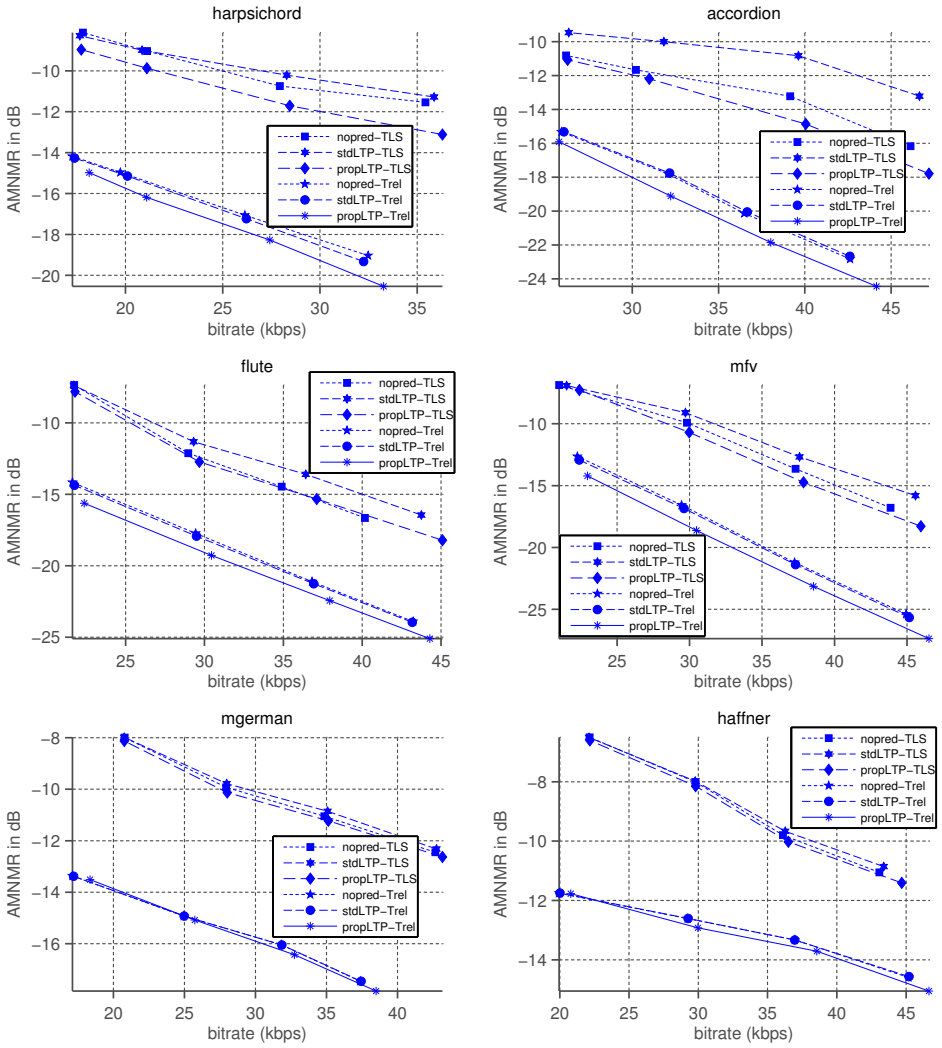

Figure 3.1: Average per-frame distortion at various bit-rates due to the different coders evaluated. Each graph corresponds to one audio sample in the dataset.

modifications, propLTP-Trel, consistently performs better than nopred-Trel and stdLTP-Trel (by about 1dB at different rates), and in particular for all samples with a steady periodic component (monophonic or polyphonic with a dominant instrument). This clearly demonstrates that the right optimization technique is key to the utility of the LTP tool.

It is noted that in case of the complex polyphonic file, which does not have a single dominant periodic component, no improvements are seen at all. This is due to a known limitation of the LTP technique: a good match with prior reconstructed data can be found only when there is a definite periodicity, i.e., only a single periodic component is present in the signal. Note that an effective solution for this major drawback is provided in Chapter 4.

Note that the LTP tool provides almost no improvements for the speech sample, which generally have a a single periodic component in the voiced segments, and one would expect coding improvements due to LTP in such sections. This behavior is due to the fact that there are small pitch period variations over time, and the time domain waveform matching inherent in LTP becomes inefficient. An effective solution for this shortcoming of LTP is provided in Chapter 5.

We also note that the low-complexity approach propLTP-TLS, that includes the proposed LTP modifications, performs better than nopred-TLS, again by about 1 dB at different rates.

3.3.2 Subjective evaluation results

A subjective evaluation of the proposed technique was conducted via MUSHRA listening tests [23]. All codecs under test were operated at 32kbps. The tests were conducted with 12 listeners and test items were scored on a scale of 0 (bad) to 100 (excellent).

Two separate tests were conducted. The first test compares the subjective quality of samples encoded by nopred-TLS, a widely used AAC-LD encoder, and by the proposed approach, propLTP-Trel. Listeners are provided with randomly ordered 4 different versions of each audio sample: a hidden reference (ref), a 3.5 kHz low-pass filtered anchor (anc), and samples encoded by nopred-TLS and propLTP-Trel encoders. The results of this test (the average MUSHRA scores and the 95% confidence intervals) for the different audio samples are shown in Fig. 3.2. The subjective evaluation results concur with the previously discussed objective evaluation results, and corroborate the fact that the proposed approach, propLTP-Trel, provides substantial improvements over the reference nopred-TLS coder.

A second test similarly compares the trellis-based encoder without LTP, nopred-Trel, against the overall optimization approach, propLTP-Trel. The results are provided in Fig. 3.3. This test indicates that the LTP tool can result in substantial quality improvements provided it is rightly optimized. As substantiated by

the objective results, the performance of stdLTP-Trel is almost the same as that of nopred-Trel. Thus, this second MUSHRA test also provides a measure of the subjective improvements solely due to the proposed LTP modifications (i.e., due to retaining multiple lag-gain survivors and embedding the per-SFB LTP flag selection into the trellis). Note that the subjective results are again in agreement with the objective distortion measurements previously discussed. Substantial gains in the average MUSHRA score are obtained for the samples with a dominant single periodic component, while the choice of the better codec is ambiguous for the complex polyphonic and speech samples.

3.4 Conclusion

This chapter demonstrates a novel perceptual distortion-rate optimization algorithm for LTP parameter value selection in MPEG AAC. Contrary to the current encoder that selects the LTP parameters via minimization of a mean squared error criterion, and decouples it from the selection of quantization and coding parameters, the proposed method jointly selects all the AAC parameters through a trellis-based search. The proposed algorithm retains a number of prediction survivors corresponding to different LTP lag-gain combinations, and evaluates them completely in terms of their perceptual distortion-rate performance. A low-

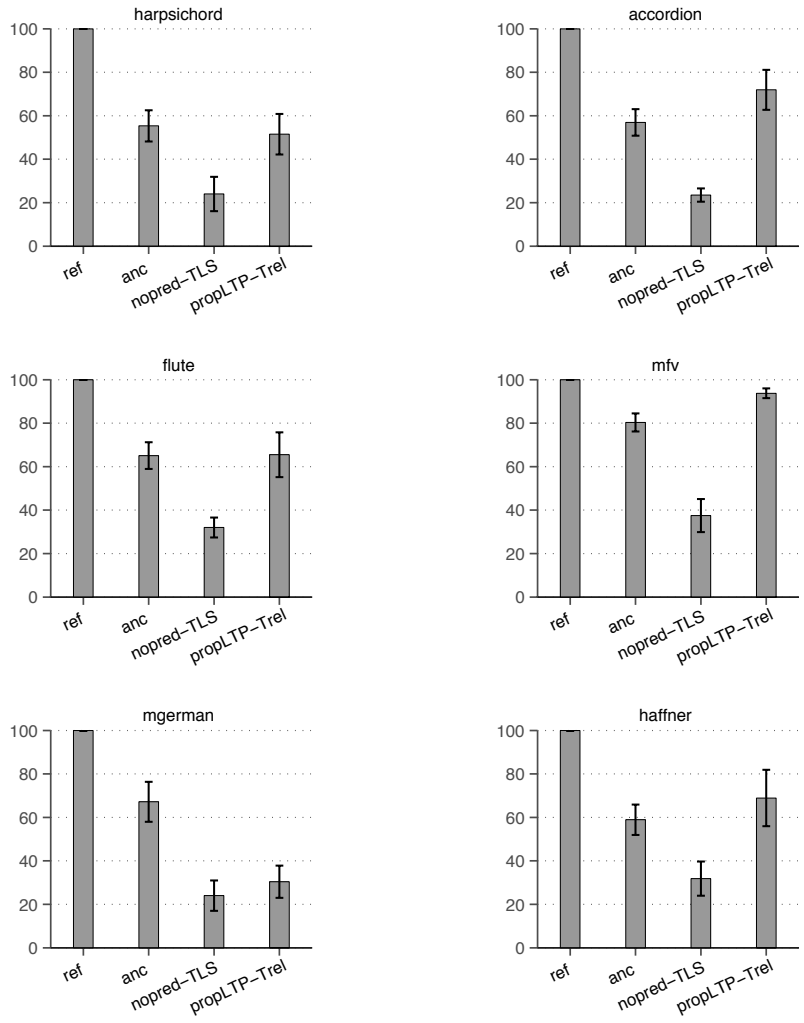

Figure 3.2: MUSHRA listening test comparing the standard approach and proposed techniques

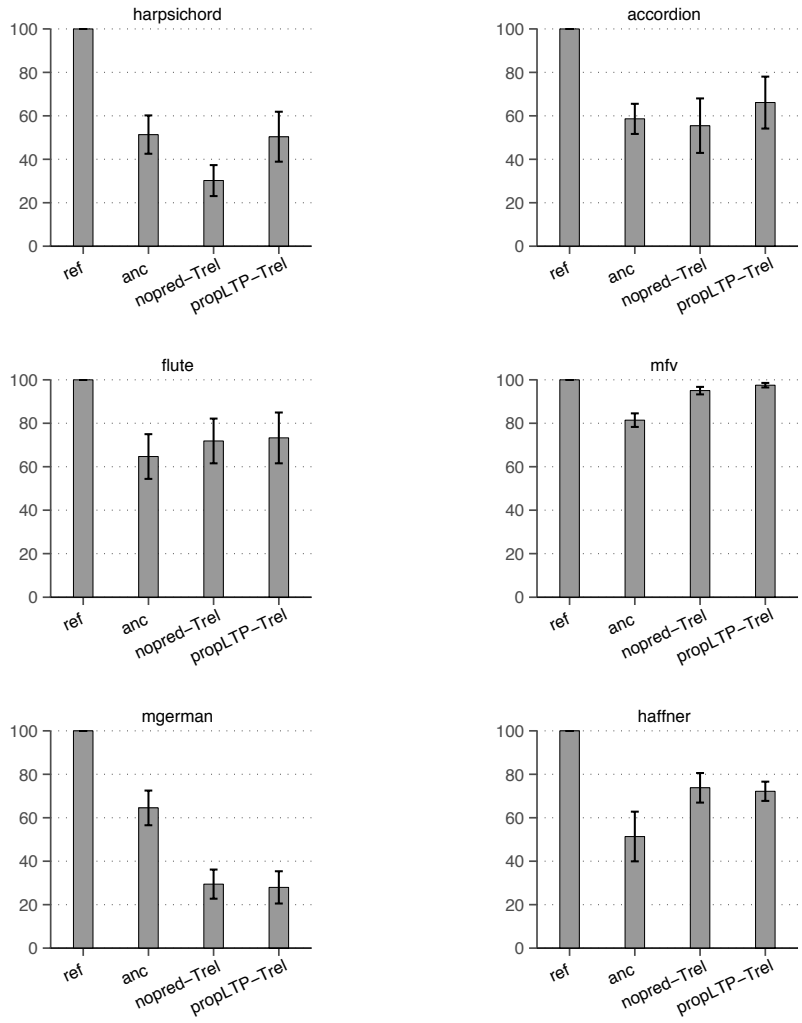

Figure 3.3: MUSHRA listening test to determine the gains due to a properly optimized LTP

complexity variant employs a two-loop search in place of the trellis. Finally the objective and subjective evaluations indicate the substantial quality improvements provided by the proposed techniques. The observations indicate that LTP could be a potent tool provided that the corresponding parameters are selected via the right optimization technique, and with consideration to the perceptual impacts of the choice of these parameters.

Chapter 4

Cascaded long term prediction for efficient compression of polyphonic audio signals

4.1 Introduction

The MPEG AAC standard (described in Section 2.1) along with the LTP tool (introduced in Section 2.3) that is perceptually optimized as described in Chapter 3 is well suited for signals containing a single periodic component, but general audio often contains a mixture of multiple periodic signals. Typically audio belongs to the class of polyphonic signals which includes as common examples, vocals with background music, orchestra, and chorus. Note that a single instrument may also produce multiple periodic components, as is the case for the piano or the guitar. In principle, the mixture is itself periodic albeit with overall period equaling the least common multiple (LCM) of all individual component periods, but the signal

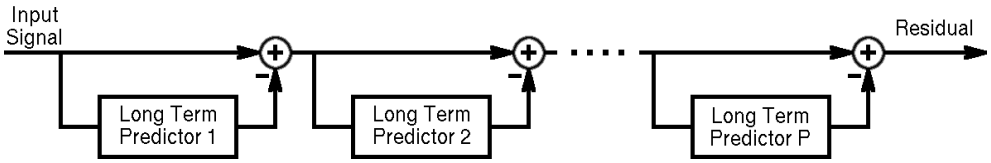

Figure 4.1: Illustration of a "Cascaded Long Term Prediction" (CLTP) filter.

rarely remains stationary over such extended duration. Consequently, LTP resorts to a compromise by predicting from a recent segment that represents some trade-off between incompatible component periods, with corresponding negative impact on its performance. This performance degradation of the LTP tool in MPEG AAC was clearly demonstrated in Chapter 3, where even perceptually motivated optimization did not yield noticeable performance improvement for polyphonic signals. Nevertheless, it is the premise of this work that, if exploited properly, the redundancies implicit in the periodic components of the signal offer a significant potential for compression gains. Similar to underlying principles used in spectral estimation for sum of sinusoids [24], we propose cascading LTP filters, each corresponding to individual periodic components of the signal, to form the overall *"cascaded long term prediction"* (CLTP) filter (as illustrated in Fig. 4.1). This construct enables predicting every periodic component in the current frame from the most recent previously reconstructed segment, with which it is maximally correlated. Moreover, the overall filter now requires only a limited history.

Given the CLTP construct, it is obvious that its efficacy is critically dependent on an effective parameter estimation technique, and even more so for coders such as MPEG AAC, where perceptual distortion criteria must be taken into account. We first propose, as a basic platform, prediction parameter optimization which targets mean squared error (MSE). It is then adapted to specific coders and their distortion criteria (e.g., the perceptual distortion criteria of MPEG AAC). To estimate such prediction parameters at acceptable complexity, while approaching optimality, we propose a "divide and conquer" recursive technique. That is, we find optimal parameters of an individual filter in the cascade, while fixing all other filter parameters. This process is then iterated for all filters in a loop, until convergence, to obtain the parameters of all LTP filters in the cascade. For the Bluetooth SBC [7,8], which uses a simple quantization MSE distortion, we employ this technique in a backward adaptive way, to minimize the side information rate, as the decoder can mimic this procedure. Backward adaptive estimation assumes local stationarity of the signal. For the MPEG AAC, we estimate the parameters in two stages, where we first employ the backward adaptive method to estimate a large subset of prediction parameters. Then these parameters are further fine tuned, with respect to the perceptual criteria, for the current frame and only refinement parameters are sent as side information. Specifically, the CLTP periods along with preliminary gains are estimated backward adaptively, to minimize the

prediction MSE. Note that the period estimation uses the MSE measure, as it is completely characterized by the signal waveform, regardless of perceptual considerations. The flags to enable prediction selectively in frequency bands are also estimated backward adaptively (unlike standard LTP) based on whether such prediction would reduce energy in a given band in the previous frame. Next, we note that gains in each filter are affected by the perceptual criteria, and they need to be adapted to the perceptually significant harmonics of the periodic component. We thus adjust the preliminarily obtained filter gains for each periodic component via quantized multiplicative coefficients, or factors, which are sent as side information. The number of periodic components in the CLTP filter is also specified as part of the side information. To optimize the perceptual effect of these refinement factors, a two-phase procedure similar to the one introduced in Chapter 3 is proposed. First, MSE is calculated for each combination of the factors for different number of periodic components, to retain S "survivor" combinations with the least mean squared prediction error. These S survivors then compete in terms of rate versus perceptual distortion performance via a two-loop search (TLS) based procedure [1, 6], which identifies the one that minimizes the perceptual distortion at the given rate. We also propose a low decoder complexity variant for the MPEG AAC, where all the parameters are sent as side information to the decoder, i.e., it is a forward adaptive coder. For this coder, we first estimate lags and preliminary

gains of the CLTP filter in an open-loop way to minimize MSE using the original input samples. The gains are now adjusted for closed loop prediction and also for taking into account the perceptual distortion criteria by introducing a multiplicative factor, and retaining S "survivors" with the least mean squared closed loop prediction error, again similar to the technique introduced in Chapter 3. The per band prediction activation flag is now estimated for each of these survivors in a forward adaptive way. Prediction side information of, number of lags, lags, gains, and per band flags, for each of the survivors are then encoded after taking into account the inter-frame dependency of parameters, to calculate the side information rate. Finally, the S survivors are evaluated via TLS based technique to identify the parameters which achieve the lowest perceptual distortion performance for a given rate. Performance gains of the proposed technique, assessed via objective and subjective evaluations for all the settings, demonstrates its effectiveness on a wide range of polyphonic signals.

