Audio Engineering Society
Convention Paper

Presented at the 116th Convention
2004 May 8–11 Berlin, Germany

Backward Linear Prediction for Lossless Coding of Stereo Audio

Jean-Luc Garcia[1], Philippe Gournay, and Roch Lefebvre

Department of Electrical and Computer Engineering
University of Sherbrooke
Sherbrooke (Québec) J1K 2R1 CANADA
Philippe.Gournay@Usherbrooke.ca

ABSTRACT

Lossless audio coding aims at achieving the lowest possible bitrate for transmission or storage of audio without any loss of information. This is usually done by first removing redundancy from the audio signal, and then applying entropy coding to the residual signal. Linear prediction (LP), when applied to monophonic signals, is a very effective way to remove redundancy. It produces minimum-phase predictors that are efficiently compressed by combining vector quantization with a meaningful representation of the LP coefficients (such as the LSFs). When applied to stereo signals however, joint channel prediction often produces non-minimum-phase predictors, whose quantization requires a high bit rate and poses stability problems. In this paper, we show that backward estimation of the LP coefficients (where those are estimated on the past decoded signal) solves most of the problems associated with the use of joint channel prediction in a lossless audio coder.

1. INTRODUCTION

The aim of audio coding in general is to reduce the amount of data necessary to transmit or store an audio signal. Lossy coding schemes are generally based on subband or transform coding and provide relatively high compression ratios. For example, MPEG audio coders, as exemplified by the MP3 format, can achieve compression ratios of up to 16 without significant loss in perceptual quality. However, there are also a number of applications that require perfect reconstruction of the original signal, which requires the use of lossless compression techniques.

Lossless audio coders usually work by first removing redundancy from the audio signal, then applying entropy coding to the residual signal. Linear prediction (LP), when applied to monophonic signals, is a very effective way to remove redundancy. It produces minimum-phase predictors that are efficiently compressed by combining vector quantization with a meaningful representation of the LP coefficients (such as the LSFs [8]). In the case of a stereo signal however, approaches such as the "mid/side" (where a standard linear predictor is applied to the middle channel, and entropy coding is applied both to the residual and side signals) or joint stereo coding [6] are generally preferred

[1] Jean-Luc Garcia is no longer with the University of Sherbrooke

to joint channel prediction. This is mainly because joint channel prediction often gives rise to non-minimum phase predictors, whose quantization requires a high bit rate and poses stability problems.

In this paper, we show that backward estimation of the LP coefficients solves most of the problems associated with the use of joint-channel linear prediction. In the backward approach, the LP coefficients are estimated on the past decoded signal, and no bit rate is dedicated to their transmission. Therefore, it is not necessary anymore to make a compromise between the bit rate, the order of the predictors (which conditions their gain), and the stability of the decoded synthesis filters. As a consequence, the compression ratios obtained by the lossless audio coder are improved.

2. LOSSLESS AUDIO CODING

The general framework for most lossless audio coders is shown in Figure 1. The input audio signal is first decomposed into consecutive blocks or frames. The frame length depends on the algorithm, but it is typically comprised between 10 and 80 ms [4]. Constant frame lengths are common, but some algorithms use variable frame lengths. In that case, the frame length has to be transmitted as side information.

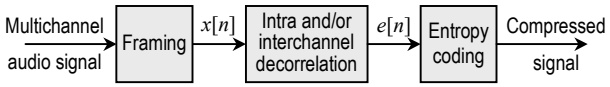

Figure 1: Basic operations of lossless encoding

The next step in the encoding process is to remove redundancy from the audio signal without removing information, so that a minimum amount of data will have to be encoded and transmitted or stored. This is generally done by decorrelating the audio samples $x[n]$ within a frame. Some algorithms use linear transforms for that purpose, others use linear prediction. Those two approaches are described in more details below. The result of the decorrelation is a residual or error signal $e[n]$, which forms the input of an entropy coder. The error signal $e[n]$ has an average amplitude that is much smaller than that of the input signal $x[n]$, and a relatively flat power spectral density. A lower amplitude means that fewer bits are needed to encode each sample $e[n]$. To be optimal, the choice of the entropy coder should depend on the distribution of $e[n]$.

