Cascaded Long Term Prediction for Enhanced Compression of Polyphonic Audio Signals

Tejaswi Nanjundaswamy, *Member, IEEE*, and Kenneth Rose, *Fellow, IEEE*

***Abstract*—Audio compression systems exploit periodicity in signals to remove inter-frame redundancies via the long term prediction (LTP) tool. This simple tool capitalizes on the periodic component of the waveform by selecting a past segment as the basis for prediction of the current frame. However, most audio signals are polyphonic in nature, containing a mixture of several periodic components. While such polyphonic signals may themselves be 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 suboptimal. Instead of seeking a past segment that represents a "compromise" for incompatible component periods, we propose a more complex filter that predicts every periodic component of the signal from its immediate history, and this is achieved by cascading LTP filters, each corresponding to individual periodic component. We also propose a recursive "divide and conquer" technique to estimate parameters of all the LTP filters. We then demonstrate the effectiveness of cascaded LTP 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. Objective and subjective results for all the settings validate the effectiveness of the proposal on a variety of polyphonic signals.**

***Index Terms*—Audio compression, long term prediction, perceptual optimization, polyphonic signals.**

I. Introduction

A WIDE range of multimedia applications such as handheld playback devices, internet radio and television, online media streaming, gaming, and high fidelity teleconferencing heavily rely on advances in audio compression. Their success and proliferation has greatly benefited from current audio coders, including the MPEG Advanced Audio Coding (AAC) standard [1], which employ a modified discrete cosine transform (MDCT), whose decorrelating properties eliminate redundancies within a block of data. Still, there is potential for exploiting redundancies across frames, as audio content typically consists of naturally occurring periodic signals, examples of which include voiced parts of speech, music from string and wind instruments, etc. Note that inter-frame redundancy removal is highly critical in the cases of short frame coders such as the ultra low delay Bluetooth Subband Codec (SBC) [2], [3] and the MPEG AAC in low delay (LD) mode [4], as decorrelation within a frame is inefficient for such coders. For an audio signal with only one periodic component (i.e., a monophonic signal), inter-frame decorrelation can be achieved by the long term prediction (LTP) tool, which exploits repetition in the waveform by providing a segment of previously reconstructed samples, scaled appropriately, as prediction for the current frame. The resulting low energy residue is encoded at reduced rate. Typically, time domain waveform matching techniques that use a correlation measure are employed to find LTP parameters so as to minimize the mean squared prediction error. Parameter optimization for the LTP tool [5] in MPEG AAC was the focus of recent work by our group where a perceptual optimization technique was proposed to jointly optimize LTP parameters along with quantization and coding parameters, while explicitly accounting for the perceptual distortion and rate tradeoffs [6].

The existing LTP is well suited for signals containing a single periodic component, but this is not the case for general audio which often contains a mixture of multiple periodic signals. Typically, audio belongs to the class of polyphonic signals that 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 may itself be periodic, with overall period equaling the least common multiple (LCM) of all individual component periods, but even then the signal rarely remains stationary over such extended duration. Consequently, LTP resorts to a compromise by predicting from a recent segment that represents some tradeoff between incompatible component periods, with corresponding negative impact on its performance. 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. We propose to exploit these redundancies by 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. 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 a competent parameter estimation tech-

Manuscript received May 14, 2013; revised August 31, 2013; accepted January 01, 2014. Date of publication January 28, 2014; date of current version February 08, 2014. This work was supported by the National Science Foundation (NSF) under Grants CCF-0917230 and CCF-1320599 . The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wai-Yip (Geoffrey) Chan.

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: tejaswi@ece.ucsb.edu; rose@ece.ucsb.edu).

Digital Object Identifier 10.1109/TASLP.2014.2303292

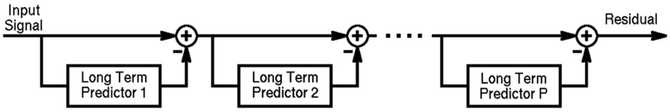


Fig. 1. Illustration of a "Cascaded Long Term Prediction" (CLTP) filter.

nique, and even more so for coders such as MPEG AAC, where perceptual distortion criteria should 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 Bluetooth SBC with backward adaptive parameter estimation. For MPEG AAC, we employ CLTP in two modes, one with mostly backward adaptive parameter estimation, and other with fully forward adaptive parameter estimation. In both AAC modes, the parameter estimation is adapted to account for the perceptual distortion criteria. 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 in a highly restricted setting, where CLTP is performed only on its first subband, have appeared in [7]. Preliminary work on extending this approach to MPEG AAC, without regard to decoder complexity has appeared in [8]. Historically, LTP has been considered since the introduction of predictive coding for speech [9]. A brief review of this LTP related prior work can be found in Section II-C. Note that the underlying notion of cascading filters has itself been employed in applications of spectral estimation for sums of sinusoids [10]. Deeper consideration of CLTP and the underlying polyphonic prediction problem points out relation to special cases of the source-separation problem, and surveying literature in this area revealed a similar construct employed to estimate fundamental frequencies in mixed speech sources [11]. An insightful recent paper [12] is motivated by the observation that a cascade of short and long term predictors yields a high order but sparse overall filter, and provides conjectures and experimental results on the optimization and use of general high-order sparse predictors in audio processing.

This paper is structured as follows: Background on Bluetooth SBC, MPEG AAC, LTP and the MPEG AAC LTP tool is provided in Section II. The polyphonic signal prediction problem is formulated in Section III. The proposed CLTP technique is introduced in Section IV. The proposed recursive CLTP parameter estimation technique is described in Section V. Specialization and derivations for enhancing Bluetooth SBC and MPEG AAC are presented in Section VI. Results are presented in Section VII, and the paper concludes in Section VIII.

II. Background

This section provides background information on the ultra low delay Bluetooth SBC, the perceptual audio coding standard of MPEG AAC in LD mode, the long term prediction technique and how it has been integrated in the MPEG AAC standard. Note that although the paper will specify how to incorporate our proposed technique into these two standards, the underlying approach is general and can easily be extended to other audio coders.

A. Bluetooth SBC

The Bluetooth Sub-band Codec (SBC) [2], [3] 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 $x[m]$, $nBK \leq m < (n+1)BK$. 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 $c_n[b,k], 0 \leq b < B, 0 \leq k < K$. The analysis filter bank is similar to MPEG Layer 1-3 [13], but with filter order of $10B$ (using $9B$ samples of history for B samples of input at time). The block of K samples in each sub-band is then quantized adaptively to minimize the quantization MSE. The effective scalefactor $s_n[b], 0 \leq b < B$ for each subband is sent to the decoder as side information along with the quantized data. The decoder operations are an appropriate reversal of encoder operations. Note that the analysis and synthesis filter banks together introduce a delay of $(9B+1)$ samples.

B. 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 B frequency bands (known as scale-factor bands or SFBs) such that all the coefficients in a band are quantized to, $\hat{c}_n[k]$, using the same scaled version of the generic AAC quantizer. The scaling factor (SF), $s_n[b]$, and the Huffman codebook (HCB), $h_n[b]$, used to encode the quantized data, control the rate and distortion for each SFB b. The SFs and HCBs are sent to the decoder as side information along with the quantized data. The decoder operations are an appropriate reversal of encoder operations. Selection of SFs and HCBs in the encoder is done to minimize the perceptual distortion, given as maximum over SFBs of quantization noise to masking threshold ratio (MNMR),

$$\mathcal{D}_n = \max_{0 \leq b < B} \frac{\sum_{k \in \text{SFB } b} (c_n[k] - \hat{c}_n[k])^2}{\mu_n[b]}, \qquad (1)$$

where the masking threshold, $\mu_n[b]$, is provided by a psychoacoustic model. 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], [14]–[17]). 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], [14]. 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.