Preliminary results of this approach for Bluetooth SBC, where CLTP is performed only on its first subband, have appeared in [25]. Early work of extending this approach to MPEG AAC, without the low decoder complexity variant has appeared in [26]. Historically, LTP has been considered since the introduction of predictive coding for speech [10]. A brief review of this LTP related prior work can be found in Section 2.2. Deeper consideration of CLTP points out relation

to special cases of the source-separation problem, and surveying literature in this area revealed a similar construct employed to mixed speech sources [27]. In [28], a higher order sparse linear predictor is proposed as an alternative approach for predicting polyphonic signals. This approach is based on the fact that cascade of multiple LTP filters and a short term linear predictor form a higher order sparse linear predictor. However, they do not demonstrate effectiveness for compression of real polyphonic signals, where accounting all the unstructured filter coefficients as side information would be a big hurdle.

This chapter is structured as follows: The polyphonic signal prediction problem is formulated in Section 4.2. The proposed CLTP technique is introduced in Section 4.3. The proposed recursive CLTP parameter estimation technique is described in Section 4.4. Specialization and derivations for enhancing Bluetooth SBC and MPEG AAC are presented in Section 4.5. Results are presented in Section 4.6, and the chapter concludes in Section 4.7.

Note that although this chapter will specify how to incorporate our proposed technique into the MPEG AAC and the Bluetooth SBC standards, the underlying approach is general and can easily be extended to other audio coders.

4.2 Polyphonic Signals and Problem Setting

This section sets up a characterization for polyphonic signals and identifies the corresponding major shortcoming of existing LTP filters.

The characterization of a simple periodic signal with period N is the relation $x[m] = x[m - N]$. But it is more realistic to assume that a naturally occurring periodic signal is not perfectly stationary and has a non-integral period, i.e.,

$$x[m] = \alpha x[m - N] + \beta x[m - N + 1], \tag{4.1}$$

where α and β capture both the amplitude changes and approximate the non-integral pitch period via linear interpolation. A polyphonic audio signal comprising a mixture of such periodic signals, can be modeled as

$$x[m] = \sum_{i=0}^{P-1} x_i[m] + w[m], \tag{4.2}$$

where P is the number of periodic components, $w[m]$ is a noise sequence, and $x_i[m]$ are periodic signals satisfying,

$$x_i[m] = \alpha_i x_i[m - N_i] + \beta_i x_i[m - N_i + 1]. \tag{4.3}$$

The prediction problem at hand is to find a filter of the form $H(z) = 1 - \sum_{k>0} a_k z^{-k}$ such that the prediction error $E(z) = S(z)H(z)$ is of minimum energy. If the signal has a single periodic component ($P = 1$), then we have an

obvious choice for the LTP filter:

$$H_0(z) = 1 - \alpha_0 z^{-N_0} - \beta_0 z^{-N_0+1}, \tag{4.4}$$

whose prediction error $e[m]$ is dependent only on the driving noise or innovation $w[m]$. An illustration of simple LTP filtering is provided in Fig. 4.2(a) for an example periodic signal (absent noise). The LTP tool in MPEG AAC standard (described in Section 2.3) can also be effective in this case by selecting a lag close to a multiple of the period in the range $\{K, \ldots, 3K-1\}$ and appropriately adapting the other parameters for the best prediction results.

For signals with multiple periodic components, i.e., $P > 1$, the LTP filter, with a single degree of freedom for lag, can only be a "compromise" solution

$$H_{\text{ltp}}(z) = 1 - \alpha_{\text{ltp}} z^{-N_{\text{ltp}}} - \beta_{\text{ltp}} z^{-N_{\text{ltp}}+1}, \tag{4.5}$$

where N_{ltp} is the lag that minimizes the mean squared prediction error, within the history available for prediction. Similarly, even the LTP tool in MPEG AAC standard, simply selects a compromise lag that minimizes the mean squared prediction error in the range $\{K, \ldots, 3K-1\}$. Theoretically, the lag selected should approximate the integer LCM of the individual periods, but in practice, as discussed earlier, it is suboptimal for real polyphonic signals as they do not remain stationary over a long duration. If the LCM falls beyond the available history, then the lag selected will clearly be a compromise seeking the best match possible

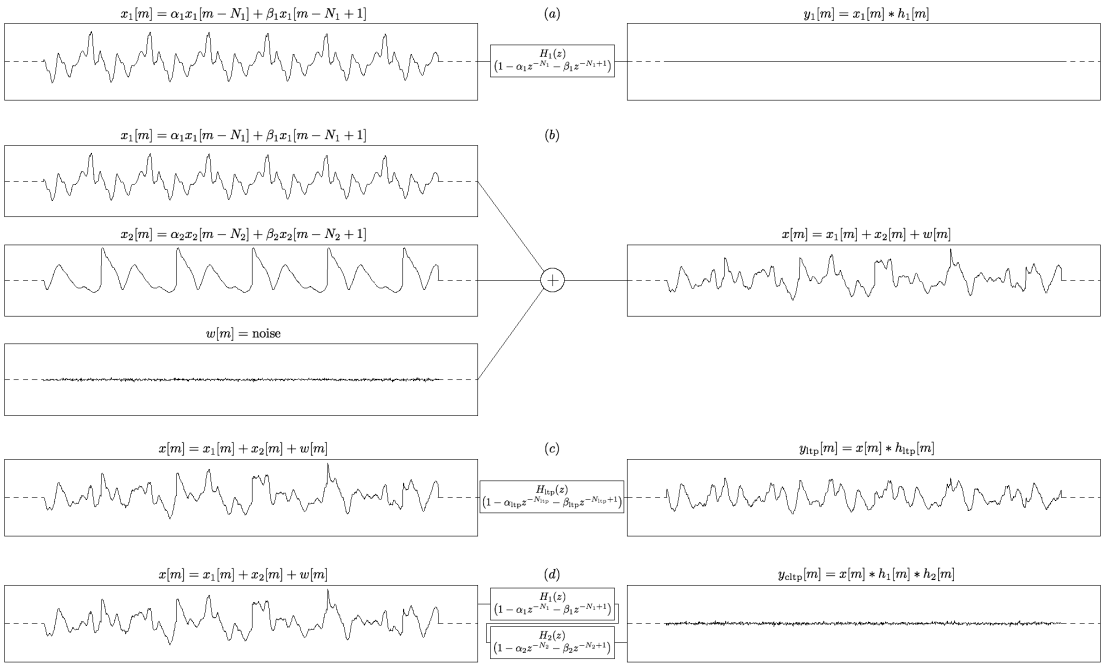

Figure 4.2: Illustration of various LTP filters. a) Output of simple LTP filtering for a periodic signal. b) A realistic polyphonic signal with 2 periodic components and noise. c) Output of simple LTP filtering, which minimizes MSE, for the polyphonic signal of (b). d) Output of cascaded LTP filtering for the polyphonic signal of (b).

despite the incompatible periods. The suboptimality of simple LTP filtering a polyphonic signal is illustrated in Fig. 4.2(c). Note that this limitation is due to the overly simplistic prediction model of LTP, and Chapter 3 confirmed that perceptually motivated parameter optimization of the LTP tool in MPEG AAC standard, while beneficial for monophonic signals, does not provide significant performance improvement for complex polyphonic signals.

4.3 Cascaded Long Term Prediction

If we apply the LTP filter $H_0(z)$ (in (4.4)) designed for a signal with single periodic component to a polyphonic signal (4.2) where $P > 1$, the filtered output is

$$\begin{aligned} e_0[m] &= x[m] - \alpha_0 x[m - N_0] - \beta_0 x[m - N_0 + 1] \\ &= \sum_{i=0}^{P-1} x_i'[m] + w'[m], \end{aligned} \tag{4.6}$$

where $x_i'[m] = x_i[m] - \alpha_0 x_i[m - N_0] - \beta_0 x_i[m - N_0 + 1]$ is the filtered version of the ith periodic component, and $w'[m] = w[m] - \alpha_0 w[m - N_0] - \beta_0 w[m - N_0 + 1]$ is the filtered noise. Designing filter H_0 for the periodic component $x_0[m]$ guarantees that $x_0'[m] = 0$. Moreover, it can be verified with straightforward algebra that all the remaining components, $x_i'[m]$, exhibit the same periodicity as $x_i[m]$, i.e.,

$$x_i'[m] = \alpha_i x_i'[m - N_i] + \beta_i x_i'[m - N_i + 1]. \tag{4.7}$$

In other words, the output of filter H_0 is, in fact, a polyphonic signal with $P - 1$ periodic components:

$$e_0[m] = \sum_{i=1}^{P-1} x_i'[m] + w'[m]. \tag{4.8}$$

It follows recursively that the cascaded LTP filter

$$H_c(z) = \prod_{i=0}^{P-1} (1 - \alpha_i z^{-N_i} - \beta_i z^{-N_i+1}) \tag{4.9}$$

will cancel all the periodic components and leave a prediction error dependent only on $w[m]$. An illustration of cascaded LTP filtering a polyphonic signal, which successfully cancels out all periodic components, is provided in Fig. 4.2(d). The CLTP filter of (4.9), appropriately designed, forms the basis of our proposal in this chapter to improve compression efficiency of polyphonic audio signals.

4.4 Basic CLTP Parameter Estimation

Estimation of CLTP filter parameter values, specifically adapted for the distortion criteria of an audio coder, is crucial to the effectiveness of this technique with real polyphonic signals. However, as a starting point to solve this problem, we first derive in this section a minimum mean squared prediction error technique to optimize the CLTP parameter set: $N_i, \alpha_i, \beta_i \; \forall i \in \{0, \ldots, P-1\}$. A straightforward exhaustive approach would be to evaluate all combinations from a predefined set of values to find the one that minimizes the prediction error. This can be done by first fixing the range of pitch periods to Q possibilities, then finding the best α_i, β_i for each of the Q^P period combination and finally selecting the period combination that minimizes the mean squared prediction error. Clearly, the complexity of this approach grows exponentially with number of periodic components. For the modest choice of $Q = 100$ and $P = 5$, there are $Q^P = 10^{10}$ combinations to

be evaluated every time the parameters need an update, resulting in prohibitive computational complexity. Thus we propose a "divide and conquer" recursive estimation technique.

For a given P, to estimate parameters of the jth filter, N_j, α_j, β_j, we fix all other filters and define the partial filter

$$H_j(z) = \prod_{\forall i, i \neq j} (1 - \alpha_i z^{-N_i} - \beta_i z^{-N_i+1}) \tag{4.10}$$

and the corresponding residue

$$X_j(z) = X(z)H_j(z). \tag{4.11}$$

We next optimize the parameters of the filter $(H_c(z)/H_j(z)) = 1 - \alpha_j z^{-N_j} - \beta_j z^{-N_j+1}$ for the residue $x_j[m]$. This boils down to the classic LTP problem, where for a given N the $\alpha_{(j,N)}, \beta_{(j,N)}$ are given by

$$\begin{bmatrix} \alpha_{(j,N)} \\ \beta_{(j,N)} \end{bmatrix} = \begin{bmatrix} r_{(N,N)} & r_{(N-1,N)} \\ r_{(N-1,N)} & r_{(N-1,N-1)} \end{bmatrix}^{-1} \begin{bmatrix} r_{(0,N)} \\ r_{(0,N-1)} \end{bmatrix} \tag{4.12}$$

where the correlation values $r_{(k,l)}$ are

$$r_{(k,l)} = \sum_{m=Y_{\text{start}}}^{Y_{\text{end}}} x_j[m-k]x_j[m-l], \tag{4.13}$$

where, $Y_{\text{start}}, Y_{\text{end}}$ are the limits of summations that depend on the length of the available history and the length of the current frame. To ensure stability of the

synthesis filter used in prediction we restrict $\alpha_{(j,N)}, \beta_{(j,N)}$ solutions to only those that satisfy the sufficient stability criteria of

$$|\alpha_{(j,N)}| + |\beta_{(j,N)}| \leq 1. \tag{4.14}$$

For details on estimating parameters which satisfy the sufficient stability criteria, please refer to Appendix A. Given $\alpha_{(j,N)}, \beta_{(j,N)}$, the optimal N_j is found as

$$N_j = \underset{N \in [N_{\min}, N_{\max}]}{\arg\min} \sum_{m=Y_{\text{start}}}^{Y_{\text{end}}} \begin{aligned}(x_j[m] &- \alpha_{(j,N)} x_j[m-N] \\ &- \beta_{(j,N)} x_j[m-N+1])^2,\end{aligned} \tag{4.15}$$

where $N_{\min}, N_{\max}$ are the lower and upper boundaries of the period search range. In equations (4.13) and (4.15), the signal can be replaced with reconstructed samples $\hat{x}[m]$ for backward adaptive parameter estimation. The process above is now iterated over the component filters of the cascade, until convergence. Convergence is guaranteed as the overall prediction error is monotonically non-increasing at every step of the iteration. Note that as the overall cost is clearly non-convex in the pitch periods N_j, a globally optimal solution is not guaranteed.

4.5 Enhancing real world codecs with CLTP

This section describes the adaptation of CLTP to the real world codecs of Bluetooth SBC and MPEG AAC.

4.5.1 CLTP for coders operating on frames

Closed-loop prediction is needed, where all samples of the current frame are predicted from previously reconstructed samples, in order to avoid error propagation and decoder drift. If the frame length is longer than the minimum pitch period, employing the CLTP (or the LTP) analysis filter as is, would utilize samples that have not yet been encoded. To address this problem, we employ an approach known as 'looped prediction'. Given the frame length, K, and number of samples available as history, M, we first formulate a prediction filter input $\hat{x}'[m]$ for every frame n, out of M previously reconstructed samples $\hat{x}[m]$ padded with zeros, specifically $\hat{x}'[m] = \hat{x}[m]$ for $-M \leq m \leq -1$ and $\hat{x}'[m] = 0$ for $0 \leq m < K$. Then the CLTP synthesis filter $1/H_c(z)$ is run through $\hat{x}'[m]$ for $0 \leq m < K$ and the resulting samples form the predicted samples $\tilde{x}_n[m],\ 0 \leq m < K$. This basically is synthesizing predicted samples while assuming prediction residue is 0 and the previously reconstructed samples as the initial state. If $P = 1$, this approach is simply repeating an appropriately scaled pitch period number of the latest reconstructed samples, so as to generate the entire frame's prediction. Even for $P > 1$ this approach effectively predicts every periodic component from its immediate history.

4.5.2 Integration with Bluetooth SBC

The Bluetooth SBC (described in Section 2.4) is clearly limited in its capability to exploit redundancies due to short block length. Thus CLTP can improve its compression efficiency by providing effective inter-frame prediction, *without increasing delay*. Also the basic CLTP parameter estimation technique described in Section 4.4 is well matched with the quantizer in SBC, as they both minimize MSE. In order to encode the samples of the n'th frame: $x[m],\ nBK \leq m < (n+1)BK$, we maintain a history of $M = 2048$ reconstructed samples: $\hat{x}[m],\ nBK - (9B+1) - M \leq m < nBK - (9B+1)$. Note that there is gap of $(9B+1)$ samples between the last reconstructed sample and the first sample of the current frame, due to the delay introduced by the analysis and synthesis filter banks. We employ CLTP to predict samples of the current frame along with the samples required for the analysis filter bank history, i.e., CLTP is employed to generate predicted samples $\tilde{x}[m],\ nBK - 9B \leq m < (n+1)BK$. The CLTP filter is of the following form

$$H_n(z) = \prod_{i=0}^{P_n-1} (1 - \alpha_{(i,n)} z^{-N_{(i,n)}} - \beta_{(i,n)} z^{-N_{(i,n)}+1}), \tag{4.16}$$

and its parameters are estimated backward adaptively, once per frame. For a tentative value of the number of periodic components, P_n, the parameters $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)},\ \forall i \in \{0, \ldots, P_n - 1\}$ are estimated backward adaptively via

the recursive technique described in Section 4.4, with the limits, $Y_{\text{start}} = nBK - (9B + 1) - M/4$, $Y_{\text{end}} = nBK - (9B + 1) - 1$, in the correlation and prediction error measures (4.13), (4.15) and using the reconstructed samples, $\hat{x}[m]$. This process is then repeated to find CLTP filters for each $P_n \in \{1, \ldots, P_{\max}\}$ and the one which minimizes the mean squared prediction error is selected. The predicted samples required to calculate this error is generated via the 'looped' prediction method described in Section 4.5.1. For the selected P_n, the predicted samples $\tilde{x}[m],\ nBK - 9B \leq m < (n+1)BK$ are now mapped into subbands to generate predicted subband samples of the current frame n, $\tilde{c}_n[b,k],\ 0 \leq b < B,\ 0 \leq k < K$. The prediction residue is calculated as $e_n[b,k] = c_n[b,k] - \tilde{c}_n[b,k]$. A per subband one bit flag, $f_n[b]$, is used to selectively enable CLTP, and this flag is set whenever the prediction residue energy is lower than the signal energy in the band:

$$f_n[b] = \begin{cases} 1, & \text{if } \sum_{k=0}^{K-1} e_n^2[b,k] < \sum_{k=0}^{K-1} c_n^2[b,k] \\ 0, & \text{otherwise.} \end{cases} \tag{4.17}$$

The actual input to the quantization module, $\forall k$ in each subband b, is denoted as,

$$q_n[b,k] = \begin{cases} e_n[b,k], & \text{if } f_n[b] = 1, \\ c_n[b,k], & \text{otherwise.} \end{cases} \tag{4.18}$$

These samples are now quantized adaptively in each block and sent to the decoder, along with the side information of the quantization step sizes $s_n[b]$, the number of

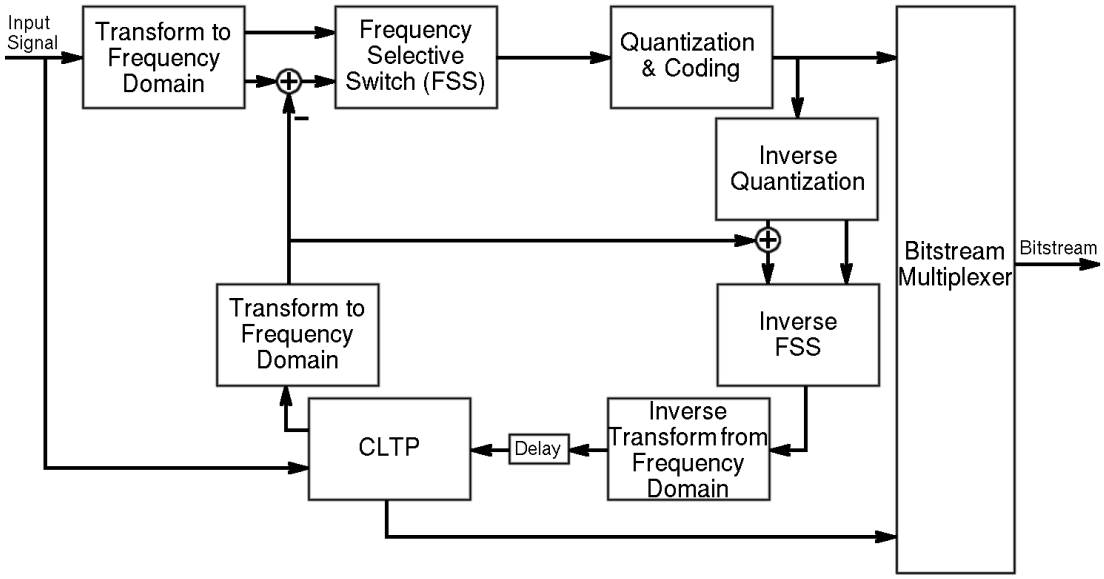

Figure 4.3: Illustration of the proposed integration of CLTP with an audio coder operating in frequency domain.

periodic components P_n, and the flags $f_n[b]$. An illustration of integrating CLTP with the Bluetooth SBC encoder is provided in Fig. 4.3 (wherein transform to frequency domain and inverse transform from frequency domain corresponds to the subband analysis and synthesis filters for Bluetooth SBC).