2.1. Linear prediction

Linear prediction is one of the most common techniques used in lossless audio coders to decorrelate the audio samples [10-12]. Using linear prediction, an estimation $\hat{x}[n]$ of the present sample $x[n]$ is given by a linear combination of the K previous values $x[n-1]$, $x[n-2]$, …, $x[n-K]$ under the form:

$$\hat{x}[n] = \sum_{k=1}^{K} a_k x[n-k] \cdot \qquad (1)$$

The optimal predictor is the set of $\{a_k, 1 \le k \le K\}$ which minimizes the variance of the prediction error $e[n] = x[n] - \hat{x}[n]$. This problem leads to the well-known *Yule-Walker* equations, which involve the autocorrelations of $x[n]$ and can be resolved by the *Durbin-Levinson* iterative algorithm [7].

It can be shown that the optimal linear predictor for a given signal $x[n]$ also decorrelates the signal, i.e. that the prediction error $e[n] = x[n] - \hat{x}[n]$ has a relatively flat power spectral density provided the order K is sufficiently large. The maximum amplitude of the error signal $e[n]$ also tends to decrease as the prediction order K increases.

The prediction gain G_p is a measure that quantifies the efficiency of a linear predictor [1]. It is defined as:

$$G_p = \frac{\sigma_x^2}{\sigma_e^2} > 1, \qquad (2)$$

where σ_x^2 is the energy of the signal, and σ_e^2 is the energy of the prediction error.

2.2. Orthogonal transform

Another way to remove redundancy from the audio signal is to encode a first approximation $\hat{x}[n]$ of the input signal $x[n]$, and then to encode the difference between $x[n]$ and $\hat{x}[n]$. The first approximation $\hat{x}[n]$ can be obtained by transform coding of $x[n]$ [4]. Transform coders generally use an orthogonal transform - such as the MDCT (modified discrete cosine transform) - to obtain a spectral representation $X[k]$. The coefficients $X[k]$ are quantized efficiently using a perceptually relevant bit allocation procedure. The temporal signal $\hat{x}[n]$ is recovered by computing the inverse transform of a quantized version $\hat{X}[k]$ of $X[k]$.

The quantization process has introduced some noise. However, if the difference $e[n]$ between $x[n]$ and $\hat{x}[n]$ is also transmitted, and provided that this error is encoded without any loss (as in Figure 1), then the original signal can also be recovered exactly without any loss of information. This approach can be seen as an embedded coder, with two levels of transmitted information: 1) the approximation from the transform coder, and 2) the error signal. The main advantage of this approach is that $\hat{x}[n]$ is a low bit rate representation of $x[n]$ with a certain level of preserved subjective quality. This feature may be useful in some applications. However, this approach is generally computationally more expensive than other approaches based merely on linear prediction plus entropy coding. Furthermore, embedded solutions are generally less efficient than non-embedded ones in terms of bit rate and compression ratio.

Since our goal here is to achieve the highest possible compression ratios, we do not consider embedded solutions. In what follows, we assume that linear prediction is used for decorrelation in the lossless coding algorithm.

2.3. Entropy coding

The linear prediction error is known to approximately follow a *Laplace* distribution [10]. Golomb-Rice codes [3,9] are variable-length codes that are very well suited for that kind of distribution.

Golomb-Rice codes make use of the fact that low-level samples need fewer bits to be represented. Since there are much more low-level samples than high-level samples in the linear prediction error, this results in a reduction in bit rate. For example, let us consider the sequence of values 9, -12, -15 and 56. When the signal is encoded with 16 bits per sample, which is standard for most audio signals, 4*16=64 bits in total are required to represent the 4 samples. It would be more effective to encode each of the samples separately using the minimum necessary number of bits. In that case, only 5 bits (including the sign) for the first three values and 7 for the last one would be necessary. This makes a total of 22 bits which is much lower that 64. However, without any side information, the decoder would be unable to separate the 4 codes and to interpret the sequence of bits correctly. Golomb codes make it possible to use different code lengths for each sample, while allowing the decoder to identify the beginning, the end and the structure of each code. Specifically, an integer n is represented by a three-field code that consists of:

1. a sign bit;
2. a prefix, which is the unary code for the quotient of the integer division "n\b" of n by b, where b is the Golomb parameter (i.e., a sequence of n\b 1's followed by a 0);
3. a suffix, composed of the $\log_2 b$ bits required to represent the remainder of n\b.