C. Long Term Prediction

Exploiting long term correlations has been well known since the advent of predictive coding for speech [9] via the technique 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

$$H(z) = 1 - \sum_{k=0}^{T-1} \beta_k z^{-N+k}, \tag{2}$$

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 improving compression performance of voiced segments in speech, have been extensively studied. A thorough review and analysis of various structures for pitch prediction filters is available in [18]. Backward adaptive parameter estimation was proposed in [19] for low-delay speech coding, but forward adaption was found to be advantageous in [20]. Different techniques to efficiently transmit the filter information were proposed in [21] and [22]. The idea of using more than one filter tap (i.e., $T > 1$ in equation (2)) was originally conceived to approximate fractional delay [23], but has been found to have broader impact in [24]. Techniques for reducing complexity of parameter estimation have been studied in [25] and [26]. For a review of speech coding work in modeling periodicity, see [27]. Note that employing inter-frame prediction while transmitting content over unreliable networks may cause significant error propagation, as frame loss will result in drift between encoder and decoder. Error propagation due to inter-frame prediction is a well known problem which has been studied extensively (e.g., see [28]) and we do not include it as it is beyond the scope of this work.

D. Long Term Prediction Tool in MPEG AAC Standard

Long term prediction has also been proposed as an optional tool for the audio coding standard of MPEG AAC, specifically targeted at the LD mode. This subsection builds on the notation introduced for the MPEG AAC standard in Section II-B 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, 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 G_n, which is selected from a set of 8 values. Thus the LTP analysis filter is of the form

$$H_{\text{LTP}}(z) = 1 - G_n z^{-L_n}, \tag{3}$$

and the prediction of the current frame is denoted as

$$\tilde{x}_n[m] = G_n \hat{x}[m + nK - L_n], 0 \leq m < 2K. \tag{4}$$

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

$$\varepsilon = \sum_{m=0}^{2K-1} (x[m+nK] - \tilde{x}_n[m])^2. \tag{5}$$

For a given L_n, G_n is optimized by setting to 0, the partial derivatives of ε with respect to G_n. The best L_n is selected as the one which minimizes ε, while using the optimal G_n for every candidate L_n. This selection procedure simplifies to the following:

$$L_n = \underset{L \in [K, 3K)}{\arg\max} \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-L]}{\sqrt{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-L]}} \tag{6}$$

$$G_n = \frac{\sum_{m=0}^{2K-1} x[m+nK]\hat{x}[m+nK-L_n]}{\sum_{m=0}^{2K-1} \hat{x}^2[m+nK-L_n]} \tag{7}$$

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 $f_n[b]$. This flag is set whenever the prediction residue energy is lower than the signal energy in the band,

$$f_n[b] = \begin{cases} 1, & \text{if} \sum_{k \in \text{SFB } b} e_n^2[k] < \sum_{k \in \text{SFB } b} c_n^2[k] \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

A global flag 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

$$R_n = \frac{1}{6} \sum_{b=0}^{B-1} 10 \log_{10} \left[\frac{\sum_{k \in \text{SFB } b} c_n^2[k]}{\min\left(\sum_{k \in \text{SFB } b} c_n^2[k], \sum_{k \in \text{SFB } b} e_n^2[k] \right)} \right] K_b \tag{9}$$

where K_b is the number of coefficients in the SFB b. 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

$$F_n = \begin{cases} 1, & \text{if } R_n > \text{LTP side information rate} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The final coefficients $\forall k \in \text{SFB } b$ are given by

$$q_n[k] = \begin{cases} e_n[k], & \text{if } f_n[b] = 1 \text{ and } F_n = 1, \\ c_n[k], & \text{otherwise.} \end{cases} \tag{11}$$

These coefficients are quantized and coded via the technique described in Section II-B. The decoder receives as additional