The decoder receives P_n and estimates $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}$, $\forall i \in \{0, \ldots, P_n - 1\}$ to generate the predicted sub-band samples. The subband samples received in the bitstream are dequantized and added to the predicted subband samples whenever the flag $f_n[b]$ is set, to generate the reconstructed sub-band samples, from which the output signal is synthesized. The recursive technique's speed of

convergence is improved by employing prediction parameters from the previous frame as initialization for the procedure.

4.5.3 Integration with MPEG AAC

The efficacy of CLTP filters in enhancing MPEG AAC is critically dependent on parameter estimation accounting for the criteria of minimizing perceptual distortion at a given rate. We propose to tackle this problem in two stages, where in the first stage we estimate a large subset of prediction parameters backward adaptively to minimize the side information rate, then in the subsequent stage these parameters are "fine tuned" for the current frame, with respect to the perceptual criteria, and only refinement parameters are sent as side information. Note that in estimating parameters backward adaptively we exploit the assumed local stationarity of the signal.

Estimation of Backward Adaptive Parameters

For a tentative number of periodic components P_n in frame n, we estimate the pitch periods and preliminary gains $(N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)} \ \forall i \in 0, \ldots, P_n - 1)$ backward adaptively from previously reconstructed samples $\hat{x}[m]$, $(n-5)K \leq$

$m < nK$, to form the following CLTP filter

$$\bar{H}_n(z) = \prod_{i=0}^{P_n-1} (1 - \alpha_{(i,n)} z^{-N_{(i,n)}} - \beta_{(i,n)} z^{-N_{(i,n)}+1}). \tag{4.19}$$

Note that the pitch period is a physical property of the signal waveform, and independent of perceptual considerations. We hence minimize the prediction MSE directly in this stage. The recursive technique described in Section 4.4 is employed with the limits, $Y_{\text{start}} = (n-2)K$, $Y_{\text{end}} = nK-1$, in the correlation and prediction error measures (4.13), (4.15) and uses the reconstructed samples, $\hat{x}[m]$. In the next step, we retain the flexibility to selectively enable prediction in SFBs, similar to the practice in the MPEG AAC LTP tool. But unlike standard LTP, which specifies the corresponding flags as side information, we backward adaptively estimate them from previously reconstructed samples $\hat{x}[m]$. Given the tentative number of periodic components P_n and the backward adaptively estimated preliminary CLTP filter $\bar{H}_n(z)$, we generate the prediction residue samples by filtering the reconstructed samples $\hat{x}[m]$ with $\bar{H}_n(z)$. Then we transform the last 2K residue samples (which correspond to frame $(n-2)$) via MDCT to generate the residual transform coefficients $e_{n-2}[k],\ 0 \leq k < K$. This is now compared to the $(n-2)$ frame's reconstructed MDCT coefficients $\hat{c}_{n-2}[k],\ 0 \leq k < K$ and its re-estimated masking thresholds $\hat{\mu}_{n-2}[l],\ 0 \leq l < L$ to decide the per-SFB prediction enable

flag $f_n[l]$, as

$$f_n[l] = \begin{cases} 1, & \text{if } \sum\limits_{k \in \text{SFB } l} \hat{c}^2_{n-2}[k] > \hat{\mu}_{n-2}[l] \text{ and} \\ & \sum\limits_{k \in \text{SFB } l} e^2_{n-2}[k] < \sum\limits_{k \in \text{SFB } l} \hat{c}^2_{n-2}[k] \\ 0, & \text{otherwise.} \end{cases} \tag{4.20}$$

Thus, prediction in an SFB is enabled if its signal energy is higher than the masking threshold and the prediction error is of lower energy than the original signal.

Perceptually Motivated Joint CLTP Parameter Refinement and core AAC Parameter Estimation

The gain factors $\alpha_{(i,n)}, \beta_{(i,n)}$ for each periodic component i are naturally affected by the perceptual distortion criteria, i.e., they should be adapted according to the perceptual significance of the harmonics. We thus introduce a corrective gain factor $\mathbf{G}_{(i,n)}$ to form the final CLTP filter

$$H_n(z) = \prod_{i=0}^{P_n - 1} (1 - \mathbf{G}_{(i,n)} \alpha_{(i,n)} z^{-N_{(i,n)}} - \mathbf{G}_{(i,n)} \beta_{(i,n)} z^{-N_{(i,n)}+1}), \tag{4.21}$$

where $\mathbf{G}_{(i,n)}$ is quantized to one of $\mathbf{N_G}$ levels, e.g., $\{0.5, 0.75, 1, 1.25\}$. We next restrict the range of P_n to $\{1, \ldots, P_{\max}\}$ and also retain the global flag $\mathbf{F}_n$ to enable/disable CLTP on a per-frame basis. Note that P_n is sent to the decoder

using $\lceil \log_2(P_{\max}) \rceil$ bits and $\mathbf{G}_{(i,n)}$ $\forall i$, are sent to the decoder using $\lceil \log_2(\mathbf{N_G}) \rceil P_n$ bits.

A straightforward way to estimate all the remaining parameters would be to evaluate for every combination of CLTP parameters $P_n, \mathbf{G}_{(i,n)}$ and $\mathbf{F}_n$, the perceptual distortion minimizing AAC quantization and coding parameters for the given rate, and select the combination that minimizes perceptual distortion. But even for a modest $P_{\max} = 5$ and $\mathbf{N_G} = 4$, we need to evaluate $4^5 + 1 = 1025$ combinations for RD performance, which considerably exacerbates the computational complexity. We thus adopt a parameter estimation technique to eliminate most non-competitive contenders, similar to the technique we proposed in Chapter 3 for the MPEG AAC LTP tool. Firstly, all the CLTP parameter combinations are evaluated with respect to plain MSE to retain only the top S survivors. Then, the S survivors are evaluated for RD performance via TLS based technique to identify the one that minimizes the perceptual distortion at the given rate. The overall algorithm can be summarized as follows:

1. For every $P_n \in \{1, \ldots, P_{\max}\}$ the preliminary CLTP filter $\bar{H}_n(z)$ and the per-SFB flags $f_n[l]$ are estimated as described above.

2. Given $\bar{H}_n(z)$, for every possible combination of $\mathbf{G}_{(i,n)}$ the predicted samples $\tilde{x}_n[m],\ 0 \leq m < 2K$ are generated using the synthesis filter $1/H_n(z)$ via

the 'looped' prediction method described in Section 4.5.1. These samples are then transformed via MDCT to produce K transform coefficients $\tilde{c}_n[k]$. The per transform coefficient prediction residue is now calculated as $e_n[k] = c_n[k] - \tilde{c}_n[k]$. Finally, prediction MSE, after considering the flags $f_n[l]$, is evaluated.

3. Top S survivors are determined based on this prediction MSE.

4. Given the CLTP parameters of each survivor, the corresponding SFs and HCBs are determined via TLS, and the associated distortion calculated for the given total rate (which includes the CLTP side information rate).

5. TLS is also employed for the case when CLTP is disabled for the frame (i.e., $\mathbf{F}_n = 0$), to calculate the associated distortion for the given total rate.

6. Amongst the $S + 1$ cases, the parameters that correspond to the minimum distortion are employed in encoding the frame.

Note that controlling the number of survivors S enables controlling the tradeoff between complexity and performance. An overall illustration of the proposed integration of CLTP with the MPEG AAC encoder is provided in Fig. 4.3, with the transform to frequency domain and inverse transform from frequency domain corresponding to MDCT and IMDCT for MPEG AAC.

The decoder receives $\mathbf{F}_n$, and if $\mathbf{F}_n$ is set it also receives P_n, and $\mathbf{G}_{(i,n)}$, $\forall i$. If $\mathbf{F}_n$ is set, the decoder estimates $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}$, $\forall i$, and the per-SFB prediction activation flag $f_n[l]$, backward adaptively. Given these parameters it then generates the predicted samples $\tilde{x}_n[m],\ 0 \leq m < 2K$ using the synthesis filter $1/H_n(z)$ via the 'looped' prediction method described in Section 4.5.1. These samples are then transformed via MDCT to produce K transform coefficients $\tilde{c}_n[k]$. The transform coefficients received in the core AAC bitstream are Huffman decoded, dequantized, and the predicted transform coefficients are added whenever the flag $f_n[l]$ is set, to generate the reconstructed transform coefficients, from which the output signal is synthesized via inverse MDCT. If $\mathbf{F}_n$ is not set, standard AAC decoding procedure is followed.

Low Decoder Complexity Variant

Clearly in a backward adaptive setting, decoder complexity increases significantly as it needs to replicate estimating parameters from previously reconstructed samples in a way identical to the encoder. While this technique keeps the side information rate minimal, some applications cannot afford the increase in decoder complexity. We thus introduce an alternative technique that employs forward adaptive parameter estimation to keep the decoder complexity in check, as in this technique the only additional step in the decoder is to synthesize the current

frame prediction using the filter parameters received as part of the side information. Note that in this approach we trade decoder complexity decrease for increase in side information rate. However, we employ parameter encoding techniques that explicitly account for inter-frame dependency of parameters to minimize the loss in overall RD performance of the coder. Details of the parameter estimation technique are described in this section, while details of the parameter encoding technique are described in Appendix C.

For a tentative number of periodic components P_n in frame n, we estimate the pitch periods and preliminary gains $(N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}\ \forall i \in 0, \ldots, P_n - 1)$ in an open loop way using original samples $x[m],\ (n-3)K \leq m < (n+2)K$, to form the following CLTP filter

$$\bar{H}_n(z) = \prod_{i=0}^{P_n-1} (1 - \alpha_{(i,n)} z^{-N_{(i,n)}} - \beta_{(i,n)} z^{-N_{(i,n)}+1}). \tag{4.22}$$

Note that as the parameter estimation is forward adaptive, we utilize the opportunity to use open loop parameter estimation, which results in better accuracy of parameters, as the original signal (uncorrupted by quantization error) is used. The recursive technique described in Section 4.4 is employed with the limits, $Y_{\text{start}} = nK$, $Y_{\text{end}} = (n+2)K - 1$, in the correlation and prediction error measures (4.13), (4.15) and uses the original samples, $x[m]$.

While the above step estimates the open loop prediction filter parameters, the actual predicted samples of the current frame are generated as described in

Section 4.5.1, from previously reconstructed samples, to avoid quantization error propagation. Hence CLTP gain factors need to be adjusted for closed loop prediction. Also as described in previous section, CLTP gain factors need to be adjusted according to the perceptual distortion criteria. Note that, the pitch period has no need to be updated for closed loop prediction or perceptual distortion criteria as it is physical property of the signal waveform. We tackle the highly non-convex problem of adjusting gain factors for closed loop prediction and perceptual distortion criteria by limiting our search to a small discrete set of neighborhood around the preliminary estimate of gain factors by introducing a multiplicative gain factor $\mathbf{G}_{(i,n)}$, which can take one of $\mathbf{N_G}$ levels, e.g., $\{0.5, 0.75, 1, 1.25\}$. The final gain factors $\mathbf{G}_{(i,n)}\alpha_{(i,n)}$, $\mathbf{G}_{(i,n)}\beta_{(i,n)}$ are then non-uniformly quantized to $\hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ for efficient encoding as side information. Please refer to Appendix B for a detailed description of this quantizer. The final CLTP filter is,

$$H_n(z) = \prod_{i=0}^{P_n-1} (1 - \hat{\alpha}_{(i,n)} z^{-N_{(i,n)}} - \hat{\beta}_{(i,n)} z^{-N_{(i,n)}+1}). \tag{4.23}$$

To find the best $\mathbf{G}_{(i,n)}$ and rest of the parameters the procedure described below is followed. For every combination of $\mathbf{G}_{(i,n)}$, the predicted samples $\tilde{x}_n[m],\ 0 \leq m < 2K$ are generated using the synthesis filter $1/H_n(z)$ via the 'looped' prediction method described in Section 4.5.1. These samples are then transformed via MDCT to produce K transform coefficients $\tilde{c}_n[k]$ and the prediction residue in transform domain is calculated as $e_n[k] = c_n[k] - \tilde{c}_n[k]$. The per-SFB energy of this prediction

residue is now used to calculate per-SFB prediction activation flag as,

$$f_n[l] = \begin{cases} 1, & \text{if } \sum\limits_{k \in \text{SFB } l} c_n^2[k] > \mu_n[l] \text{ and} \\ & \sum\limits_{k \in \text{SFB } l} e_n^2[k] < \sum\limits_{k \in \text{SFB } l} c_n^2[k] \\ 0, & \text{otherwise.} \end{cases} \tag{4.24}$$

Similar to the previous section, for a range of $P_n \in \{1, \ldots, P_{\max}\}$, we retain the top S survivors with respect to transform domain mean squared prediction error, after accounting for the per-SFB prediction activation flags. The parameter set of P_n, $N_{(i,n)}, \hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ $\forall i \in 0, \ldots, P_n - 1$, and $f_n[l]$ $\forall l$, are encoded for each survivor, as described in Appendix C, to calculate the rate for transmission as side information to the decoder. Finally, the S survivors are evaluated for RD performance via TLS based technique to identify the one that minimizes the perceptual distortion at the given rate. The overall algorithm can be summarized as follows:

1. For every $P_n \in \{1, \ldots, P_{\max}\}$ the preliminary CLTP filter $\bar{H}_n(z)$ is estimated in an open loop way.

2. Given $\bar{H}_n(z)$, for every possible combination of $\mathbf{G}_{(i,n)}$ the predicted samples are generated via the 'looped' prediction method described in Section 4.5.1. These samples are then transformed via MDCT and the prediction residue in transform domain is calculated. The per-SFB energy of this prediction

residue is now used to calculate $f_n[l]$ as given in (4.24). Finally, prediction MSE, after considering the flags $f_n[l]$, is evaluated.

3. Top S survivors are determined based on this prediction MSE and all the prediction parameters are encoded to calculate the side information rate.

4. Given the CLTP parameters of each survivor, the corresponding SFs and HCBs are determined via TLS, and the associated distortion calculated for the given total rate (which includes the CLTP side information rate).

5. TLS is also employed for the case when CLTP is disabled for the frame (i.e., $\mathbf{F}_n = 0$), to calculate the associated distortion for the given total rate.

6. Amongst the $S + 1$ cases, the parameters that correspond to the minimum distortion are employed in encoding the frame.

This variant decoder first receives $\mathbf{F}_n$ as the side information. If $\mathbf{F}_n$ is set, it also receives P_n, $N_{(i,n)}, \hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ $\forall i$, and $f_n[l]$ $\forall l$. The decoder then generates the predicted samples $\tilde{x}_n[m]$, $0 \leq m < 2K$ using the synthesis filter $1/H_n(z)$ via the 'looped' prediction method described in Section 4.5.1. These samples are then transformed via MDCT to produce K transform coefficients $\tilde{c}_n[k]$. The transform coefficients received in the core AAC bitstream are Huffman decoded, dequantized, and the predicted transform coefficients are added whenever the flag $f_n[l]$ is set, to generate the reconstructed transform coefficients, from which the

output signal is synthesized via inverse MDCT. If $\mathbf{F}_n$ is not set, standard AAC decoding procedure is followed.

4.6 Results

This section presents the results of experiments conducted with the proposed CLTP technique adapted for the Bluetooth SBC coder and the MPEG AAC coder. The experiments were conducted with single channel 44.1/48kHz audio sample subset from the standard MPEG and EBU SQAM database. We extracted a 4 seconds portion of each audio file for time efficient evaluation. The evaluated subset is:

- Single instrument multiple chords: Grand Piano, Guitar, Tubular Bells
- Orchestra: Mfv, Mozart
- Chorus: Vocal Quartet

4.6.1 Results for Bluetooth SBC

We compare the following coders in our experiments:

- Reference SBC with no prediction (referred to in figure as "NoLTP")
- SBC with one LTP filter (obtained by setting $P_{\max} = 1$)

- SBC with the proposed CLTP.

The SBC is operated at $B = 4$ and $K = 16$; and we restricted CLTP to $P_{\max} = 5$. The boundary points in equation (4.15) are $N_{\min} = 100$, $N_{\max} = 800$ for both LTP and CLTP. Thus, side information rate is 4 bits/block (2.8/3 kbps) for LTP (1 bit per subband prediction activation flag) and 7 bits/block (4.8/5.25 kbps) for CLTP (1 bit per subband prediction activation flag, 3 bits for P_n) and are included in the rate totals. Note that the SBC with one LTP filter is non-standard, but is included in our experiments to specifically demonstrate the performance improvements of using CLTP over LTP.

Objective evaluation results

As SBC encodes with the aim of minimizing signal to quantization noise ratio (SNR) (effectively the MSE criteria), we first evaluate SNR gains to measure our performance improvements. The prediction gains and the reconstruction gains, for LTP over no LTP, and for CLTP over LTP, at an operating point of around 80 kbps, for each of the six files, are given in Table 4.1. The table shows that CLTP provides truly major prediction gains of on the average 6.9 dB over LTP, which translate to substantial compression performance gains of on the average 6.1 dB. The table also shows that these gains came on top of already substantial

Filename	Prediction gains		Reconstruction gains	
	LTP over NoLTP	CLTP over LTP	LTP over NoLTP	CLTP over LTP
Piano	5.7	9.0	5.2	8.5
Guitar	10.4	4.6	7.4	3.1
Bells	5.0	9.9	4.7	9.2
Mfv	6.0	6.7	6.2	6.0
Mozart	7.7	7.0	6.7	6.0
Quartet	2.7	4.2	2.3	3.5
Average	6.3	6.9	5.4	6.1

Table 4.1: Prediction gains and reconstruction gains in dB for the Bluetooth SBC experiments

gains provided by LTP. We note also that the prediction gains are substantially but not fully translated into reconstruction gains.

We then evaluate SNR versus bit-rate to generate operational rate-distortion (RD) plots for each coder. RD plots averaged over files in each of the three classes of the test dataset, are shown in Fig. 4.4. The plots clearly demonstrate that substantial gains are provided by CLTP for a wide range of polyphonic signals at various rates.

Subjective evaluation results

A subjective evaluation of all the three competing Bluetooth SBC coders, operating at around 80kbps, was conducted via MUSHRA listening tests [23]. The tests were conducted with 16 listeners and test items were scored on a scale of 0 (bad) to 100 (excellent). Listeners were provided with randomly ordered

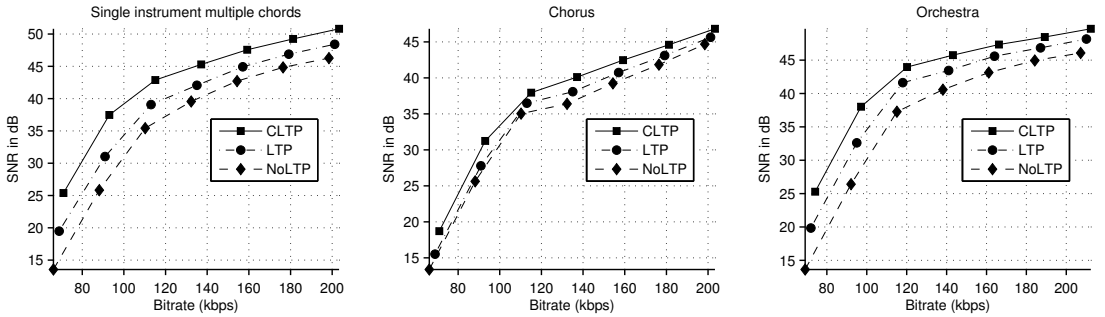

Figure 4.4: Signal to quantization noise ratio versus bit-rates of the competing coders for Bluetooth SBC experiments, evaluated and averaged over files in each of the three classes of dataset

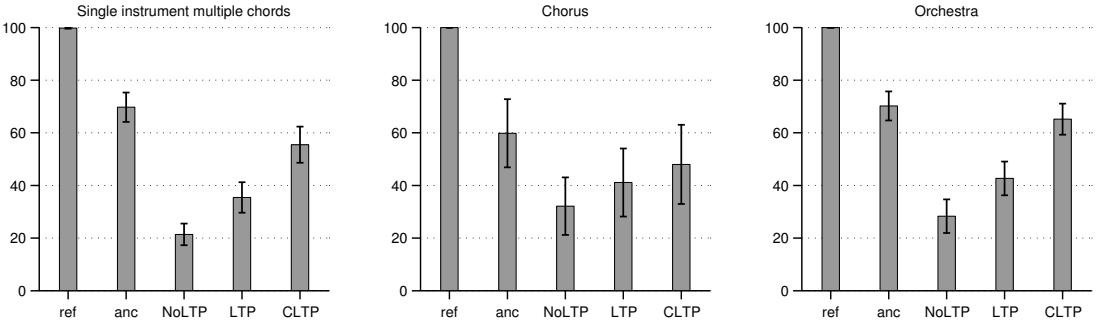

Figure 4.5: MUSHRA listening test average scores with 95% confidence intervals, comparing Bluetooth SBC encoders with no LTP, LTP and proposed CLTP, for the three classes of dataset

5 different versions of each audio sample: a hidden reference (ref), a 3.5 kHz low-pass filtered anchor (anc), and samples encoded with no LTP, LTP and the proposed CLTP. The average MUSHRA scores with the 95% confidence intervals, for the three classes of the test dataset, are shown in Fig. 4.5. The subjective evaluation results confirm that the significant gains in objective criteria translate to substantial subjective quality improvements.

4.6.2 Results for MPEG AAC

We compare the following four AAC LD coders in our experiments:

- MPEG AAC LD reference coder with no LTP (referred to in figure as "NoLTP")
- MPEG AAC LD reference coder with standard LTP tool
- Proposed MPEG AAC LD coder with CLTP
- Proposed MPEG AAC LD coder with low decoder complexity variant of CLTP (referred to in figure as "CLTP-LDC").

All coders employ a simple psychoacoustic model based on the MPEG reference software. Both variants of the proposed CLTP coders uses $N_{\min} = 23, N_{\max} = 800, P_{\max} = 5, \mathbf{N_G} = 4, S = 64$, and $\mathbf{G}_{(i,n)}$ quantization levels of $\{0.5, 0.75, 1, 1.25\}$. The low decoder complexity variant of CLTP coder uses $\mathbf{N}_r = 10$, $\mathbf{N}_\theta = 20$, $\mathbf{N}_N = 10$. Note that the CLTP side information rate varies for every frame depending on the estimated parameters and this is included in the total rate.

Objective evaluation results

For thorough objective evaluation, all coders were evaluated at bit-rates in the range of 20 to 40 kbps. The distortion (MNMR) was calculated for each

frame, and averaged across frames to arrive at a single distortion value for each file called average MNMR (AMNMR). The AMNMR achieved at different bit-rates averaged over files in each of the three classes of the dataset, was used to generate the operational RD plots shown in Fig. 4.6.

As is evident from the RD plots, the standard LTP provides almost no improvements in AMNMR over no-LTP for most of the polyphonic files, while in some cases improvement of around 1dB was observed. These modest gains were due to the fact that these files had a dominant periodic component (e.g., in mfv) and the LTP tool succeeded in providing a good prediction for this dominant component.

The additional performance gains of CLTP, over standard LTP, were considerable for all polyphonic music files and in the range of 1 to 3 dB at various bit-rates. This reinforces the argument that the variety of music files, which contain a mixture of periodic components, represents a considerable potential for exploiting inter-frame redundancies, even in perceptual audio coders, but the standard LTP tool is limited in its capability to do so. Note that the performance gains in the chorus file are less impressive at 0.3dB, and we attribute this to the fact that the pitch periods in this file vary rapidly in time and thus the efficacy of CLTP, which depends on matching periodic components' waveforms, is compromised. A first step towards addressing this drawback is provided in Chapter 5, wherein

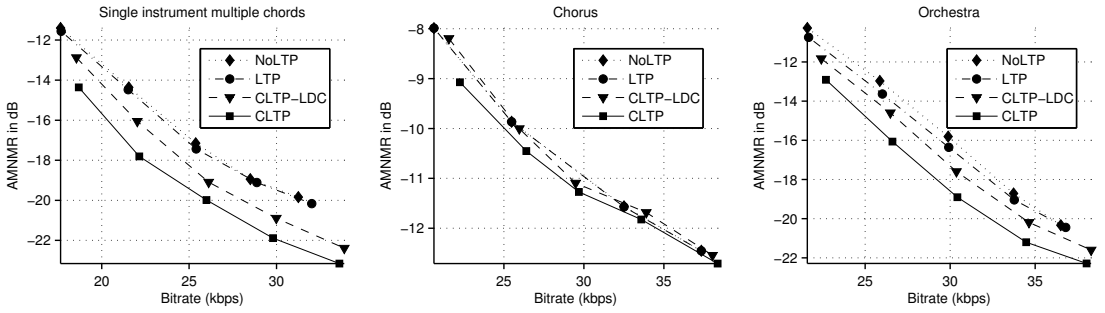

Figure 4.6: Average per-frame distortion at various bit-rates of the different coders for the MPEG AAC experiments, evaluated and averaged over files for each of the three classes of dataset.

accommodating pitch variations in audio signals with single periodic component is proposed. Also note that the additional performance gains of the low decoder complexity variant of CLTP, over standard LTP, though not as impressive as the full complexity CLTP, were still significant and in the range of 0.6 to 1.8 dB for all polyphonic music files at various bit-rates. Clearly, this variant trades off some performance for decoder complexity reduction (presented in Section 4.6.3).

Subjective evaluation results

The competing MPEG AAC coders were evaluated for subjective quality via the MUSHRA listening tests [23]. Only the full complexity CLTP is included in this test, as its evaluation showcases the best performance that can be achieved with CLTP and the low decoder complexity variant is left out as the performance-complexity tradeoff it provides is already highlighted by the objective results. All

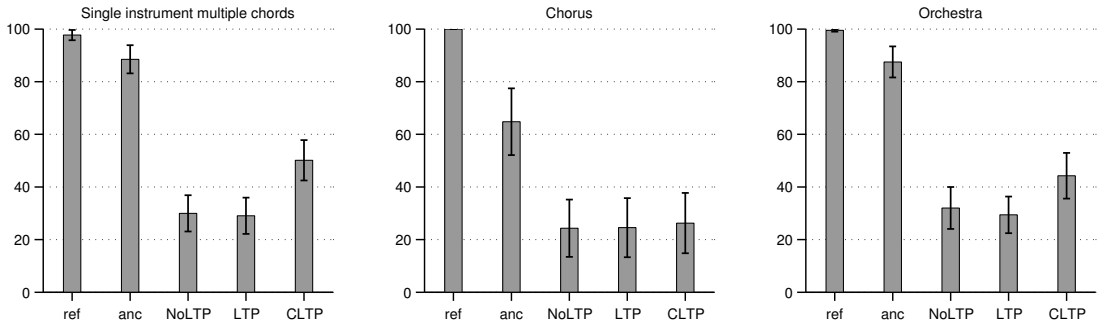

Figure 4.7: MUSHRA listening test average scores with 95% confidence intervals, comparing MPEG AAC encoders with no LTP, standard LTP and proposed CLTP, for the three classes of dataset

codecs were operated at 24kbps. The tests were conducted with 15 listeners and test items were scored on a scale of 0 (bad) to 100 (excellent). Listeners were provided with randomly ordered 5 different versions of each audio sample: a hidden reference (ref), a 3.5 kHz low-pass filtered anchor (anc), and samples encoded with no LTP, standard LTP and the proposed full complexity CLTP. The results of this test (the average MUSHRA scores and the 95% confidence intervals) for the three classes of the test dataset are shown in Fig. 4.7. The subjective evaluation results concur with the previously discussed objective evaluation results, and corroborate the fact that the proposed CLTP technique provides substantial improvements over the LTP tool of the MPEG AAC standard for a variety of polyphonic signals, while optimizing perceptual distortion criteria.

4.6.3 Complexity

The prediction technique proposed in this chapter is of higher complexity than LTP, mainly due to the elaborate estimation of parameters performed for each P, recursively. The complexity information for crude implementations of the proposed coders for the evaluated dataset is listed in Table 4.2. As the main objective of this work was to validate the concept of CLTP, no significant effort was put into minimizing complexity of the proposed coders. Note that there are many straightforward ways to drastically reduce CLTP complexity, e.g., controlling the convergence criteria of the recursive technique to optimize the tradeoff between complexity and prediction quality. Similarly, the high complexity of using LTP over not using prediction (specifically in Bluetooth SBC), can be reduced using well known techniques [18, 19] that tradeoff estimation accuracy for complexity, e.g., using subsampled version of data while estimating lags, and reducing the number of elements in equations (4.13) and (4.15). Also as CLTP parameter estimation complexity is mainly due to multiple iterations of LTP parameter estimation in a loop, any factor of reduction in LTP complexity, translates to almost same factor of reduction in CLTP complexity. We also observe from Table 4.2 that our proposed low decoder complexity variant for MPEG AAC is successful in its objective of keeping the decoder complexity under check.

Encoder	Encoder complexity		Decoder complexity	
	CLTP over LTP	LTP over NoLTP	CLTP over LTP	LTP over NoLTP
Bluetooth SBC	51x	75x	51x	75x
MPEG AAC	30x	6x	120x	1.02x
MPEG AAC low decoder complexity	27x	6x	1.03x	1.02x

Table 4.2: Complexity of the proposed coders

4.7 Conclusion

This chapter demonstrates that the derivation of a long term prediction technique from basic principles, coupled with appropriate parameter estimation, results in substantial improvement in compression efficiency for polyphonic audio signals. Contrary to the existing LTP technique, which predicts a mixture of periodic signals via a compromised shared lag, the proposed technique predicts individual components optimally from the most recently available reconstructed samples. We also propose an effective, recursive technique for estimation of the filter parameters. This technique was deployed to predict subband samples in the ultra low delay Bluetooth SBC, as its compression efficiency is limited due to very short block lengths. For deploying CLTP in MPEG AAC, we proposed a computationally efficient two stage estimation of the filter parameters, specifically

adapted to the needs of optimizing perceptual criteria. This is achieved by backward adaptive estimation of an initial set of parameters to minimize the mean squared prediction error, followed by a refinement stage, where parameters are adjusted to minimize the perceptual distortion. We also proposed a low decoder complexity variant for MPEG AAC, which employs forward adaptive parameter estimation. Finally the objective and subjective evaluations substantiate the effectiveness of the approach in exploiting redundancies within variety of polyphonic signals. Such inter-frame redundancy removal could potentially recoup most of the performance loss due to low delay.

Chapter 5

Accommodating pitch variations in long term prediction

5.1 Introduction

The MPEG AAC standard (described in Section 2.1) along with the LTP tool (introduced in Section 2.3) that is perceptually optimized as described in Chapter 3 is well suited for periodic signals that are stationary over relatively long durations. However, amongst the commonly occurring periodic signals in audio content, the class of voiced speech (e.g., in movies) and vocals in music is well known to be quasi-stationary and is characterized by small variations in pitch period. Over the duration of a frame, a small pitch variation can lead to predicted samples trailing the current samples by a large margin, thus undermining the effectiveness of the LTP tool. This performance degradation relative to other stationary periodic signals of musical instruments has been extensively documented

in prior LTP related research, including in Chapter 3, where even perceptually motivated optimization did not yield any performance improvements for speech files, and in Chapter 4, the cascaded LTP filter's performance was limited for polyphonic audio signals with speech and vocal content. While various time-warping based solutions to this shortcoming have been considered for speech coders, we propose a novel technique of modifying the LTP filter with a single additional parameter of 'geometric' warping, i.e., the continuous-time *warped* LTP analysis filter is

$$e(t) = x(t) - \mathbf{G}x\left(\frac{t-\mathbf{L}}{\mathbf{A}}\right), \tag{5.1}$$

where $\mathbf{L}$ is the pitch period, $\mathbf{G}$ is the scaling factor and $\mathbf{A}$ is the new 'geometric' warping parameter. Effectively, periodicity of past samples is warped by a factor $\mathbf{A}$ and the adjusted samples are provided as prediction for the current samples. Repeating the operation recursively per pitch period provides a prediction for the entire current frame. Note that the 'geometric' warping parameter $\mathbf{A}$, along with a non-integer pitch period $\mathbf{L}$, introduce fractional delay in the filter. We approximate this fractional delay via linear interpolation. A clear advantage of our technique, compared to other time-warping based LTP techniques, is the very marginal increase in side information, as the single additional 'geometric' warping parameter efficiently accounts for small pitch period changes.

Clearly, for this warped LTP filter to be effective in MPEG AAC, a parameter estimation technique, which accounts for the perceptual distortion criteria, is critical. To achieve this at acceptable complexity, we propose a three stage parameter estimation technique. In the first stage, a simple LTP filter's parameters of pitch period and scaling factor are estimated via well known open-loop technique that minimizes the mean squared prediction error. This forms our preliminary unwarped filter, i.e., with a 'geometric' warping parameter of 1. Next, in a small search space around the preliminary estimates of pitch period, scaling factor, *and warping parameter*, we find the S best parameter sets that minimize the mean squared closed-loop prediction error. The subset of bands which result in best prediction gains is also found for these candidates or "survivors". Finally, each of these S survivors is rate-distortion evaluated via the two-loop search based technique [1,6], and the one that minimizes the perceptual distortion at a given rate is selected as the final parameter set. These parameters are sent as side information to the decoder, with pitch period differentially coded and quantized, while scaling factor and warping parameter directly quantized. Effectiveness of our proposed approach on speech and vocals is demonstrated by the considerable performance gains observed in the objective and subjective evaluations. The results of this work have appeared in [29].