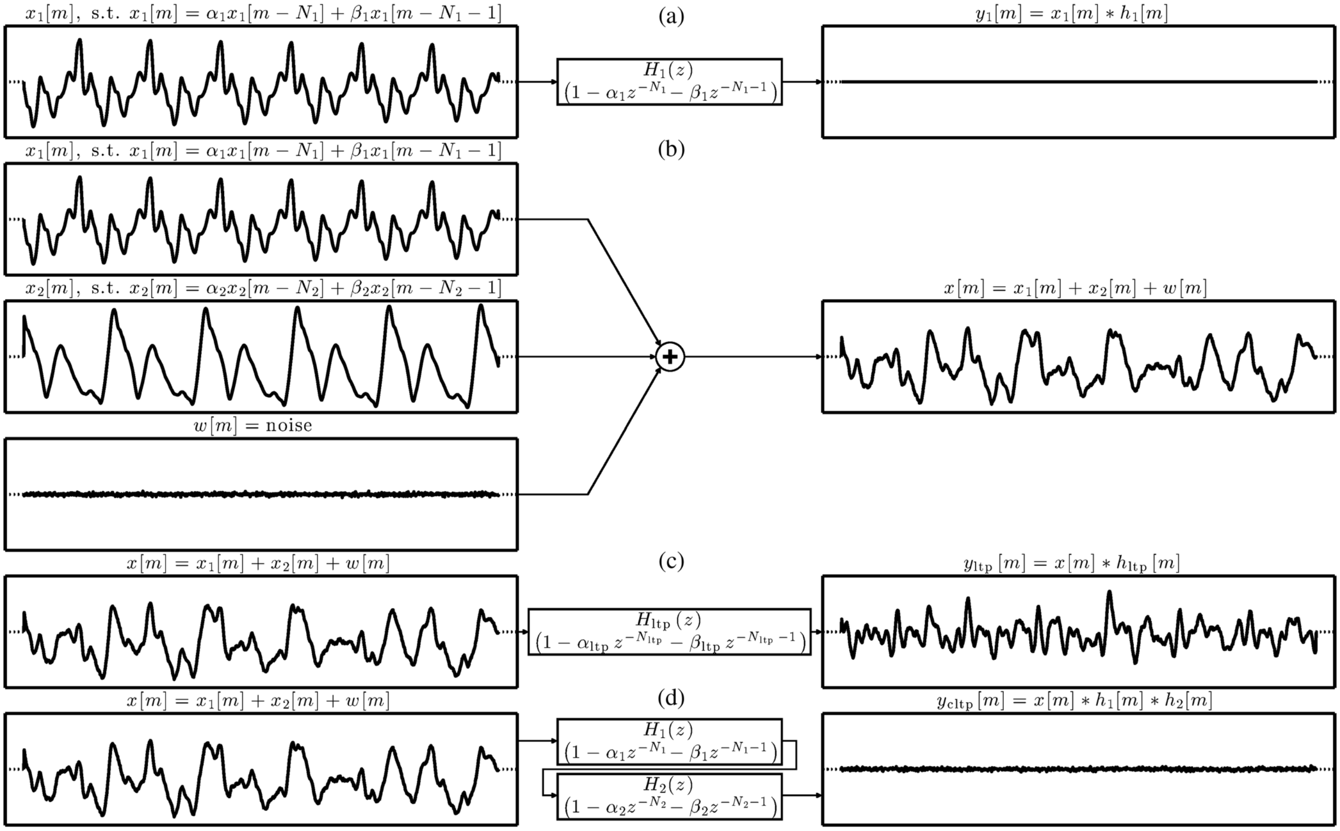

$x_1[m],\ \text{s.t. } x_1[m] = \alpha_1 x_1[m-N_1] + \beta_1 x_1[m-N_1-1]$
(a)
$H_1(z)$
$(1-\alpha_1 z^{-N_1} - \beta_1 z^{-N_1-1})$
$y_1[m] = x_1[m] * h_1[m]$
$x_1[m],\ \text{s.t. } x_1[m] = \alpha_1 x_1[m-N_1] + \beta_1 x_1[m-N_1-1]$
(b)
$x_2[m],\ \text{s.t. } x_2[m] = \alpha_2 x_2[m-N_2] + \beta_2 x_2[m-N_2-1]$
$x[m] = x_1[m] + x_2[m] + w[m]$
$w[m] = \text{noise}$
$x[m] = x_1[m] + x_2[m] + w[m]$
(c)
$y_{\text{ltp}}[m] = x[m] * h_{\text{ltp}}[m]$
$H_{\text{ltp}}(z)$
$(1-\alpha_{\text{ltp}} z^{-N_{\text{ltp}}} - \beta_{\text{ltp}} z^{-N_{\text{ltp}}-1})$
$x[m] = x_1[m] + x_2[m] + w[m]$
(d)
$y_{\text{cltp}}[m] = x[m] * h_1[m] * h_2[m]$
$H_1(z)$
$(1-\alpha_1 z^{-N_1} - \beta_1 z^{-N_1-1})$
$H_2(z)$
$(1-\alpha_2 z^{-N_2} - \beta_2 z^{-N_2-1})$

Fig. 2. Illustration of various LTP filters. (a) Output of simple LTP filtering for an example periodic signal, $x_1[m]$ with $\alpha_1 = 0.59, \beta_1 = 0.39, N_1 = 148$. (b) An example polyphonic signal with 2 periodic components ($x_1[m]$ as in (a) and $x_2[m]$ with $\alpha_2 = 0.76, \beta_2 = 0.27, N_2 = 193$) 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).

side information $L_n, G_n, F_n, f_n[b], \forall b$, which it uses them to reverse the LTP operation.

III. Polyphonic Signals and Problem Setting

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

A periodic signal with period N is characterized by the relation $x[m] = x[m-N]$. However, this strict definition is overly limiting and must be relaxed in practice. Many naturally occurring sounds exhibit amplitude decay/growth (e.g., piano/guitar chord) necessitating a coefficient that may differ from one, i.e., $x[m] = \alpha x[m-N]$. We also need to account for non-integer pitch periods while operating in the discrete domain. For simplicity, we approximate a non-integer pitch period via linear interpolation, leading to an approximate model for naturally occurring periodic signals, $x[m] = \alpha x[m-N] + \beta x[m-N-1]$, where α and β implement both the amplitude changes and non-integral pitch period. Note that although this model is a simplistic approximation of non-integer pitch periods, it is sufficient to convey the main points of the paper. The evaluation of other techniques to approximate non-integer pitch periods will be pursued in future work. A polyphonic audio signal comprising a mixture of such periodic signals, can be modeled as

$$x[m] = \sum_{i=0}^{P-1} x_i[m] + w[m], \tag{12}$$

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

$$x_i[m] = \alpha_i x_i[m-N_i] + \beta_i x_i[m-N_i-1]. \tag{13}$$

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:

$$H_0(z) = 1 - \alpha_0 z^{-N_0} - \beta_0 z^{-N_0-1}, \tag{14}$$

whose prediction error $e[m]$ is dependent only on the noise or aperiodic component $w[m]$. An illustration of simple LTP filtering is provided in Fig. 2(a) for an example periodic signal (absent noise). The LTP tool in MPEG AAC standard (described in Section II-D) can also predict well 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 to optimize the prediction.

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

$$H_{\text{ltp}}(z) = 1 - \alpha_{\text{ltp}} z^{-N_{\text{ltp}}} - \beta_{\text{ltp}} z^{-N_{\text{ltp}}-1}, \tag{15}$$

where N_{ltp} is the lag that minimizes the mean squared prediction error, within the history available for prediction. Consequently, 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 (when it exists) 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 despite the incompatible periods. The suboptimality of simple LTP filtering a polyphonic signal is illustrated in Fig. 2(c). Note that this limitation is due to the overly simplistic prediction model of LTP, and it was confirmed in [6] that perceptually motivated parameter optimization of the LTP tool in MPEG AAC standard, while beneficial for monophonic signals, did not provide significant performance improvement for complex polyphonic signals.

IV. Cascaded Long Term Prediction

If we apply the LTP filter $H_0(z)$ (in (14)) designed for a signal with single periodic component to a polyphonic signal (12) where $P > 1$, the filter output is expressed as

$$\begin{aligned} e_0[m] &= x[m] - \alpha_0 x[m-N_0] - \beta_0 x[m-N_0-1] \\ &= \sum\nolimits_{i=0}^{P-1} x'_i[m] + w'[m], \end{aligned} \tag{16}$$

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, the following straightforward algebra verifies that all the remaining components, $x'_i[m]$, exhibit the same periodicity as $x_i[m]$, i.e., the same period as prior to filtering:

$$\begin{aligned} x'_i[m] &= x_i[m] - \alpha_0 x_i[m-N_0] - \beta_0 x_i[m-N_0-1] \\ &= \alpha_i x_i[m-N_i] + \beta_i x_i[m-N_i-1] \\ &\quad - \alpha_0(\alpha_i x_i[m-N_0-N_i] \\ &\quad + \beta_i x_i[m-N_0-N_i-1]) \\ &\quad - \beta_0(\alpha_i x_i[m-N_0-1-N_i] \\ &\quad + \beta_i x_i[m-N_0-1-N_i-1]) \\ &= \alpha_i(x_i[m-N_i] - \alpha_0 x_i[m-N_0-N_i] \\ &\quad - \beta_0 x_i[m-N_0-1-N_i]) \\ &\quad + \beta_i(x_i[m-N_i-1] - \alpha_0 x_i[m-N_0-N_i-1] \\ &\quad - \beta_0 x_i[m-N_0-1-N_i-1]) \\ &= \alpha_i x'_i[m-N_i] + \beta_i x'_i[m-N_i-1]. \end{aligned} \tag{17}$$

In other words, the output of filter H_0 is, in fact, a polyphonic signal with $P-1$ periodic components. It follows recursively that the cascaded LTP filter

$$H_c(z) = \prod_{i=0}^{P-1}(1 - \alpha_i z^{-N_i} - \beta_i z^{-N_i-1}) \tag{18}$$

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 (in this case both) periodic components, is provided in Fig. 2(d). The CLTP filter of (18), appropriately designed, forms the basis of our proposal to improve compression efficiency in polyphonic audio signals. Note that this fundamental approach of cascading LTP filters is applicable more generally and can be used with linear LTP filters with any number of taps.

V. Basic CLTP Parameter Estimation

In this section we derive 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 re-evaluated every time the parameters undergo updates, resulting in prohibitive computational complexity. Note that a related problem is analysis of mixtures of periodic components, for which many full fledged solutions have been proposed, including [29] and [30]. These techniques are involved and not fully applicable to our problem at hand of simple but effective prediction of a frame of polyphonic audio signal well, e.g., [29] involves having to manage tradeoff between time and frequency resolution; while [30] involves unnecessary (to us here) estimation of the number of harmonics of each periodic component. Thus, we propose a practical "divide and conquer" recursive estimation technique.