This chapter is structured as follows: The proposed modifications to the LTP filter and its integration with MPEG AAC is described in Section 5.2. Results are presented in Section 5.3, and the chapter concludes in Section 5.4.

Note that although the emphasis in this chapter is on the AAC-LD standard, the approach proposed is generic and can easily be extended to other audio coders.

5.2 Accommodating pitch variations in LTP

The problem of pitch variations is well known in the field of speech compression and various techniques have been proposed to accommodate pitch variations in LTP. The adaptive code books of code-excited linear predictive (CELP) speech coders are based on the principles of LTP, and updating the pitch period at small regular intervals was proposed in [30] to accommodate pitch variations in CELP coders. The side-information cost was reduced by restricting pitch changes to a small range. In [31] the adaptive code book was generated by time-warping the previously reconstructed samples to accurately account for the pitch changes. This approach was further generalized to waveform interpolative coding [32], where the speech segments were coded after their pitch period is normalized. Our proposed method uses similar underlying principles as the time-warped prediction technique employed in speech coders, but is specifically adapted for the audio coding

framework with minimal increase in side information. Note that the non-linearity introduced due to pitch variation causes inefficiency even in the MDCT. Thus time-warped MDCT [33] was introduced in the recent unified speech-audio coding standard [34]. Here samples of the current frame are warped to maintain a constant pitch period so that transformation via MDCT results in better energy compaction. The warping factor is updated at frequent regular intervals and sent as side information. While time-warped MDCT effectively accounts for pitch variations within a frame, the LTP tool used for inter-frame prediction is ineffective in accommodating pitch variations. We thus propose a novel way of addressing this shortcoming in the following subsections.

5.2.1 Proposed filter structure

A continuous time single tap LTP analysis filter is given as,

$$e(t) = x(t) - \mathbf{G}x(t - \mathbf{L}), \tag{5.2}$$

where $\mathbf{L}$ is the pitch period (or lag) and $\mathbf{G}$ is the filter gain. We propose to accommodate pitch variations by modifying this filter to have a constant 'geometric' warping factor $\mathbf{A}$, i.e.,

$$\begin{aligned} e(t) &= x(t) - \mathbf{G}x\left(\frac{t - \mathbf{L}}{\mathbf{A}}\right) \\ &= x(t) - \mathbf{G}x(t - \mathcal{L}(t, \mathbf{L}, \mathbf{A})), \end{aligned} \tag{5.3}$$

where $\mathcal{L}(t, \mathbf{L}, \mathbf{A}) = (\mathbf{L} + t(\mathbf{A} - 1))/\mathbf{A}$ is the time varying lag function. For the discrete-time case, we allow non-integer lags that are approximated via linear interpolation, resulting in the following discrete-time LTP analysis filter,

$$\begin{aligned} e[m] &= x[m] - \\ &\quad \mathbf{G}\mathcal{F}(m, \mathbf{L}, \mathbf{A})x[m - \lfloor \mathcal{L}(m, \mathbf{L}, \mathbf{A}) \rfloor + 1] - \\ &\quad \mathbf{G}(1 - \mathcal{F}(m, \mathbf{L}, \mathbf{A}))x[m - \lfloor \mathcal{L}(m, \mathbf{L}, \mathbf{A}) \rfloor], \end{aligned} \tag{5.4}$$

where $\mathcal{L}(m, \mathbf{L}, \mathbf{A}) = (\mathbf{L} + m(\mathbf{A} - 1))/\mathbf{A}$ is the discrete-time lag function and $\mathcal{F}(m, \mathbf{L}, \mathbf{A}) = \mathcal{L}(m, \mathbf{L}, \mathbf{A}) - \lfloor \mathcal{L}(m, \mathbf{L}, \mathbf{A}) \rfloor$ is the fractional part of the lag. We also allow the reference lag, $\mathbf{L}$, to be non-integer. The warped LTP filter of (5.4) forms the basis of our proposal to improve performance of audio coders for speech and vocals.

5.2.2 Frame Prediction

Closed loop prediction is needed, where all samples of the current frame, $x[m]$, $nK \leq m < (n+2)K$, are predicted from previously reconstructed samples, $\hat{x}[m]$, $m < nK$, in order to avoid error propagation and decoder drift. However, given the warped LTP filter, if the frame length is longer than the pitch period then we would have to utilize samples that have not yet been encoded. This problem is addressed in the standard LTP tool by predicting the entire cur-

rent frame from previously reconstructed samples, which results in samples being predicted from a fairly distant past, at the cost of significant loss of correlation. We address this problem via a better approach, known as 'looped prediction' in the error concealment literature [35], and 'virtual search procedure' in the speech coding literature [36, 37]. This approach is generating the predicted samples, $\tilde{x}_n[m],\ 0 \leq m < K$, for the current frame using the synthesis filter corresponding to the warped LTP filter of (5.4), with prediction residue as 0 and previously reconstructed samples as the initial state. That is, if M previously reconstructed samples are available as history, $\tilde{x}_n[m] = \hat{x}[nK + m]$ for $-M \leq m \leq -1$, and for $0 \leq m < 2K$,

$$\begin{aligned} \tilde{x}_n[m] &= \mathbf{G}\mathcal{F}(m, \mathbf{L}, \mathbf{A})\tilde{x}_n[m - \lfloor \mathcal{L}(m, \mathbf{L}, \mathbf{A}) \rfloor + 1] \ + \\ &\quad \mathbf{G}(1 - \mathcal{F}(m, \mathbf{L}, \mathbf{A}))\tilde{x}_n[m - \lfloor \mathcal{L}(m, \mathbf{L}, \mathbf{A}) \rfloor]. \end{aligned}$$

This basically is repeating an appropriately scaled and warped pitch period number of the latest reconstructed samples, so as to generate the entire frame's prediction.

5.2.3 Parameter estimation

For the proposed filter to be effective as an inter-frame decorrelation tool in MPEG AAC, it is crucial that the parameter estimation procedure takes into

account the perceptual distortion criteria. We propose a three stage parameter estimation technique to tackle this important problem at an acceptable complexity. The three stages are described below.

Stage 1: Preliminary LTP filter

In the first stage we estimate a single-tap open-loop LTP filter,

$$e[m] = x[m] - \mathbf{G}_n x[m - \mathbf{L}_n]. \tag{5.5}$$

We use source samples, $x[m]$, in the well known mean squared prediction error minimizing LTP parameter estimation technique to get,

$$\mathbf{L}_n = \underset{\mathbf{L}\in[L_{\min},L_{\max}]}{\arg\max} \frac{\sum_{m=0}^{2K-1} x[m+nK]x[m+nK-\mathbf{L}]}{\sqrt{\sum_{m=0}^{2K-1} x^2[m+nK-\mathbf{L}]}},$$

$$\mathbf{G}_n = \frac{\sum_{m=0}^{2K-1} x[m+nK]x[m+nK-\mathbf{L}_n]}{\sum_{m=0}^{2K-1} x^2[m+nK-\mathbf{L}_n]},$$

where $L_{\min}, L_{\max}$ are the end points of the pitch period search range. This forms our preliminary LTP filter with lag, $\mathbf{L}_n$ (an integer for now), gain, $\mathbf{G}_n$ and 'geometric' warping factor, $\mathbf{A}_n = 1$.

Stage 2: Closed-loop parameter estimation

In this stage we estimate closed-loop prediction parameters, where the predicted samples are calculated via the 'looped' prediction technique of Section 5.2.2. To simplify the parameter search procedure and to transmit these parameters as side information to the decoder we uniformly quantize all the parameters. $\mathbf{G}_n$ is limited to the range $[\mathbf{G}_{\min}, \mathbf{G}_{\max}]$ and uniformly quantized with $N_{\mathbf{G}}$ levels. Non-integer $\mathbf{L}_n$ is allowed, with its fractional value uniformly quantized to $N_{\mathbf{L}}$ levels. The warping parameter $\mathbf{A}_n$ was observed to be sensitive to quantization errors, and hence it is derived from the secondary parameter, $\mathbf{\Delta L}_n$, as,

$$\mathbf{A}_n = \frac{\mathbf{\Delta L}_n}{\mathbf{L}_n} + 1. \tag{5.6}$$

This ensures that $\mathbf{A}_n\mathbf{L}_n = \mathbf{L}_n + \mathbf{\Delta L}_n$, i.e., the pitch period, $\mathbf{L}_n$, increases by $\mathbf{\Delta L}_n$ after warping. $\mathbf{\Delta L}_n$ is sent as side information to the decoder after limiting it to the range of $[\mathbf{\Delta L}_{\min}, \mathbf{\Delta L}_{\max}]$ and uniformly quantizing with $N_{\mathbf{\Delta L}}$ levels. To find the prediction error minimizing parameters set, a straightforward approach would be to exhaustively try all possible combinations of the parameters. Instead, to keep complexity in check, we use the preliminary LTP filter parameters as an 'informed' initialization, and search for parameters that minimize the closed-loop prediction error in the neighborhood of the initial parameters. That is, the three parameters are restricted to $P_{\mathbf{L}}, P_{\mathbf{G}}, P_{\mathbf{\Delta L}}$ number of choices in their

respective quantized domain, with preliminary parameters from first stage, $\mathbf{L}_n$, $\mathbf{G}_n$, and $\mathbf{\Delta L}_n = 0$, at the center of the search space. Evaluating all the $P_{\mathbf{L}} P_{\mathbf{G}} P_{\mathbf{\Delta L}}$ combinations for the closed-loop prediction error, produces the locally optimal parameters set. But recall that, eventually, we must estimate the parameters set that minimizes the perceptual distortion at the given rate. In a naive approach, even with very modest choices of $P_{\mathbf{L}} = 32, P_{\mathbf{G}} = 16, P_{\mathbf{\Delta L}} = 16$, we would have to evaluate 8192 parameter sets for rate-distortion performance to identify the best perceptual distortion optimizing parameters set. Thus, similar to the standard-compatible LTP optimization technique we proposed in Chapter 3, we use mean squared prediction error as a criterion to "weed out" the vast majority of non-competitive contenders. That is, amongst the $P_{\mathbf{L}} P_{\mathbf{G}} P_{\mathbf{\Delta L}}$ choices of parameter sets, we retain only the top S parameter sets in terms of prediction error minimization. We also retain the per-SFB prediction activating flags (similar to the standard LTP tool) and calculate these flags for each of the S "survivors" via the technique described in Section 2.3.

Stage 3: Perceptual refinement of the parameters

In the final stage, each of the S "survivors" of the previous stage are rate-distortion evaluated via a TLS based technique to identify the parameter set that minimizes the perceptual distortion at the target rate. We also retain the

frame level prediction flag, $\mathbf{F}_n$, which is estimated based on the rate-distortion performance of the prediction disabled case. The overall algorithm is enumerated below:

1. Preliminary LTP filter is open-loop estimated for the current frame in terms of lag, $\mathbf{L}_n$, gain, $\mathbf{G}_n$, and 'geometric' warping factor, $\mathbf{A}_n = 1$ or $\mathbf{\Delta L}_n = 0$.

2. In a small search space around these preliminary parameters, the set of top S closed-loop mean squared prediction error minimizing parameters set is identified. Also for each of these survivors, the per-SFB prediction activating flags are calculated.

3. For each of the S survivors, TLS is employed to optimize the quantization and coding parameters, and to calculate the associated perceptual distortion at the given total rate (which includes the warped LTP filter's side information rate)

4. TLS is also employed for the case when prediction is disabled for the frame (i.e., $\mathbf{F}_n = 0$), to calculate the associated perceptual distortion at the given total rate.

5. Amongst the $S + 1$ cases, the parameters that correspond to the minimum perceptual distortion are employed in encoding the frame.

To efficiently send the lag, $\mathbf{L}_n$, as side information, we differentially encode it subject to the condition $L'_{min} \leq (\mathbf{L}_n - \mathbf{L}_{n-1}) \leq L'_{max}$, where $L'_{\min}, L'_{\max}$ are chosen to reduce the number of bits required to indicate the difference of lags. The prediction side information finally includes; one bit to indicate $\mathbf{F}_n$, $\lceil \log_2(N_{\mathbf{G}}) \rceil$ bits to indicate gain, $\mathbf{G}_n$; $\lceil \log_2(N_{\mathbf{\Delta L}}) \rceil$ bits to indirectly indicate 'geometric' warping factor, $\mathbf{A}_n$; L bits to indicate the SFB-wise flags; one bit to indicate if the lag, $\mathbf{L}_n$, is differentially coded; if being differentially coded, $\lceil \log_2(N_{\mathbf{L}}(L'_{\max} - L'_{\min})) \rceil$ bits to indicate the difference $(\mathbf{L}_n - \mathbf{L}_{n-1})$, else $\lceil \log_2(N_{\mathbf{L}}(L_{\max} - L_{\min})) \rceil$ bits to indicate the actual lag, $\mathbf{L}_n$. This prediction side information, along with the core AAC bitstream, is sent to the decoder.

Note that the choice of parameters $P_{\mathbf{L}}, P_{\mathbf{G}}, P_{\mathbf{\Delta L}}$, and S controls the tradeoff between complexity and performance.

5.3 Results

In this section we provide results of the evaluations we conducted to compare the following three AAC LD coders:

- MPEG reference encoder with no LTP (referred to in figure as "NoLTP")
- MPEG reference encoder with standard LTP tool (referred to in figure as "LTP")

- Proposed encoder with the warped LTP filter (referred to in figure as "propLTP")

A simple psychoacoustic model based on the MPEG reference software was employed by all the coders. We chose the following parameters for the proposed coder; $\mathbf{G}_{\min} = 0.57$, $\mathbf{G}_{\max} = 1.2$, $N_{\mathbf{G}} = 256$; $\mathbf{\Delta L}_{\min} = -2$, $\mathbf{\Delta L}_{\max} = 1.75$, $N_{\mathbf{\Delta L}} = 16$; $L_{\min} = 23$, $L_{\max} = 800$, $N_{\mathbf{L}} = 8$, $L'_{\min} = -4$, $L'_{\max} = 3.875$; $P_{\mathbf{L}} = 32$, $P_{\mathbf{G}} = 16$, $P_{\mathbf{\Delta L}} = 16$, and $S = 64$. For this choice of parameters, when all 36 SFBs of a 44.1/48 kHz audio file are coded, the prediction side information is a maximum of 63 bits (corresponding to 5.4/5.9 kbps) and a minimum of 56 bits (corresponding to 4.8/5.3 kbps). Note that this prediction side information is included in the rate totals. The experiments are conducted with single channel 44.1/48kHz audio samples from the class of speech and vocals in the standard MPEG and EBU SQAM database. We extracted a 4 seconds portion of each audio file for time efficient evaluation. The resulting subset is:

- Speech: mgerman, fenglish
- Pop vocals: vega
- Opera vocals: soprano, tenor

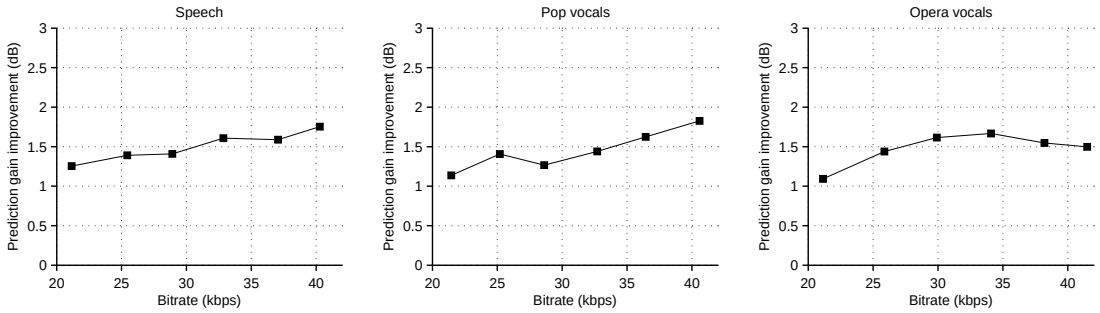

Figure 5.1: Prediction gain improvement (in dB) of the proposed coder over standard LTP based coder versus bit-rate.

5.3.1 Objective evaluation results

We use prediction gain (signal to prediction residue energy ratio) as a measure for objective evaluation. For a thorough evaluation, the standard LTP based coder and the proposed coder were evaluated at bit-rates in the range of 20 to 40 kbps and the prediction gain improvement of the proposed coder over the standard LTP based coder was calculated. The plots of prediction gain improvement (averaged over files of a subset) at different bit-rates is shown in Fig. 5.1.

We can clearly see from the plots that the proposed modifications to the LTP tool consistently provides a considerable prediction gain improvement of greater than 1 dB and on the average 1.5 dB. This substantiates that the proposed approach is indeed effective in accommodating pitch variations in speech and vocals.

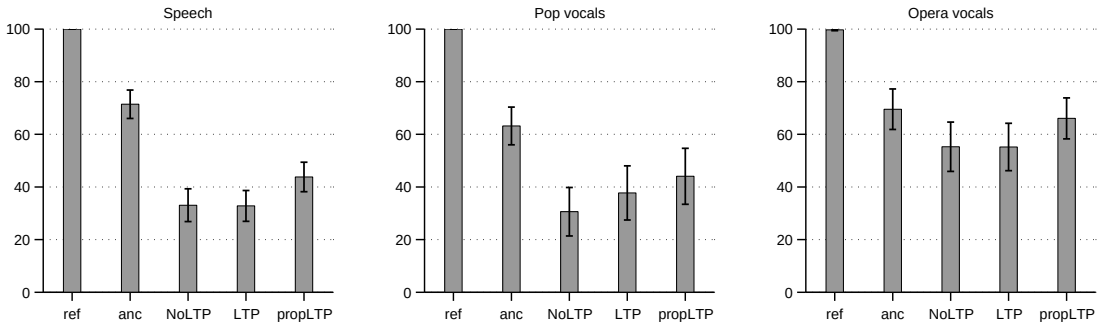

Figure 5.2: MUSHRA listening test comparing no LTP, standard LTP and proposed warped LTP

5.3.2 Subjective evaluation results

MUSHRA listening tests were used to conduct subjective quality evaluation of all the coders. The operating point for all the coders was at around 32 kbps. 15 listeners participated in the tests and scored the test items on a scale of 0 (bad) to 100 (excellent). Participants were provided with randomly ordered 5 different versions of each audio sample: a hidden reference (ref), a 3.5 kHz low-pass filtered anchor (anc), and samples encoded with no LTP, standard LTP and the proposed warped LTP. The test results of average MUSHRA scores, along with the 95% confidence intervals, for each test subset are shown in Fig. 5.2. The subjective evaluations confirm previously observed results that using the standard LTP rarely improves the quality for speech and vocals. While the warped LTP filter, which is appropriately designed to accommodate pitch variations in speech and vocals, provides considerable subjective quality improvement.

5.3.3 Complexity

The approach proposed in this chapter is of higher complexity than standard LTP due to the elaborate estimation of parameters. A crude implementation of the proposed encoder (with no significant effort spent on minimizing code complexity) took, on the average, 18 times longer than standard LTP based encoder for the evaluated dataset. Note that the standard LTP based encoder took, 5 times longer than the encoder with no prediction. Also note that the proposed encoder needs $L_{\max}$ source samples history and $L_{\max}$ previously reconstructed samples history instead of the $3K$ previously reconstructed samples history needed by the standard LTP tool. The proposed decoder needs only $L_{\max}$ samples history instead of the $3K$ samples history needed by the standard LTP tool.

5.4 Conclusion

The work in this chapter demonstrates a novel approach for accommodating pitch variation in long term prediction of speech and vocals in audio coding. Contrary to the existing LTP technique, whose gains diminish for speech and vocals due to the assumption of stationarity for relatively long durations, the proposed approach warps the periodicity of the previously reconstructed samples to take small pitch variations into account. We also propose a three stage technique to

estimate parameters at an acceptable complexity, while accounting for the perceptual criteria of coding in MPEG AAC. Considerable quality improvements demonstrated in the objective and subjective evaluations evidence the effectiveness of the proposed approach for speech and vocals. Such improved inter-frame redundancy removal may be an important bridge for a step towards truly unified speech and audio coding.

Chapter 6

Frame loss concealment of polyphonic audio signals

6.1 Introduction

Audio transmission over networks enables a wide range of applications such as multimedia streaming, online radio and high-definition teleconferencing. These applications are often plagued by the problem of unreliable networking conditions, which leads to intermittent loss of data. Frame loss concealment (FLC) forms a crucial tool amongst the various strategies used to mitigate this issue. The FLC objective is to exploit all available information to approximate the lost frame while maintaining smooth transition with neighboring frames.

Various techniques have been proposed for FLC, amongst which the simple techniques of replacing the lost frame with silence or the previous frame, result in poor quality [38]. Advanced techniques are usually based on source modeling and

were inspired from solutions to the equivalent problem of click removal in audio restoration [39]. For example, speech signals have one periodic component, and FLC techniques based on pitch waveform repetition are widely used. But these techniques fail for most audio signals which are polyphonic in nature, because they contain a mixture of periodic components. In principle, the mixture is itself periodic with period equalling the least common multiple (LCM) of its individual periods, but the signal rarely remains stationary over this extended period, rendering the pitch repetition techniques ineffective. To handle signals with multiple periodic signals, various frequency domain techniques have been proposed. FLC techniques based on sub-band domain prediction [35, 40] handle multiple tonal components in each sub-band via a higher order linear predictor. This approach does not utilize samples from future frames and is effectively an extrapolation technique with the shortcoming that it disregards smooth transition into future frames. An alternative approach to perform FLC in the modified discrete cosine transform (MDCT) domain, which accounts for future frames, was developed in our group [41]. This technique isolated tonal components in MDCT domain and interpolated the relevant missing MDCT coefficients of the lost frame using available past and future frames. Its performance gains, while substantial, were limited in the presence of multiple periodic components in polyphonic signals, whenever isolating individual tonal components was compromised by the frequency resolu-

tion of MDCT. This problem is notably pronounced in low delay coders which use low resolution MDCT.

The shortcomings of existing FLC techniques motivated the approach proposed in this chapter and builds on our work on efficient compression of polyphonic signals presented in Chapter 4, to predict each periodic component in the time domain from its immediate past. Specifically, a long term prediction filter corresponding to each periodic component is cascaded to form the *cascaded long term prediction* (CLTP) filter. A preliminary set of parameters for these filters is estimated from past reconstructed samples via a recursive divide and conquer technique. In this recursion, parameters of one filter in the cascade are estimated while parameters of the others are fixed, and the process is iterated until convergence. Amongst these preliminary parameters, the pitch periods of each component are assumed to be stationary during the lost frame, while the filter coefficients are enhanced via a multiplicative factor to minimize the squared prediction error across future reconstructed samples. The predicted samples required for this minimization are generated via the 'looped' prediction (described in [35]), wherein given all the parameters, the filter is operated in the synthesis mode in a loop, with predictor output acting as input to the filter as well. The minimization is achieved via the well known quasi-Newton method called limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [42] along with backtrack-

ing line search [43] for step size. Similarly, another set of multiplicative factors are generated for predicting the lost frame in the reverse direction from future samples. Finally the two sets of predicted samples are overlap-added with a triangular window to reconstruct the lost frame. The proposed scheme is incorporated within an MPEG AAC low delay (LD) mode [1,2] decoder, with band-wise energy adjustment when there is a large deviation from the geometric mean of energies in the bands of adjacent frames. Subjective and objective evaluation results for a wide range of polyphonic signals substantiate the effectiveness of the proposed technique. The results of this work have appeared in [44]. The CLTP filter of (4.9) introduced in Chapter 4 plays a central role in the FLC technique proposed in this chapter, but the filter is suitably modified to utilize all the information available for reconstructing a lost frame.

This chapter is structured as follows: The proposed technique for frame loss concealment is described in Section 6.2. Results are presented in Section 6.3, and the chapter concludes in Section 6.4.

6.2 CLTP for Frame Loss Concealment

When a frame is lost and the CLTP filter is known, the samples of the lost frame are predicted by first padding the previously reconstructed samples by zeros

and then operating the synthesis filter $1/H_c(z)$ in this region of zeros, while using the previously reconstructed samples as initial state. This type of technique was called 'looped' prediction in [35], wherein output samples are recursively fed back to the filter to generate future predicted samples. Clearly estimation of parameters is critical to the performance of this predictor and the FLC technique. The proposed parameter estimation method and details of the overall technique are described in the following subsections.

6.2.1 Estimation of preliminary set of CLTP parameters

We assume the signal to be quasi-stationary in the vicinity of the lost frame and estimate using the past reconstructed samples, the pitch period and a preliminary set of filter coefficients. This is achieved at acceptable complexity via the recursive "divide and conquer" technique introduced in Section 4.4 of Chapter 4, using $\hat{x}[m], \ -M_p \leq m < 0$, the M_p past reconstructed samples available to the FLC module.

6.2.2 CLTP parameter refinement

In the networking applications where FLC is mainly used, availability of future frames while reconstructing a lost frame is usually assured. That is, if a frame with K samples is lost, usually M_f future reconstructed samples given as

$\hat{x}[m],\ K \leq m < K + M_f$, are available to the FLC module. Using these samples to reconstruct a lost frame that transitions smoothly into the future is critical for good concealment quality and this is achieved by refining the preliminary CLTP filter parameters. We nevertheless assume that the pitch periods N_i are stationary in the vicinity of the lost frame, and hence employ multiplicative factors G_i to form an updated CLTP filter,

$$H_c(z) = \prod_{i=0}^{P-1}(1 - G_i(\alpha_i z^{-N_i} + \beta_i z^{-N_i+1})). \tag{6.1}$$

The CLTP filter allows us to generate the predicted future samples $\tilde{x}[m],\ K \leq m < K + M_f$,via 'looped' prediction. We now adjust the multiplicative factors G_i such that they minimize the squared prediction error, i.e., the cost function is given as

$$\varepsilon(\mathbf{G}) = \sum_{m=K}^{K+M_f-1} (\hat{x}[m] - \tilde{x}[m])^2, \tag{6.2}$$

where $\mathbf{G} = [G_0, \ldots, G_{P-1}]$ is the set of all multiplicative factors. Since the cost function has a complex dependency on $\mathbf{G}$, we use a generic quasi-Newton optimization method called limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [42] method. This is chosen as it converges faster than a plain gradient descent method. More details about this iterative method can be found in [42]. Since calculating the gradient function of the cost function is also complex, we approximate the partial derivatives as a difference in cost function for a small

perturbation, i.e.,

$$\frac{\partial \varepsilon(\mathbf{G})}{\partial G_i} \approx \frac{\varepsilon(\bar{\mathbf{G}}_\mathbf{i}, G_i + h) - \varepsilon(\bar{\mathbf{G}}_\mathbf{i}, G_i)}{h}, \tag{6.3}$$

where $\bar{\mathbf{G}}_\mathbf{i}$ is the set of all multiplicative factors except G_i. Also the step size used within the L-BFGS algorithm is adapted via the backtracking line search method described in [43]. We note that the cost function is not convex and thus the above optimization cannot guarantee a global optima. But, as we will see experimentally, locally optimal multiplicative factors provide substantial improvement in concealment quality as they adapt the prediction filter parameters to exploit the available future reconstructed samples. Given the resulting CLTP filter, one set of samples of the lost frame is generated via the 'looped' prediction as $\tilde{x}[m],\ 0 \leq m < K$.

6.2.3 Bidirectional prediction

Further improvement in concealment quality is achieved by using samples predicted in the reverse direction from the future samples. To use an approach similar to the one described above for prediction in the forward direction, a reversed set of reconstructed samples available to the FLC module, is defined as $\hat{x}_r[m] = \hat{x}[K-1-m]$. This set in the range $-M_f \leq m < 0$ forms the new "past" reconstructed samples and the range $K \leq m < K + M_p$ forms the new "future" reconstructed samples. Since pitch periods are assumed to be stationary close to the lost frame, we start with the same preliminary CLTP filter estimated in Sec-

tion 6.2.1 for the reverse direction as well and estimate a new set of multiplicative factors G_i^r via the technique described in Section 6.2.2, to form the reverse CLTP filter,

$$H_c^r(z) = \prod_{i=0}^{P-1}(1 - G_i^r(\alpha_i z^{-N_i} + \beta_i z^{-N_i+1})). \tag{6.4}$$

Given this reverse CLTP filter, another set of samples of the lost frame is generated via the 'looped' prediction as $\tilde{x}_r[m],\ 0 \leq m < K$. Finally the overall lost frame $\tilde{x}_o[m],\ 0 \leq m < K$ is generated as a weighted average of the two sets as,

$$\tilde{x}_o[m] = \tilde{x}[m]g[m] + \tilde{x}_r[K-1-m](1-g[m]), \tag{6.5}$$

where $g[m] = (1 - m/(K-1))$ are the weights which are proportional to each predicted sample's distance from the set of reconstructed samples used for their generation.

6.2.4 Integration within MPEG AAC-LD

MPEG AAC-LD coder segments data into 50% overlapped frames of length $K = 1024$. Thus one frame data loss results in inability to reconstruct K samples. We use $M_p = 2K$ past reconstructed samples and $M_f = K/2$ future reconstructed samples. Note that to have $M_f = K/2$ future samples, 2 future frames have to be available to the FLC module. This requirement is same as the energy interpolation method proposed in [41], while more than the sub-band domain prediction

based method proposed in [35], as it requires only one future frame to adjust energy within each band of the lost frame. The sub-band domain prediction based technique has been observed to result in poor concealment quality when compared to the method proposed in [41] and we hypothesize this to be due to the fact that the prediction does not account for smooth transition into future samples. Thus we emphasize on having future samples available and for this we need at least 2 future frames, as with only one future frame no future samples can be reconstructed due to the overlapped frames and the aliasing introduced during MDCT. Given the data of frame n is lost and the neighboring reconstructed samples are available, the FLC module first estimates the preliminary set of CLTP parameters via the method described in Section 4.4 of Chapter 4, with the parameters $P = 3$, $N_{min} = 50$, $N_{max} = 800$, $Y_{\text{start}} = -K/2$, and $Y_{\text{end}} = -1$. Then these parameters are refined to account for future reconstructed samples as described in Section 6.2.2 and one set of samples of the lost frame is generated. Another set is generated via prediction in the reverse direction from future samples and the overall reconstruction of the lost frame is obtained via the method described in Section 6.2.3. These K reconstructed samples are now transformed into MDCT domain, which enables utilizing the aliased samples from adjacent frames for final reconstruction and also enables maintaining a smooth transition in energies between adjacent frames. For energy adjustment the MDCT coefficients are divided

into scale-factor bands as described in the standard [1] and for each band l the energy in all three frames $e_n[l]$, $e_{n-1}[l]$ and $e_{n+1}[l]$ is calculated. Now energy in the reconstructed frame is corrected by comparing it with the geometric mean $e_{gm}[l] = \sqrt{e_{n-1}[l]e_{n+1}[l]}$ and a gain factor $f[l]$, which is multiplied with all MDCT coefficients of the band l, is calculated as,

$$f[l] = \begin{cases} \sqrt{\frac{e_{gm}[l]}{e_n[l]}}, & \text{if } \frac{e_n[l]}{e_{gm}[l]} > T \text{ or } \frac{e_n[l]}{e_{gm}[l]} < 1/T, \\ 1, & \text{otherwise.} \end{cases} \tag{6.6}$$

That is, if the energy in a band deviates a lot from the geometric mean of energies in corresponding bands of adjacent frames, then it is corrected to the geometric mean. The threshold is chosen as $T = 5$. After multiplying the MDCT coefficients with their corresponding gain factors, final time domain samples are generated via the inverse MDCT process.

6.3 Results

In experiments for this chapter, MPEG reference AAC-LD encoder is operated at 64 kbps to generate the bit-streams and the following four decoder modes are compared:

- Reference decoder with no frame loss

- Reference decoder with sub-band prediction based FLC module as proposed in [35, 40] (further referred as SBP-FLC)
- Reference decoder with MDCT domain energy interpolation FLC module as proposed in [41] (further referred as MDCT-FLC)
- Reference decoder with the proposed CLTP based FLC module (further referred as CLTP-FLC)

For decoders operating with FLC module the frames were randomly dropped at the rate of 10%, with same frames dropped in every decoder for a fair comparison. Also for simplicity, loss of consecutive frames was not allowed. The sub-band prediction based FLC module was operated at best quality by deciding to switch to shaped noise insertion only after checking prediction gain in all 32 sub-bands. The experiments are conducted with 44.1/48 kHz single channel audio sample subset from the EBU-SQAM and MPEG dataset. We restrict the length of each test file to 4 seconds to reduce evaluation times. The test subset includes:

- Single instrument multiple chords: Grand Piano, Guitar, Harp, Tubular Bells
- Orchestra: Mfv, Mozart

Filename	SBP-FLC	MDCT-FLC	CLTP-FLC
Piano	-3.16	-0.67	5.10
Guitar	-1.95	0.19	7.15
Harp	-3.59	-1.77	3.80
Bells	-2.08	0.06	4.26
Mfv	2.27	0.34	11.53
Mozart	-2.03	1.22	8.4
Average	-1.76	-0.11	6.71 (+6.82)

Table 6.1: SSNR in dB for various FLC techniques

6.3.1 Objective evaluation results

We first evaluate segmental signal to noise ratio (SNR) as an objective measure. Segmental SNR (SSNR) is the average of SNR in dB at each of the lost frame. For SSNR the signal energy is of the originally decoded MDCT coefficients and noise energy is of the difference between originally decoded MDCT coefficients and the MDCT coefficients generated by an FLC module. SSNR results for each FLC technique, evaluated for all the files is given in Table 6.1. The table clearly shows that the lost frame reconstructed via the proposed FLC technique is closest to the original frame, with an average segmental SNR improvement of on the average 6.82 dB over previously known best technique described in [41].

6.3.2 Subjective evaluation results

Note that the poor SSNR results of the competitive methods is mainly because their objective is not to absolutely match the waveform of the lost frame and

have sections of MDCT coefficients adjusted with random signs. Thus subjective evaluations were conducted to identify the true perceptual gains via the MUSHRA listening tests [23]. The test items were scored on a scale of 0 (bad) to 100 (excellent) and the tests were conducted with 16 listeners. The tests compared the outputs of 3 FLC techniques along with a output decoded with no frame loss. Randomly ordered 6 versions of each audio sample were presented to the listeners and these were a hidden reference (Ref), a 3.5 kHz low-pass filtered anchor (Anc), decoder output with no frame loss (NoLoss), decoder outputs with SBP-FLC, MDCT-FLC and CLTP-FLC module with 10% frame loss. Figure 6.1 shows the results of these tests, which include the average MUSHRA scores and the 95% confidence intervals, for the two types of files. These subjective evaluation results clearly demonstrates the greatly improved quality due to the proposed FLC technique for a variety of polyphonic signals.