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

$$\bar{H}_j(z) = \prod_{\forall i, i \neq j}(1 - \alpha_i z^{-N_i} - \beta_i z^{-N_i-1}) \tag{19}$$

and the corresponding residue

$$X_j(z) = X(z)\bar{H}_j(z). \tag{20}$$

We next optimize the parameters of the jth filter $H_j(z) = 1 - \alpha_j z^{-N_j} - \beta_j z^{-N_j-1}$ 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

$$\begin{bmatrix} \alpha_{(j,N)} \\ \beta_{(j,N)} \end{bmatrix} = \begin{bmatrix} r_{(N,N)} & r_{(N+1,N)} \\ r_{(N+1,N)} & r_{(N+1,N+1)} \end{bmatrix}^{-1} \begin{bmatrix} r_{(0,N)} \\ {}_{(0,N+1)} \end{bmatrix} \tag{21}$$

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

$$r_{(k,l)} = \sum_{m=Y_{\text{start}}}^{Y_{\text{end}}} x_j[m-k]x_j[m-l], \tag{22}$$

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 (especially when predicting an entire frame of data as described in Section VI-A) we restrict $\alpha_{(j,N)}, \beta_{(j,N)}$ solutions to only those that satisfy the sufficient stability condition

$$|\alpha_{(j,N)}| + |\beta_{(j,N)}| \leq 1. \tag{23}$$

For details of this stability criterion and further analysis of LTP filter stability, please refer to [31]. For details on optimization procedure satisfying stability criteria, please refer to [32]. Given $\alpha_{(j,N)}, \beta_{(j,N)}$, the optimal N_j is found as

$$N_j = \underset{N \in [N_{\min}, N_{\max}]}{\arg\min} \sum_{m=Y_{\text{start}}}^{Y_{\text{end}}} \big(x_j[m] - \alpha_{(j,N)} x_j[m-N] - \beta_{(j,N)} x_j[m-N-1]\big)^2, \tag{24}$$

where $N_{\min}, N_{\max}$ are the lower and upper boundaries of the period search range. In equations (22) and (24), 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 non-convex in the pitch periods N_j, a globally optimal solution is not guaranteed.

VI. Enhancing Real World Codecs With CLTP

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

A. 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.

B. Integration With Bluetooth SBC

The Bluetooth SBC (described in Section II-A) 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 V 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: $\hat{x}[m], nBK - (9B+1) - M \leq m < nBK - (9B+1)$. The gap of $(9B+1)$ is due to the filter bank delay. We employ CLTP to predict samples of the current frame along with the samples required for the analysis filter bank history, which is, $\tilde{x}[m], nBK - 9B \leq m < (n+1)BK$. The CLTP filter of (18) is updated once per frame as $H_n(z)$ with parameters $P_n, N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}, \forall i \in \{0, \ldots, P_n - 1\}$. 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 V, 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 (22), (24) and using the reconstructed samples, $\hat{x}[m]$. This parameter estimation is then repeated to find CLTP filters for each $P_n \in \{1, \ldots, P_{\max}\}$ and the P_n which minimizes the mean squared prediction error is selected. The predicted samples required to calculate this error are generated via the 'looped' prediction method described in Section VI-A. 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 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:

$$f_n[b] = \begin{cases} 1, & \text{if} \sum\limits_{k=0}^{K-1} e_n^2[b,k] < \sum\limits_{k=0}^{K-1} c_n^2[b,k] \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

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

$$q_n[b,k] = \begin{cases} e_n[b,k], & \text{if} f_n[b] = 1, \\ c_n[b,k], & \text{otherwise.} \end{cases} \tag{26}$$

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 periodic components P_n, and the flags $f_n[b]$. An illustration of integrating CLTP with a generic audio coder is provided in Fig. 3 (wherein for Bluetooth

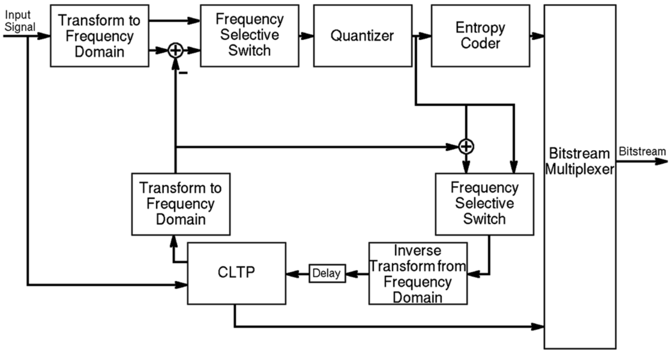

Input Signal
Transform to Frequency Domain
Frequency Selective Switch
Quantizer
Entropy Coder
Bitstream Multiplexer
Bitstream
Transform to Frequency Domain
Frequency Selective Switch
CLTP
Delay
Inverse Transform from Frequency Domain

Fig. 3. Illustration of the proposed integration of CLTP with an audio coder operating in frequency domain.

SBC, the transform to frequency domain and inverse transform from frequency domain corresponds to the subband analysis and synthesis filters, respectively).

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.

C. 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 reduce 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.

1) Estimation of Backward Adaptive Parameters in the Encoder: For a tentative number of periodic components P_n in frame n, we estimate a CLTP filter (18), denoted here as $H'_n(z)$ with parameters $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}$ $\forall i \in 0, \ldots, P_n - 1$, backward adaptively to minimize MSE via the recursive technique described in Section V with the limits, $Y_{\text{start}} = (n-2)K$, $Y_{\text{end}} = nK - 1$, in the correlation and prediction error measures (22), (24) while using the reconstructed samples, $\hat{x}[m]$. Amongst the parameters estimated at this stage, the pitch periods for the given P_n are final and not adjusted further as they are physical property of the signal waveform and independent of perceptual considerations. 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 CLTP filter $H'_n(z)$, we generate the prediction residue samples by filtering the reconstructed samples $\hat{x}[m]$ with $H'_n(z)$. 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 $\hat{\mu}_{n-2}[b], 0 \leq b < B$ to decide the per-SFB prediction activation flag $f_n[b]$, as

$$f_n[b] = \begin{cases} 1, & \text{if } \sum\limits_{k \in \mathrm{SFB}\, b} \hat{c}^2_{n-2}[k] > \hat{\mu}_{n-2}[b] \text{ and} \\ & \sum\limits_{k \in \mathrm{SFB}\, b} e^2_{n-2}[k] < \sum\limits_{k \in \mathrm{SFB}\, b} \hat{c}^2_{n-2}[k] \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

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.

2) Perceptually Motivated Joint CLTP Parameter Refinement and Core AAC Parameter Estimation: The gains $\alpha_{(i,n)}, \beta_{(i,n)}$ 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 $G_{(i,n)}$ to form the final CLTP filter

$$H_n(z) = \prod_{i=0}^{P_n - 1} \left(1 - G_{(i,n)} \alpha_{(i,n)} z^{-N_{(i,n)}} - G_{(i,n)} \beta_{(i,n)} z^{-N_{(i,n)} - 1}\right), \tag{28}$$

where $G_{(i,n)}$ is quantized to one of $\mathcal{N}_G$ 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 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 $G_{(i,n)} \forall i$, are sent to the decoder using $\lceil \log_2(\mathcal{N}_G) \rceil P_n$ bits.

A straightforward way to estimate all the remaining parameters would be to evaluate for every combination of CLTP parameters $P_n, G_{(i,n)}$ and 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 $\mathcal{N}_G = 4$, 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 [6] for the MPEG AAC LTP tool. The overall approach is summarized in Algorithm 1.

Algorithm 1

```
1:  for all $P_n \in \{1, \ldots, P_{\max}\}$
2:      Estimate preliminary CLTP filter $H'_n(z)$.
3:      Estimate per-SFB flags $f_n[b]$ as given in (27).
4:      for every possible combination of $G_{(i,n)}, \forall i$
5:          Get $H_n(z)$ as in (28).
6:          Generate 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 VI-A.
7:          Transform predicted samples via MDCT to
            produce $K$ transform coefficients $\tilde{c}_n[k]$.
8:          Calculate per transform coefficient prediction
            residue as $e_n[k] = c_n[k] - \tilde{c}_n[k]$.
9:          Evaluate prediction MSE after considering the
            flags $f_n[b]$.
10:     end for
11: end for
12: Determine top $S$ survivors based on prediction MSE
    calculated in Step 9.
13: for all $S$ survivors
14:     Determine best SFs and HCBs to encode the
        prediction residue via TLS, and calculate the
        associated distortion for the given total rate (which
        includes the CLTP side information rate).
15: end for
16: Determine best SFs and HCBs to alternatively encode
    the original frame via TLS (i.e., CLTP is disabled or
    $F_n = 0$), and calculate the associated distortion for
    the given total rate.
17: Of the $S + 1$ above options choose the one that yields
    minimum distortion and encode the frame.
```

Note that controlling the number of survivors S enables controlling the tradeoff between complexity and performance. Fig. 3 also illustrates the proposed integration of CLTP with MPEG AAC, wherein the transform to frequency domain and inverse transform from frequency domain corresponds to MDCT and inverse MDCT, respectively.

The decoder receives F_n. If $F_n = 1$ it receives P_n, and $G_{(i,n)}, \forall i$, and estimates $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}, \forall i$, and $f_n[b], \forall b$. 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 VI-A. 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, to which the predicted transform coefficients are added whenever the flag $f_n[b]$ is set, to generate the reconstructed transform coefficients. The output signal is synthesized via inverse MDCT. If $F_n = 0$, standard AAC decoding procedure is followed.

3) Low Decoder Complexity Variant: Clearly in a backward adaptive setting, the decoder is of significantly higher complexity as it needs to replicate parameter estimation from previously reconstructed samples, as in the encoder. While this technique minimizes the side information rate, 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 the only additional step in the decoder is to synthesize the current frame prediction using the filter parameters received as side information. Note that in this approach we trade decrease in decoder complexity 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 B.

For a tentative number of periodic components P_n in frame n, we estimate a CLTP filter (18), denoted here as $H'_n(z)$ with parameters $N_{(i,n)}, \alpha_{(i,n)}, \beta_{(i,n)}$ $\forall i \in 0, \ldots, P_n - 1$, in an open loop to minimize MSE via the recursive technique described in Section V with the limits, $Y_{\text{start}} = nK, Y_{\text{end}} = (n+2)K - 1$, in the correlation and prediction error measures (22), (24) while using the original samples, $x[m]$. Note that as parameter estimation is forward adaptive, we use the original signal (uncorrupted by quantization error) to improve accuracy of estimated parameters, especially the pitch periods which are physical property of the signal waveform and not adjusted further for the given P_n.

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 VI-A, 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 Section VI-C2, CLTP gain factors need to be adjusted according to the perceptual distortion criteria as well. We tackle the highly non-convex problem of adjusting gain factors by limiting our search to a small discrete set of neighborhood around the preliminary estimate of gain factors by introducing a multiplicative gain factor $G_{(i,n)}$, which can take one of $\mathcal{N}_G$ levels, e.g., $\{0.5, 0.75, 1, 1.25\}$. The final gain factors $G_{(i,n)}\alpha_{(i,n)}$, $G_{(i,n)}\beta_{(i,n)}$ are then non-uniformly quantized to $\hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ as described in Appendix A for efficient encoding as side information. The final CLTP filter for the given P_n is, denoted as $H_n(z)$, with parameters $N_{(i,n)}, \hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ $\forall i \in 0, \ldots, P_n - 1$. We also retain per-SFB prediction activation flag, $f_n[b]$. Selection procedure of $G_{(i,n)}, f_n[b], \forall b$, and P_n follows very closely the procedure described in previous section, but with mandatory modifications for forward adaptation of all the parameters. The overall approach is summarized in Algorithm 2.

Algorithm 2

```
1:  for all $P_n \in \{1, \ldots, P_{\max}\}$
2:      Estimate preliminary CLTP filter $H'_n(z)$ in an open loop.
3:      for every possible combination of $G_{(i,n)}, \forall i$
4:          Quantize $G_{(i,n)}\alpha_{(i,n)}, G_{(i,n)}\beta_{(i,n)}$ to
            $\hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ and get $H_n(z)$.
5:          Generate 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 VI-A.
6:          Transform predicted samples via MDCT to
            produce $K$ transform coefficients $\tilde{c}_n[k]$.
7:          Calculate per transform coefficient prediction
            residue as $e_n[k] = c_n[k] - \tilde{c}_n[k]$.
8:          Estimate per-SFB flags $f_n[b]$ as,
            $f_n[b] = \begin{cases} 1, & \text{if } \sum_{k \in \text{SFB } b} c_n^2[k] > \mu_n[b] \text{ and } \sum_{k \in \text{SFB } b} e_n^2[k] < \sum_{k \in \text{SFB } b} c_n^2[k] \\ 0, & \text{otherwise.} \end{cases}$   (29)
9:          Evaluate prediction MSE after considering the
            flags $f_n[b]$.
10:     end for
11: end for
12: Determine top $S$ survivors based on prediction MSE
    calculated in Step 9.
13: for all $S$ survivors
14:     Encode prediction 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
        $f_n[b] \forall b$, as described in Appendix B, to calculate
        side information rate.
15:     Determine best SFs and HCBs to encode the
        prediction residue via TLS, and calculate the
        associated distortion for the given total rate (which
        includes the CLTP side information rate).
16: end for
17: Determine best SFs and HCBs to alternatively encode
    the original frame via TLS (i.e., CLTP is disabled or
    $F_n = 0$), and calculate the associated distortion for
    the given total rate.
18: Of the $S + 1$ above options choose the one that yields
    minimum distortion and encode the frame.
```

The low complexity decoder first receives F_n as the side information. If $F_n = 1$ it receives P_n, $N_{(i,n)}, \hat{\alpha}_{(i,n)}, \hat{\beta}_{(i,n)}$ $\forall i$, and $f_n[b], \forall b$. 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 VI-A. These

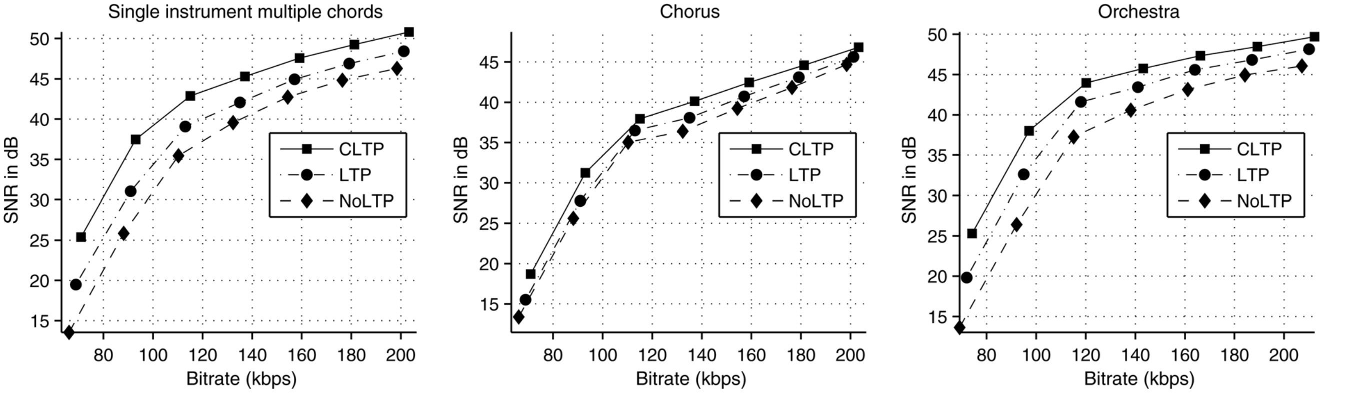


Fig. 4. Signal to quantization noise ratio versus bit-rates of the competing coders for Bluetooth SBC experiments, evaluated and averaged over files in each of the three classes of dataset.