6.3.3 Complexity

The proposed CLTP-FLC technique is clearly of higher complexity. As the main objective of this work was to validate the concept of using CLTP for FLC, no significant effort put into minimizing complexity. Without complexity optimization, a crude implementation of the proposed approach was 70 times more

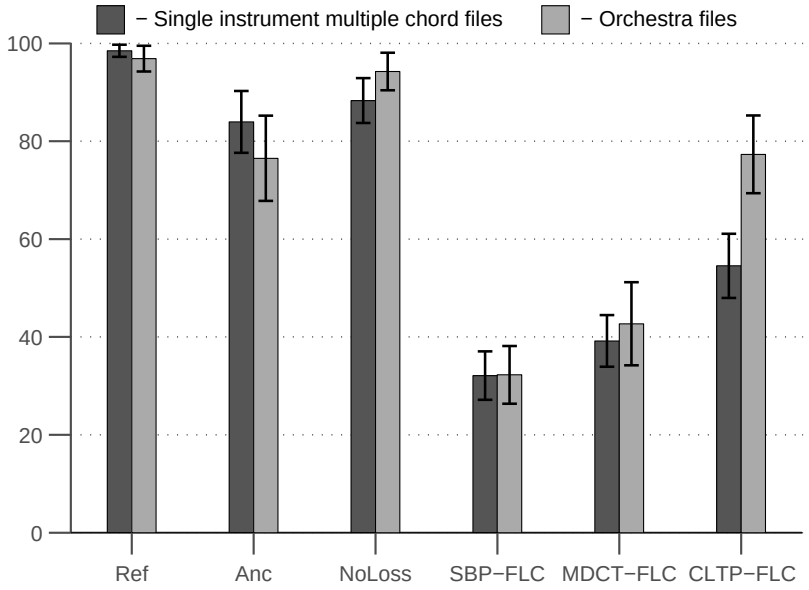

Figure 6.1: MUSHRA listening test results comparing the FLC techniques

complex than the SBP-FLC technique. Clearly there are many simple ways of complexity reduction, but they are all beyond the scope of this work.

6.4 Conclusion

This chapter demonstrates a novel bidirectional cascaded long term prediction based frame loss concealment technique which substantially improves the reconstruction quality for polyphonic signals when used with low delay coders. Contrary to the currently used frequency domain techniques, the proposed technique operates in time domain, but addresses the problem of multiple periodic components by cascading their corresponding LTP filters. The prediction is done

in both directions to better utilize available future samples and the filter parameters in each direction are optimized to account for samples on the other side of the lost frame. Subjective and objective evaluation of the proposed technique deployed within MPEG AAC-LD decoder substantiates the effectiveness of the proposed technique. An important future direction would be enhancing the proposed technique to not assume pitch period to be stationary in the neighborhood of the lost frame.

Bibliography

[1] *Information technology - Coding of audio-visual objects - Part 3: Audio - Subpart 4: General audio coding (GA)*, ISO/IEC Std. ISO/IEC JTC1/SC29 14 496-3:2005, 2005.

[2] E. Allamanche, R. Geiger, J. Herre, and T. Sporer, "MPEG-4 low delay audio coding based on the AAC codec," in *Proc. 106th AES Convention*, May 1999, paper 4929.

[3] J. Ojanperä, M. Väänänen, and L. Yin, "Long term predictor for transform domain perceptual audio coding," in *Proc. 107th AES Convention*, Sep. 1999, paper 5036.

[4] A. Aggarwal, S. L. Regunathan, and K. Rose, "Trellis-based optimization of MPEG-4 advanced audio coding," in *Proc. IEEE Workshop on Speech Coding*, 2000, pp. 142–144.

[5] ——, "A trellis-based optimal parameter value selection for audio coding," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 14, no. 2, pp. 623–633, 2006.

[6] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 advanced audio coding," *J. Audio Eng. Soc.*, vol. 45, no. 10, pp. 789–814, Oct. 1997.

[7] *Bluetooth Specification: Advanced Audio Distribution Profile*, Bluetooth SIG Std. Bluetooth Audio Video Working Group, 2002.

[8] F. de Bont, M. Groenewegen, and W. Oomen, "A high quality audio-coding system at 128 kb/s," in *Proc. 98th AES Convention*, Feb. 1995, paper 3937.

[9] C. Bauer and M. Vinton, "Joint optimization of scale factors and huffman codebooks for MPEG-4 AAC," in *Proc. 6th IEEE Workshop. Multimedia Sig. Proc.*, Sep. 2004.

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

[11] R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 4, pp. 467–477, 1989.

[12] R. Pettigrew and V. Cuperman, "Backward pitch prediction for low-delay speech coding," in *Conf. Rec., IEEE Global Telecommunications Conf.*, Nov. 1989, pp. 34.3.1–34.3.6.

[13] H. Chen, W. Wong, and C. Ko, "Comparison of pitch prediction and adaptation algorithms in forward and backward adaptive CELP systems," in *Communications, Speech and Vision, IEE Proceedings I*, vol. 140, no. 4, 1993, pp. 240–245.

[14] M. Yong and A. Gersho, "Efficient encoding of the long-term predictor in vector excitation coders," *Advances in Speech Coding*, pp. 329–338, Dordrecht, Holland: Kluwer, 1991.

[15] S. McClellan, J. Gibson, and B. Rutherford, "Efficient pitch filter encoding for variable rate speech processing," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 1, pp. 18–29, 1999.

[16] J. Marques, I. Trancoso, J. Tribolet, and L. Almeida, "Improved pitch prediction with fractional delays in CELP coding," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 1990, pp. 665–668.

[17] D. Veeneman and B. Mazor, "Efficient multi-tap pitch prediction for stochastic coding," *Kluwer international series in engineering and computer science*, pp. 225–225, 1993.

[18] P. Kroon and K. Swaminathan, "A high-quality multirate real-time CELP coder," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 5, pp. 850–857, 1992.

[19] J. Chen, "Toll-quality 16 kb/s CELP speech coding with very low complexity," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 1995, pp. 9–12.

[20] W. Kleijn and K. Paliwal, *Speech coding and synthesis*. Elsevier Science Inc., 1995, pp. 95–102.

[21] *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio*, ISO/IEC Std. ISO/IEC JTC1/SC29 11 172-3, 1993.

[22] T. Nanjundaswamy, V. Melkote, E. Ravelli, and K. Rose, "Perceptual distortion-rate optimization of long term prediction in MPEG AAC," in *Proc. 129th AES Convention*, Nov. 2010, paper 8288.

[23] *Method of Subjective Assessment of Intermediate Quality Level of Coding Systems*, ITU Std. ITU-R Recommendation, BS 1534-1, 2001.

[24] S. M. Kay, *Modern Spectral Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[25] T. Nanjundaswamy and K. Rose, "Cascaded long term prediction for coding polyphonic audio signals," in *Proc. IEEE Workshop on App. of Sig. Proc. to Audio and Acoustics*, Oct. 2011, pp. 21–24.

[26] ——, "Perceptually optimized cascaded long term prediction of polyphonic signals for enhanced MPEG-AAC," in *Proc. 131st AES Convention*, Oct. 2011, paper 8518.

[27] A. de Cheveigné, "A mixed speech F_0 estimation algorithm," in *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech '91)*, Sept. 1991.

[28] D. Giacobello, T. van Waterschoot, M. Christensen, S. Jensen, and M. Moonen, "High-order sparse linear predictors for audio processing," in *Proc. 18th European Sig. Proc. Conf.*, Aug. 2010, pp. 234–238.

[29] T. Nanjundaswamy and K. Rose, "On accommodating pitch variation in long term prediction of speech and vocals in audio coding," in *Proc. 133st AES Convention*, Oct. 2012, paper 8767.

[30] M. Yong and A. Gersho, "Efficient encoding of the long-term predictor in vector excitation coders," *Advances in Speech Coding, Kluwer Academic Publishers*, pp. 329–338, 1991.

[31] W. B. Kleijn, R. P. Ramachandran, and P. Kroon, "Generalized analysis-by-synthesis coding and its application to pitch prediction," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, Mar. 1992, pp. 337–340.

[32] W. B. Kleijn and J. Haagen, "Waveform interpolation for coding and synthesis," *Speech Coding and Synthesis, Elsevier Amsterdam*, pp. 175–208, 1995.

[33] B. Edler, S. Disch, S. Bayer, G. Fuchs, and R. Geiger, "A time-warped MDCT approach to speech transform coding," in *Proc. 126th AES Convention*, May 2009, paper 7710.

[34] *Information technology – MPEG audio technologies – Part 3: Unified speech and audio coding*, ISO/IEC Std. ISO/IEC JTC1/SC29 23 003-3, 2012.

[35] J. Herre and E. Eberlein, "Evaluation of concealment techniques for compressed digital audio," in *Proc. 94th AES Convention*, Feb. 1993, paper 3460.

[36] W. Kleijn, D. Krasinski, and R. Ketchum, "Improved speech quality and efficient vector quantization in SELP," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 1988, pp. 155–158.

[37] P. Kabal, J. Moncet, and C. Chu, "Synthesis filter optimization and coding: Applications to CELP [speech analysis]," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 1988, pp. 147–150.

[38] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, 1998.

[39] S. Godsill and P. Rayner, *Digital audio restoration: a statistical model based approach*. Springer verlag, 1998.

[40] R. Sperschneider and P. Lauber, "Error concealment for compressed digital audio," in *Proc. 111th AES Convention*, Nov. 2003, paper 5460.

[41] S.-U. Ryu and K. Rose, "An MDCT domain frame-loss concealment technique for MPEG advanced audio coding," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 2007, pp. 273–276.

[42] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[43] J. Nocedal and S. Wright, *Numerical optimization*. Springer verlag, 1999.

[44] T. Nanjundaswamy and K. Rose, "Bidirectional cascaded long term prediction for frame loss concealment in polyphonic audio signals," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Sig. Proc.*, 2012, pp. 417–420.

[45] R. P. Ramachandran and P. Kabal, "Stability and performance analysis of pitch filters in speech coders," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 7, pp. 937–946, 1987.

[46] A. Said, "Introduction to arithmetic coding-theory and practice," *Hewlett Packard Laboratories Report*, 2004.

Appendices

Appendix A

Stabilization of LTP synthesis filter

Stability of long term prediction filter has been extensively studied in [45], due to widespread usage of pitch filters in speech coding. That paper also analyzed the sufficient stability criteria we have used in our CLTP parameter estimation and specified in equation (4.14) in Chapter 4. A noteworthy property of this condition is that it is asymptotically necessary and sufficient as the pitch period N increases, for the 2 tap LTP filters we have used. While [45] also proposes LTP filter stabilization techniques, they do not guarantee optimality for the given condition. Thus we present here a stability constrained optimal gain factors estimation technique. Recall that the problem definition is to find $\alpha_{(j,N)},\ \beta_{(j,N)}$ which minimize the following error,

$$\varepsilon_N = \sum \begin{array}{lll} (x_j[m] & - & \alpha_{(j,N)} x_j[m-N] \\ & - & \beta_{(j,N)} x_j[m-N+1])^2, \end{array} \tag{A.1}$$

subject to the sufficient stability criterion

$$|\alpha_{(j,N)}| + |\beta_{(j,N)}| \leq 1. \tag{A.2}$$

Clearly, the cost function defined in equation (A.1), is convex in $\alpha_{(j,N)},\ \beta_{(j,N)}$. So if the globally optimal $\alpha_{(j,N)},\ \beta_{(j,N)}$ (found via equation (4.12)) lies outside the sufficient stability region, then the optimal solution satisfying this condition must lie on the boundary of this region, specified by

$$|\alpha_{(j,N)}| + |\beta_{(j,N)}| = 1. \tag{A.3}$$

The cost function, restricted to the boundary, can be written as:

$$\varepsilon_N = \sum \begin{array}{lll} (x_j[m] & - & \alpha_{(j,N)} x_j[m-N] \\ & - & (A + B\alpha_{(j,N)}) x_j[m-N+1])^2, \end{array} \tag{A.4}$$

where $(A, B) \in \{(1,-1),(1,1),(-1,-1),(-1,1)\}$, corresponding to the 4 linear segments of the boundary. Optimal $\alpha_{(j,N)}$ for this cost function is derived as the following by setting its derivative with respect to $\alpha_{(j,N)}$ to 0,

$$\alpha_{(j,N)} = \frac{r_{(0,N)} + Br_{(0,N-1)} - Ar_{(N,N-1)} - ABr_{(N-1,N-1)}}{r_{(N,N)} + B^2 r_{(N-1,N-1)} + 2Br_{(N,N-1)}}, \tag{A.5}$$

where $r_{(k,l)}$ is the correlation as defined in equation (4.13). $\alpha_{(j,N)}$ corresponding to each of the 4 segments is then limited to a range $[0,1]$ or $[-1,0]$ to ensure (A.3) is satisfied. Finally, amongst these 4 solutions, the $\alpha_{(j,N)}$ and the corresponding $\beta_{(j,N)} = A + B\alpha_{(j,N)}$, which results in the minimum error (A.1), is selected as the optimal stable gain factors.

Appendix B

Non-uniform quantization of gain factors

We first convert the gain factors α, β to polar coordinates, $r = \sqrt{\alpha^2 + \beta^2}$, $\theta = \tan^{-1}(\beta/\alpha)$, $\theta \in [-\pi, \pi]$, so that r captures the amplitude decay and θ effectively captures the non-integral part of the pitch period. This separation of information is favorable for entropy coding, and was also observed to be more robust to quantization error. Next, r, θ are independently scalar quantized non-uniformly, with $\mathbf{N}_r$, $\mathbf{N}_\theta$ levels, to give $\hat{r}$, $\hat{\theta}$ and $\hat{\alpha} = \hat{r}\cos(\hat{\theta})$, $\hat{\beta} = \hat{r}\sin(\hat{\theta})$. The non-uniform quantizers are learnt via k-means clustering algorithm using parameters obtained from a wide range of audio signals. The resulting constellation of the overall quantizer codebook is shown in Fig. B.1 for $\mathbf{N}_r = 10$ and $\mathbf{N}_\theta = 20$.

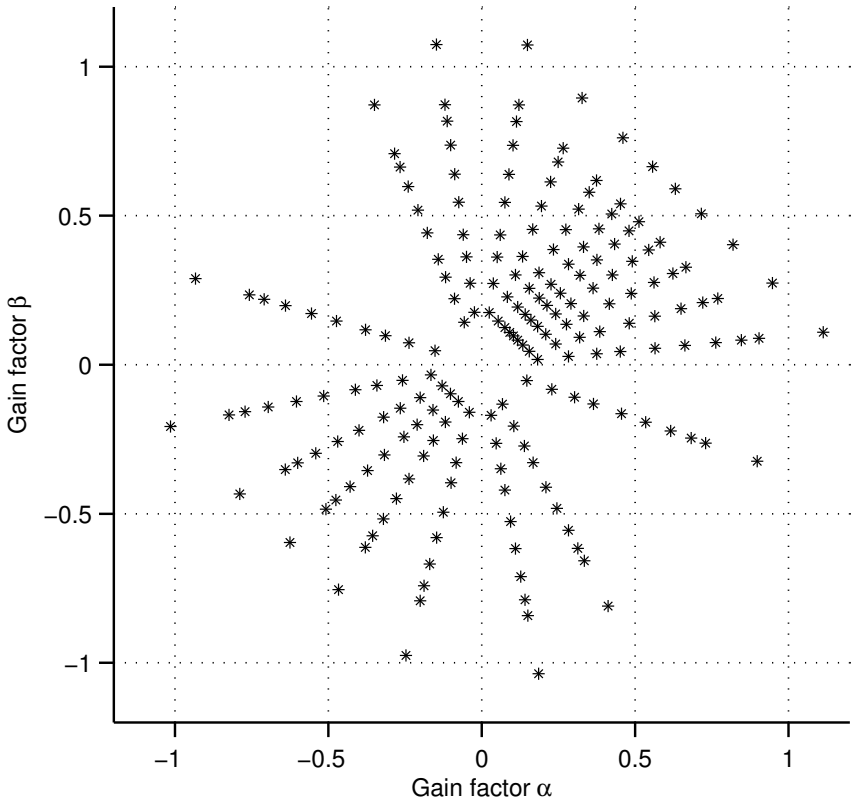

Figure B.1: Constellation of the overall gain quantizer codebook used for $\mathbf{N}_r = 10$ and $\mathbf{N}_\theta = 20$.