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 $f_n[b]$ is set, to generate the reconstructed transform coefficients, from which the output signal is synthesized via inverse MDCT. If $F_n = 0$, standard AAC decoding procedure is followed.

VII. 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 44.1/48 kHz 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 resulting subset is:

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

A. 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
- SBC with the proposed CLTP.

The SBC is operated at $B = 4$ subbands and $K = 16$ number of samples in each subband; and we restricted CLTP to $P_{\max} = 5$ maximum number of periodic components. The lag search range in equation (24) is $N_{\min} = 99$, $N_{\max} = 799$ for both LTP and CLTP, corresponding to fundamental frequencies of 55.125/60 Hz to 441/480 Hz for sampling rates of 44.1/48 kHz. The 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 and obtained by setting $P_{\max} = 1$ in the system described in Section VI-B, which results in two-tap LTP filters. This mode is included in our experiments to specifically demonstrate the performance improvements of using CLTP over LTP.

TABLE I
Prediction Gains and Reconstruction Gains in dB for the Bluetooth SBC Experiments

Filename	Prediction gains		Reconstruction gains	
	LTP over NoLTP	CLTP over LTP	LTP over NoLTP	CLTP over LTP
Piano	5.3	9.4	4.6	7.1
Guitar	9.4	3.5	6.6	1.9
Bells	5.7	10.8	5.2	9.2
Mfv	7.8	6.2	8	5.5
Mozart	7.1	4.8	6.2	3.9
Quartet	2.5	3.7	2	3.1
Average	6.3	6.4	5.4	5.1

1) 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 I. For a thorough evaluation of SNR gains, we generate results given in Table I using longer duration files spanning a total of around 150 seconds (instead of 4 seconds each) over the six files. The table shows that CLTP provides truly major prediction gains of on the average 6.4 dB over LTP, which translate to substantial compression performance gains of on the average 5.1 dB. The table also shows that these gains came on top of already substantial 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. The plots clearly demonstrate that substantial gains are provided by CLTP for a wide range of polyphonic signals at various rates.

2) Subjective Evaluation Results: A subjective evaluation of all the three competing Bluetooth SBC coders, operating at around 80kbps, was conducted via MUSHRA listening tests [33]. We operate at low rates to emphasize the improvements provided by our proposed scheme. 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 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 (which acts as another anchor (anc2),

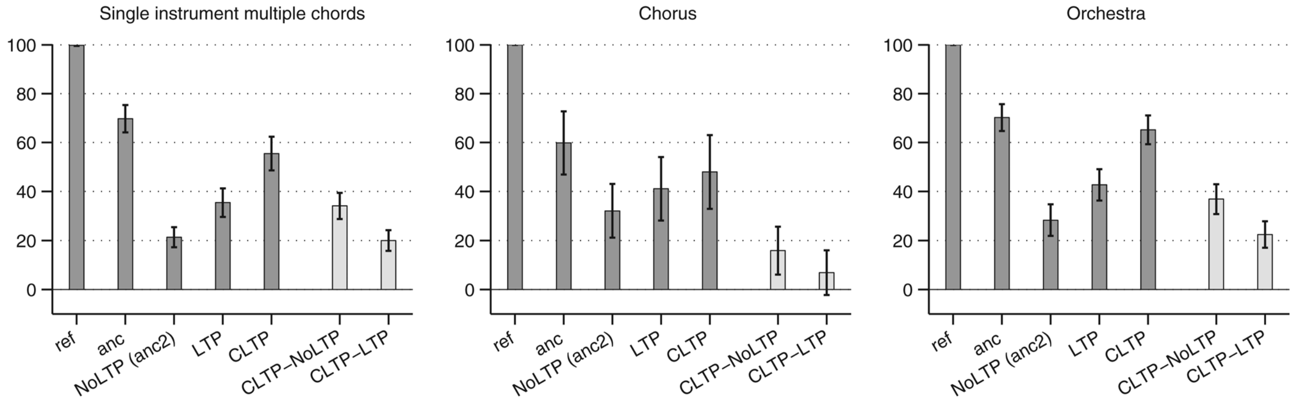


Fig. 5. MUSHRA listening test average scores with 95% confidence intervals, comparing Bluetooth SBC encoders with no LTP, LTP and proposed CLTP, for the three classes of dataset. The dark grey bars are absolute scores and the light grey bars are difference scores.

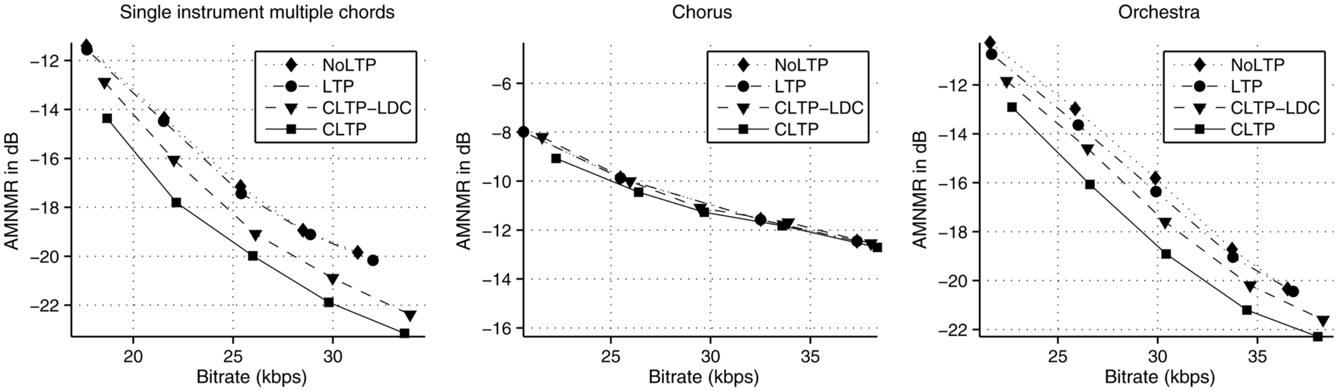


Fig. 6. Average per-frame distortion at various bit-rates of the different coders for the MPEG AAC experiments, evaluated and averaged over files for each of the three classes of dataset. AMNMR = average MNMR.

representing a well known quality level), LTP and the proposed CLTP. The MUSHRA test results with average and 95% confidence intervals for absolute and difference scores are shown in Fig. 5 for the three classes of the test dataset. The difference scores were included to show consistency of *preference* by listeners despite variation in scoring habits (conservative vs. lenient). They highlight the degree of preference of CLTP over no LTP, and over LTP. Note that the 95% confidence interval for the difference score of CLTP over LTP for the chorus file extends somewhat to negative territory, indicating a less consistent performance, 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. For all other files, the subjective evaluation results confirm that the significant gains in objective criteria translate to substantial subjective quality improvements.

B. Results for MPEG AAC

We compare the following four AAC LD coders (i.e. with frame length $2K = 1024$) 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 use lag search range of $N_{\min} = 22, N_{\max} = 799$, corresponding to fundamental frequencies of 55.125/60 Hz to 1.92/2.09 kHz for sampling rates of 44.1/48 kHz; maximum number of periodic components of $P_{\max} = 5$; number of gain correction quantization levels of $\mathcal{N}_G = 4$; number of survivors of $S = 64$; and $G_{(i,n)}$ quantization levels of $\{0.5, 0.75, 1, 1.25\}$. The low decoder complexity variant of CLTP coder uses number of gain magnitude quantization levels of $\mathcal{N}_r = 10$, gain angle quantization levels of $\mathcal{N}_\theta = 20$, and $\mathcal{N}_N = 10$ number of lag clusters. Note that the CLTP side information rate varies for every frame depending on the estimated parameters and this is included in the total rate.

1) 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. 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 1 dB was observed. These modest gains were due to the fact that these files had a dominant periodic component (e.g., in mfv) and

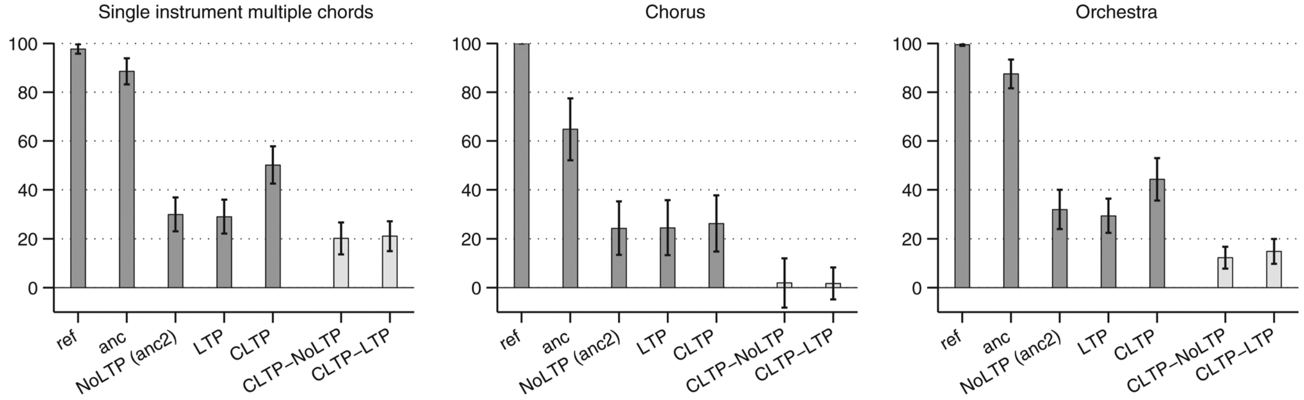


Fig. 7. MUSHRA listening test average scores with 95% confidence intervals, comparing MPEG AAC encoders with no LTP, standard LTP and proposed CLTP, for the three classes of dataset. The dark grey bars are absolute scores and the light grey bars are difference scores.

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 0.3 dB due to the fact that the pitch periods in this file vary rapidly over the duration of a relatively long prediction frame length (of $2K = 1024$ for AAC, versus $B(9+K) = 208$ for SBC), thus compromising the efficacy of CLTP, which depends on matching periodic components' waveforms. Also note that the additional performance gains of the low decoder complexity variant of CLTP, over standard LTP, though not as large as those of 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 VII-C).

2) Subjective Evaluation Results: The competing MPEG AAC coders, operating at around 24kbps, were evaluated for subjective quality via the MUSHRA listening tests [33]. 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. We operate at low rates to emphasize the improvements provided by our proposed scheme. 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 (which acts as another anchor (anc2), representing a well known quality level), standard LTP and the proposed full complexity CLTP. The MUSHRA test results with average and 95% confidence intervals for absolute and difference scores are shown in Fig. 7 for the three classes of the test dataset. The difference scores highlight the degree of preference of CLTP over no LTP, and over standard LTP. Note that the 95% confidence interval for the difference scores of the chorus file extends to negative territory, indicating an inconsistent performance improvement, due to the fact that the pitch periods in this file vary rapidly over the duration of a relatively long prediction frame length. For all other files, the subjective evaluation results 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.

C. Complexity

The proposed prediction technique is of higher complexity than LTP, mainly due to the elaborate parameter estimation performed for each P, recursively. The complexity information for crude implementations of the proposed coders for the evaluated dataset is listed in Table II. 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 increase in complexity due to LTP over non-predictive coder (specifically in Bluetooth SBC), can be reduced using well known techniques that trade estimation accuracy for complexity, e.g., using subsampled version of data while estimating lags, and reducing the number of elements in equations (22) and (24). 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 roughly same factor of reduction in CLTP complexity. We also observe from Table II that our proposed low decoder complexity variant for MPEG AAC is successful in its objective of incurring virtually no complexity penalty at the decoder. The development of low decoder complexity variant for Bluetooth is currently underway, as many Bluetooth applications are power constrained. However, the proposed backward adaptive systems are already useful for applications that are not power sensitive, such as multichannel wireless home theater systems.

TABLE II
COMPLEXITY OF THE PROPOSED CODERS

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

VIII. CONCLUSION

This work 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 a moderate complexity, 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.

APPENDIX A
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 representation 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 $\mathcal{N}_r$, $\mathcal{N}_\theta$ 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.

APPENDIX B
ENCODING CLTP SIDE INFORMATION

Based on our assumption that the audio signal is locally stationary, we exploit temporal dependencies of CLTP side information across 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 $m[i] = -1$. 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,

$$J = \sum_{i=0}^{P_n - 1} \left\{ \begin{array}{ll} |N_{(i,n)} - N_{(m[i],n-1)}|, & \text{if } m[i] \neq -1, \\ N_{(i,n)}, & \text{otherwise.} \end{array} \right\}. \quad (30)$$

Minimizing this cost function will 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 $m[i] \neq -1$, and $\tilde{N}_{(i,n)} = 0$, if $m[i] = -1$. We find the mapping using a low complexity technique summarized in Algorithm 3, which is a simplification and extension of the well known Hungarian algorithm for assignment [34] and is similar to the frame-to-frame peak matching proposed in [35].

Algorithm 3

```
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
    $$D_{(i,j)} = \begin{cases} |N_{(i,n)} - N_{(j,n-1)}|, & \text{if } j \neq P_{n-1}, \\ N_{(i,n)}, & \text{otherwise.} \end{cases} \quad (31)$$
Assign $C = 10^{20}$ (infinite for practical purposes relative to other quantities).
while $D_{(i,j)} \neq C$, for any $(i, j)$
    Identify $(i_{\min}, j_{\min}) = \arg\min_{\forall i,j} D_{(i,j)}$.
    Assign match index for this $i_{\min}$ as,
        $$m[i_{\min}] = \begin{cases} j_{\min}, & \text{if } j_{\min} \neq P_{n-1}, \\ -1, & \text{otherwise.} \end{cases} \quad (32)$$
    Set $D_{(i_{\min},j)} = C$, for $j = 0, \ldots, P_{n-1}$.
    if $j_{\min} \neq P_{n-1}$
        Set $D_{(i,j_{\min})} = C$, for $i = 0, \ldots, P_n - 1$.
    end if
end while
```