Appendix C

Encoding CLTP side information

Based on our assumption that audio signal is locally stationary, we exploit dependency between CLTP side information of consecutive frames via conditional coding. The first step for exploiting this inter-frame dependency is for each periodic component of current frame to be either matched to a periodic component of the previous frame or declared as a new periodic component. Let $m[i], i \in \{0, \ldots, P_n - 1\}$, denote the match index for each of the current periodic component. If the current periodic component is matched to a previous periodic component then, $m[i] \in \{0, \ldots, P_{n-1} - 1\}$, else $m[i] = \phi$. We also do not allow multiple current periodic components to map to the same previous periodic component. As each periodic component is characterized by its lag, the optimal mapping would minimize the following cost function,

$$J = \sum_{i=0}^{P_n - 1} \left\{ \begin{array}{ll} |N_{(i,n)} - N_{(m[i],n-1)}|, & \text{if } m[i] \neq \phi, \\ N_{(i,n)}, & \text{otherwise.} \end{array} \right\}. \tag{C.1}$$

Minimizing this cost function would effectively associate each current lag to the closest previous lag or leave it unmatched if it is very different from all previous lags. The match index is effectively providing the predicted current lag $\tilde{N}_{(i,n)} = N_{(m[i],n-1)}$, if $m[i] \neq \phi$, and $\tilde{N}_{(i,n)} = 0$, if $m[i] = \phi$. We find the mapping using a low complexity nearly-optimal technique, summarized below:

1. Create a matrix $\mathbf{D}$ of size $P_n \times (P_{n-1} + 1)$, with elements $D_{(i,j)}$ for $i = 0, \ldots, P_n - 1,\ j = 0, \ldots, P_{n-1}$ given as

$$D_{(i,j)} = \left\{ \begin{array}{ll} |N_{(i,n)} - N_{(j,n-1)}|, & \text{if } j \neq P_{n-1}, \\ N_{(i,n)}, & \text{otherwise.} \end{array} \right. \tag{C.2}$$

2. Identify $(i_{\min}, j_{\min}) = \arg\min_{\forall i,j} D_{(i,j)}$

3. Assign the match index for this $i_{\min}$ as,

$$m[i_{\min}] = \begin{cases} j_{\min}, & \text{if } j_{\min} \neq P_{n-1}, \\ \phi, & \text{otherwise.} \end{cases} \tag{C.3}$$

4. Set $D_{(i_{\min},j)} = \infty,\ j = 0, \ldots, P_{n-1}$.

5. If $j_{\min} \neq P_{n-1}$, set $D_{(i,j_{\min})} = \infty,\ i = 0, \ldots, P_n - 1$.

6. If $D_{(i,j)} \neq \infty, \forall i, j$ go to Step 2.

The lag bitstream sent to the decoder finally has, the match indices $m[i]$ as obtained above, the number of current lags P_n, and the current lags $N_{(i,n)}$ conditioned on its predicted lag $\tilde{N}_{(i,n)}$. The number of current lags, P_n, is encoded in a straightforward way using a single entropy coding table, $U_P[p],\ p = 1, \ldots, P_{\max}$. The probability density required to calculate this table was estimated using parameters obtained from a wide range of audio signals, with the resulting distribution of $\{0.06, 0.09, 0.14, 0.24, 0.47\}$. For encoding current lags, lowest average bits can be achieved if we use conditional entropy coding tables for every possible predicted lag. We would then have $N_{\max} - N_{\min} + 2$ tables, each of length $N_{\max} - N_{\min} + 1$, which would be enormous memory requirement, even for a very nominal $N_{\max} = 800, N_{\min} = 23$. On the other hand, if we use a single table (of length $2N_{\max} + 1$) to encode the lag prediction residue $\bar{N}_{(i,n)} = N_{(i,n)} - \tilde{N}_{(i,n)}$, we would require much smaller memory, but would result in higher average bits for encoding current lags. As a tradeoff between the two extremes we classify previous lags into one of $\mathbf{N}_N$ groups, with each group having its own entropy coding table for the lag prediction residue $\bar{N}_{(i,n)}$. This approach requires only $\mathbf{N}_N$ tables, each of length $2N_{\max} + 1$, thus keeping the memory requirement under check, but also incorporates conditional coding aspect to reduce the average bits required for encoding current lags. To create these $\mathbf{N}_N$ clusters, we use a tree-pruning approach, where we first start with $N_{\max} - N_{\min} + 2$ conditional entropy coding tables corresponding to every possible predicted lag, then we iteratively merge two of the existing tables which result in least increase in average bits required for encoding all lags, and finally stop this merging process when we have the desired number of clusters. During this process we also keep track of which predicted lag's conditional entropy coding tables were merged into each cluster and this information is stored as the cluster indexing table $I[p] \in \{1, \ldots, \mathbf{N}_N\}$, for $p = 0, N_{\min}, \ldots, N_{\max}$. We denote by, $U_N[q, p],\ q = 1, \ldots, \mathbf{N}_N,\ p = -N_{\max}, \ldots, 0, \ldots, N_{\max}$, the final $\mathbf{N}_N$ conditional entropy coding tables of the lag prediction residue. Note that all the probability densities required to calculate these tables were estimated using parameters obtained from a wide range of audio signals. The resulting indexing table

$I[p]$, and the resulting cluster wise probability densities of lag prediction residues, for $\mathbf{N}_N = 10$ is shown in Fig. C.1 and Fig. C.2. In the final lag bitstream we also optimize transmission of match indices, wherein instead of explicitly sending match indices, for each of the previous lag, we send a bit indicating if there is a matched current lag or not, and if this bit is set, then we send following this bit its corresponding lag prediction residue. This information is denoted as $B_j,\ j = 0, \ldots, P_{n-1} - 1$, and defined as:

$$B_j = \begin{cases} 0, & \text{if } m[i] \neq j\ \forall i, \\ 1, U_N[I[\tilde{N}_{(i,n)}], \bar{N}_{(i,n)}], & \text{if } m[i] = j. \end{cases} \tag{C.4}$$

We send the remaining unmatched lags of the current frame after all B_j. Thus the lag bitstream consists of $U_P[P_n], B_0, \ldots, B_{P_{n-1}-1}, U_N[I[0], \bar{N}_{(i,n)}], \forall\{i \mid m[i] = \phi\}$. Note that this encoding scheme reorders the periodic components of the current frame and effectively requires only 1 bit per periodic component to indicate its match index to previous periodic components.

The match index also provides predicted polar coordinates of the current gain factors as $\tilde{r}_{(i,n)} = \hat{r}_{(m[i],n-1)}, \tilde{\theta}_{(i,n)} = \hat{\theta}_{(m[i],n-1)}$, if $m[i] \neq \phi$, and $\tilde{r}_{(i,n)} = 0, \tilde{\theta}_{(i,n)} = 0$, if $m[i] = \phi$. The current polar coordinates of gain factors $\hat{r}_{(i,n)}, \hat{\theta}_{(i,n)}$ are coded separately conditioned on their predicted values $\tilde{r}_{(i,n)}, \tilde{\theta}_{(i,n)}$. Note that the number of possible predicted polar coordinates are $\mathbf{N}_r + 1$ and $\mathbf{N}_\theta + 1$, and since the nominal $\mathbf{N}_r$ and $\mathbf{N}_\theta$ are small, e.g., $\mathbf{N}_r = 10$ and $\mathbf{N}_\theta = 20$, using a conditional entropy coding table for every possible predicted value, results in manageable size of tables, $(\mathbf{N}_r + 1)\mathbf{N}_r$ and $(\mathbf{N}_\theta + 1)\mathbf{N}_\theta$. We denote by, $U_r[q, p],\ q = 0, \ldots, \mathbf{N}_r,\ p = 1, \ldots, \mathbf{N}_r$ and $U_\theta[q, p],\ q = 0, \ldots, \mathbf{N}_\theta,\ p = 1, \ldots, \mathbf{N}_\theta$, the conditional entropy coding tables of polar coordinates of the gain factors. The gain bitstream consists of $U_r[\tilde{r}_{(i,n)}, \hat{r}_{(i,n)}], U_\theta[\tilde{\theta}_{(i,n)}, \hat{\theta}_{(i,n)}], \forall i$, with elements arranged as per the new order of lags. Note that all the probability densities required to calculate these tables were estimated using parameters obtained from a wide range of audio signals. The resulting probability densities conditioned on each of the previous indices for gain magnitude and angle is shown in Fig. C.3 and Fig. C.4.

Finally the per-SFB prediction activation flags, $f_n[l]$, have to be sent to the decoder. Even these flags were observed to exhibit dependency between consecutive frames, thus we take this dependency into account by using the conditional probability density for each flag, $\mathbf{P}_l[q, p], l \in \{0, \ldots, L-1\}$, where $q \in \{0, 1\}$ indicates the state of the lth flag in previous frame, and $p \in \{0, 1\}$ indicates the state of the lth flag in current frame. Note that these densities were estimated using parameters obtained from a wide range of audio signals. The resulting conditional probabilities of these flags for all the bands in the operating bitrate range of 20 to 40 kbps is shown in Fig. C.5. Also we assume these flags to be independent of

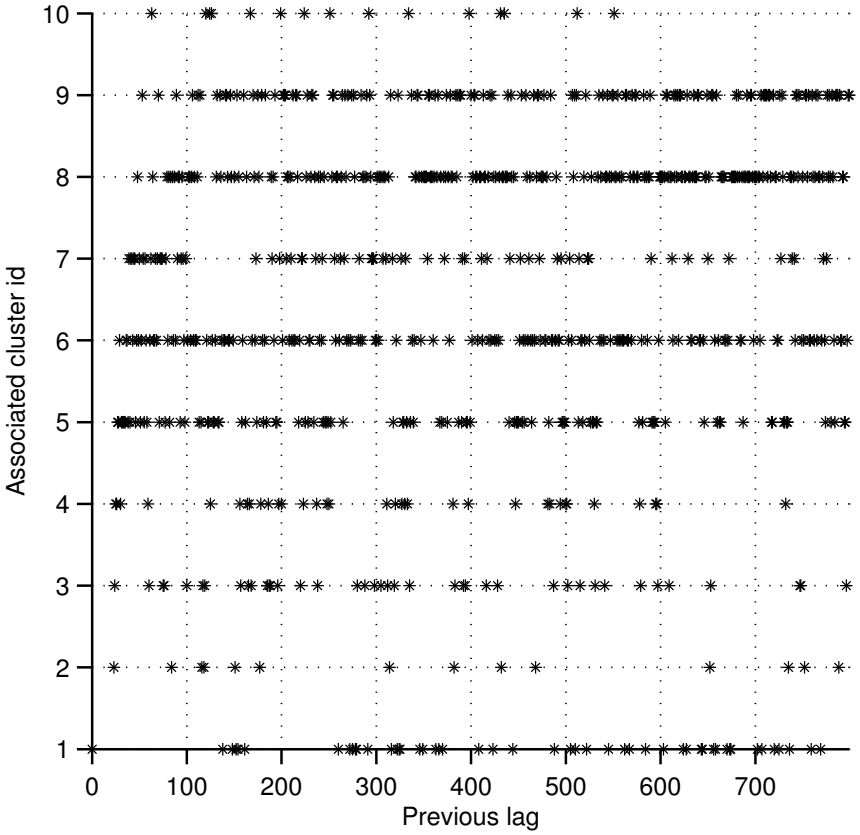

Figure C.1: The indexing table $I[p]$ used for $\mathbf{N}_N = 10$.

each other as we observed while estimating the probability densities that the joint density was very closely approximated by the product of the individual densities. Independently encoding these flags would require L bits, and for the AAC-LD encoder with $L = 36$, this will be a significant increase in side information rate. Instead to encode the flags with bits in line with probability of occurrence of the sequence of flags, we do arithmetic coding of the flags, wherein we require only $\lceil \prod_l \mathbf{P}_l[f_{n-1}[l], f_n[l]] \rceil$ bits to encode the flags. We employ an arithmetic coder using fixed-point precision of 15 bits as described in [46].

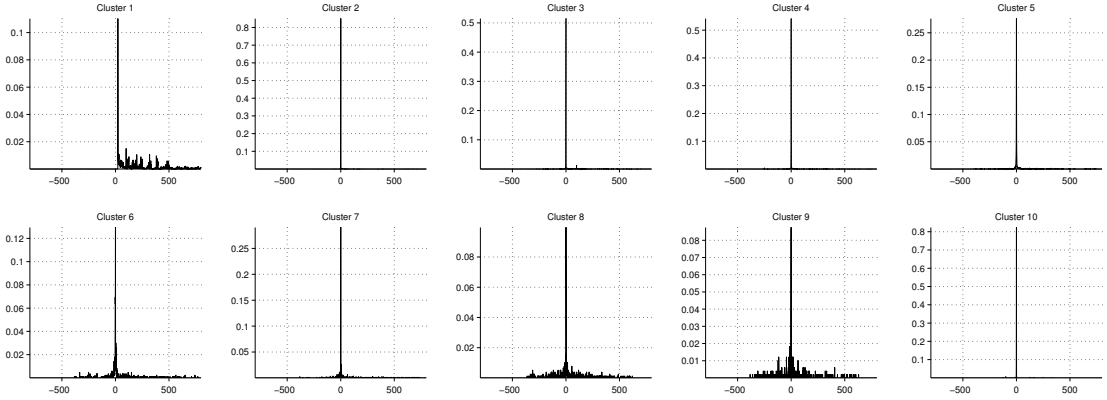

Figure C.2: The cluster wise lag prediction residue probability densities used for $\mathbf{N}_N = 10$. x-axis represents the lag prediction residue, y-axis represents the probabilities.

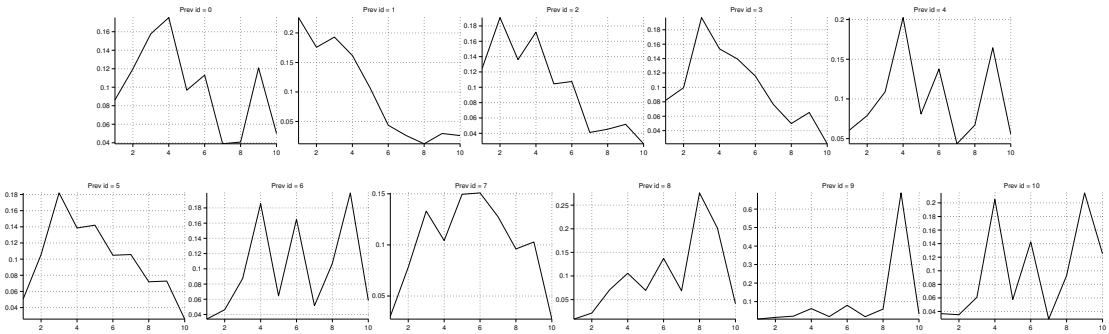

Figure C.3: The probability densities (conditioned on each of the previous indices of gain magnitude) used for $\mathbf{N}_r = 10$. x-axis represents current gain magnitude index, y-axis represents the probabilities.

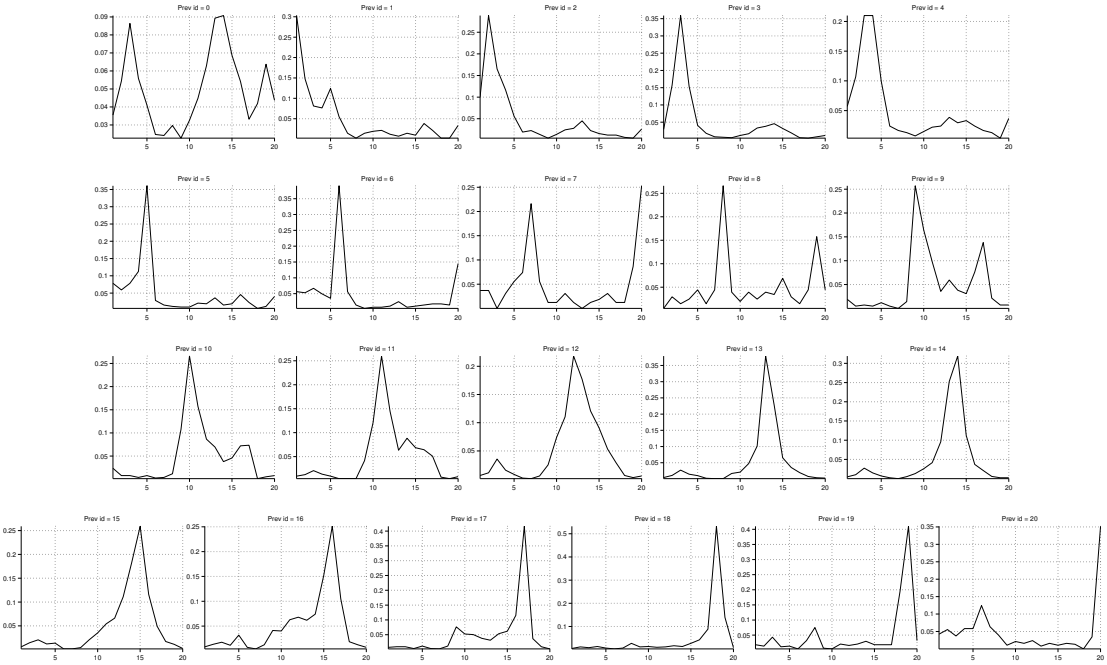

Figure C.4: The probability densities (conditioned on each of the previous indices of gain angle) used for $\mathbf{N}_\theta = 20$. x-axis represents current gain angle index, y-axis represents the probabilities.

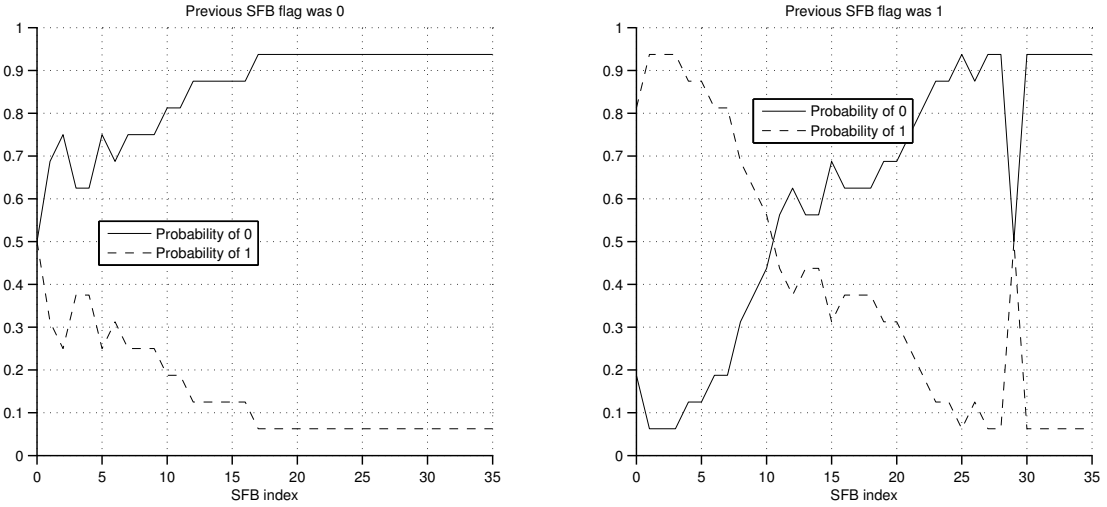

Figure C.5: The conditional probabilities of per-SFB prediction activation flags for all the bands.