The lag bitstream sent to the decoder finally contains: 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 mass function required to calculate this table was estimated using parameters obtained from a wide range of audio signals. For encoding current lags, the minimum rate achievable requires 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 is considerable memory requirement, even for a modest $N_{\max} = 800, N_{\min} = 23$. On the other hand, a single table (of length $2N_{\max} + 1$) to encode the lag prediction residue $\bar{N}_{(i,n)} = N_{(i,n)} - \tilde{N}_{(i,n)}$, would require considerably smaller memory, but would result in rate increase for encoding current lags. As a tradeoff between the two extremes we classify previous lags into one of $\mathcal{N}_N$ groups, with each group having its own entropy coding table for the lag prediction residue $\bar{N}_{(i,n)}$. This approach requires only $\mathcal{N}_N$ 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 $\mathcal{N}_N$ 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 $I[p] \in \{1, \ldots, \mathcal{N}_N\}$, for $p = 0, N_{\min}, \ldots, N_{\max}$. We denote by, $U_N[q,p], q = 1, \ldots, \mathcal{N}_N, p = -N_{\max}, \ldots, 0, \ldots, N_{\max}$, the final $\mathcal{N}_N$ conditional entropy coding tables of the lag prediction residue. Note that all the probability mass functions required to calculate these tables were estimated using parameters obtained from a wide range of audio signals. 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:

$$B_j = \begin{cases} 0, & \text{if } m[i] \neq j \forall i, \\ 1, U_N[I[\tilde{N}_{(i,n)}], \bar{N}_{(i,n)}], & \text{if } m[i] = j. \end{cases} \quad (33)$$

We send the remaining unmatched lags of the current frame after all B_j. Thus the lag bitstream consists of $U_P[P_n], B_0, \ldots, B_{P_{n-1}-1}, U_N[I[0], \bar{N}_{(i,n)}], \forall\{i|m[i] = -1\}$. 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 $m[i] \neq -1$, and $\tilde{r}_{(i,n)} = 0, \tilde{\theta}_{(i,n)} = 0$, if $m[i] = -1$. 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 $\mathcal{N}_r + 1$ and $\mathcal{N}_\theta + 1$, and since the nominal $\mathcal{N}_r$ and $\mathcal{N}_\theta$ are small, e.g., $\mathcal{N}_r = 10$ and $\mathcal{N}_\theta = 20$, using a conditional entropy coding table for every possible predicted value, results in manageable size of tables, $(\mathcal{N}_r + 1)\mathcal{N}_r$ and $(\mathcal{N}_\theta + 1)\mathcal{N}_\theta$. We denote by, $U_r[q,p], q = 0, \ldots, \mathcal{N}_r, p = 1, \ldots, \mathcal{N}_r$ and $U_\theta[q,p], q = 0, \ldots, \mathcal{N}_\theta, p = 1, \ldots, \mathcal{N}_\theta$, 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 mass functions required to calculate these tables were estimated using parameters obtained from a wide range of audio signals.

Finally the per-SFB prediction activation flags, $f_n[b]$, 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 mass function for each flag, $\mathbf{P}_b[q,p], b \in \{0, \ldots, B-1\}$, where $q \in \{0,1\}$ indicates the state of the bth flag in previous frame, and $p \in \{0,1\}$ indicates the state of the bth flag in current frame. Note that these probability mass functions were estimated using parameters obtained from a wide range of audio signals. Also we assume these flags to be independent of each other as we observed while estimating the probability mass functions that the joint probability was very closely approximated by the product of the marginal probabilities. Independently encoding these flags would require B bits, and for the AAC-LD encoder with $B = 36$, this will be a significant increase in side information rate. Instead, to encode the flags at the rate dictated by the probability of occurrence of the sequence of flags, we employ arithmetic coding and require only $\lceil \prod_b \mathbf{P}_b[f_{n-1}[b], f_n[b]] \rceil$ bits to encode the flags. An arithmetic coder with fixed-point precision of 15 bits as described in [36] was used in simulations.

REFERENCES

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

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

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

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

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

[7] T. Nanjundaswamy and K. Rose, "Cascaded long term prediction for coding polyphonic audio signals," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, Oct. 2011, pp. 21–24.

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

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

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

[11] A. de Cheveigné, "A mixed speech F_0 estimation algorithm," in *Proc. 2nd Eur. Conf. Speech Commun. Technol. (Eurospeech '91)*, Sep. 1991.

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

[13] *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.

[14] 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.

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

[16] A. Aggarwal, S. L. Regunathan, and K. Rose, "A trellis-based optimal parameter value selection for audio coding," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 2, pp. 623–633, Mar. 2006.

[17] C. Bauer and M. Vinton, "Joint optimization of scale factors and huffman codebooks for MPEG-4 AAC," in *Proc. 6th IEEE Workshop. Multimedia Signal Process.*, Sep. 2004, pp. 111–114.

[18] 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, Apr. 1989.

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

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

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

[22] 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.

[23] J. Marques, I. Trancoso, J. Tribolet, and L. Almeida, "Improved pitch prediction with fractional delays in CELP coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1990, pp. 665–668.

[24] D. Veeneman and B. Mazor, *Efficient Multi-Tap Pitch Prediction for Stochastic Coding*, ser. Kluwer Int. Ser. in Engineering and Computer Science. New York, NY, USA: Springer, 1993, pp. 225–225.

[25] 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, Jun. 1992.

[26] J. Chen, "Toll-quality 16 kb/s CELP speech coding with very low complexity," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 9–12.

[27] W. Kleijn and K. Paliwal, *Speech coding and synthesis*. Norwell, MA, USA: Elsevier, 1995, pp. 95–102.

[28] A. Shah, S. Atungsiri, A. Kondoz, and B. Evans, "Lossy multiplexing of low bit rate speech in thin route telephony," *Electron. Lett.*, vol. 32, no. 2, pp. 95–97, 1996.

[29] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. ISMIR*, 2006, pp. 216–221.

[30] M. Christensen, L. Højvang, A. Jakobsson, and S. Jensen, "Joint fundamental frequency and order estimation using optimal filtering," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 1–18, 2011.

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

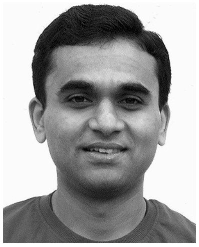
[32] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist. Soc.. Ser. B (Methodol.)*, pp. 267–288, 1996.

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

[34] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, no. 1-2, pp. 83–97, 1955.

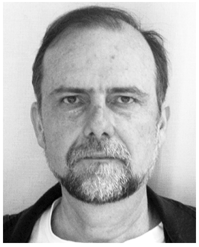
[35] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 4, pp. 744–754, Aug. 1986.

[36] A. Said, "Introduction to arithmetic coding-theory and practice," Hewlett Packard Laboratories Rep., 2004.



Tejaswi Nanjundaswamy (S'11–M'14) received the B.E degree in electronics and communications engineering from the National Institute of Technology Karnataka, India, in 2004 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara (UCSB), in 2009 and 2013, respectively. He is currently a post-doctoral researcher at Signal Compression Lab in UCSB, where he focuses on audio/video compression, processing and related technologies. He worked at Ittiam Systems, Bangalore, India from 2004 to 2008 as Senior Engineer on audio codecs and effects development. He also interned in the Multimedia Codecs division of Texas Instruments (TI), India in 2003.

Dr. Nanjundaswamy is an associate member of the Audio Engineering Society (AES). He won the Student Technical Paper Award at the AES 129th Convention.



Kenneth Rose (S'85–M'91–SM'01–F'03) received the Ph.D. degree from the California Institute of Technology, Pasadena, in 1991.

He then joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Professor. His main research activities are in the areas of information theory and signal processing, and include rate-distortion theory, source and source-channel coding, audio-video coding and networking, pattern recognition, and non-convex optimization. He is interested in the relations between information theory, estimation theory, and statistical physics, and their potential impact on fundamental and practical problems in diverse disciplines.

Prof. Rose was corecipient of the 1990 William R. Bennett Prize Paper Award of the IEEE Communications Society, as well as the 2004 and 2007 IEEE Signal Processing Society Best Paper Awards.