Rice codes are a special case of Golomb codes where b is a power of 2. This property allows for many simplifications in the encoding and decoding processes. The value of b that provides maximum coding efficiency depends on the distribution of the input. Since speech and audio signals are non-stationary by nature, this parameter should be updated regularly.

3. JOINT CHANNEL PREDICTION

Stereo audio signals generally exhibit correlation both within and between channels. Some lossless audio codecs work separately on the left and the right channels, they therefore do not take advantage of the correlation between channels. Other codecs encode separately the sum and the difference between the channels. When the two channels are highly correlated, the difference is small and can be encoded at a lower bit rate. Yet other coders encode only two of the following signals: left channel, right channel, and difference between channels. The two signals are chosen such as to minimize the bit rate.

In [1] and [5], joint-channel stereo linear prediction was proposed in order to perform simultaneously both intra and interchannel decorrelation. We follow this approach here. We call $x_1[n]$ the samples from the left channel, and $x_2[n]$ the samples from the right channel. Each sample from the left channel $x_1[n]$ is estimated by a linear combination of the K_a past values $x_1[n\text{-}k]$ and the K_b past values $x_2[n\text{-}k]$, as given by

$$\hat{x}_1[n] = \sum_{k=1}^{K_a} a_k x_1[n-k] + \sum_{k=1}^{K_b} b_k x_2[n-k], \qquad (3)$$

where the set of $\{a_k\}$ is called the autopredictor and the set of $\{b_k\}$ is called the crosspredictor. As in classical

monophonic linear prediction, we want to minimize the variance of the prediction error $e_1[n] = x_1[n] - \hat{x}_1[n]$. Minimizing this error with respect to a_k (and b_k) leads to (4) (and (5)), respectively:

$$r_{11}[k] = \sum_{l=1}^{K_a} a_l r_{11}[k-l] + \sum_{l=1}^{K_b} b_l r_{12}[l-k] \quad \forall k \in \{1,\ldots,K_a\} \tag{4}$$

$$r_{12}[k] = \sum_{l=1}^{K_a} a_l r_{12}[k-l] + \sum_{l=1}^{K_b} b_l r_{22}[k-l] \quad \forall k \in \{1,\ldots,K_b\} \tag{5}$$

where $r_{pq}[k]=\mathrm{E}\{x_p[n]x_q[n-k]\}$ is the correlation function between signals $x_p[n]$ and $x_q[n]$. Equations (4) and (5) can be rearranged in a matrix form:

$$\begin{bmatrix} \mathbf{r_{11}} \\ \mathbf{r_{12}} \end{bmatrix} = \begin{bmatrix} \mathbf{R_{11}} & \mathbf{R_{12}} \\ \mathbf{R_{21}} & \mathbf{R_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{R_1} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \tag{6}$$

with the correlations and coefficients vectors:

$$\begin{aligned} \mathbf{r_{11}} &= [r_{11}[1], r_{11}[2], \ldots, r_{11}[K_a]]^T \\ \mathbf{r_{12}} &= [r_{12}[1], r_{12}[2], \ldots, r_{12}[K_b]]^T \\ \mathbf{a} &= [a_1, a_2, \ldots, a_{K_a}]^T \\ \mathbf{b} &= [b_1, b_2, \ldots, b_{K_b}]^T \end{aligned} \tag{7}$$

The autocorrelation matrices $\mathbf{R_{11}}$ and $\mathbf{R_{22}}$ and the crosscorrelation matrices $\mathbf{R_{12}}$ and $\mathbf{R_{21}}$ are *Toeplitz* and symmetric matrices. The optimal autopredictor and crosspredictor are given by inverting the $(K_a+K_b)\times(K_a+K_b)$ matrix $\mathbf{R_1}$ in the linear system given in Equation (6). In the special case where $K_a=K_b$, it is possible to use the multichannel *Durbin-Levinson* algorithm [7]. When $K_a \neq K_b$, the symmetry of the matrix allows the use of the *Cholesky* decomposition [5].

On the right channel, as both channels are interleaved, the new sample on the left channel $x_1[n]$ can be used to predict the value of $x_2[n]$ [1,5]. The crosspredictor $\{c_k\}$ consequently starts at index 0 rather than at index 1, as shown in the following equation expressing the intra and inter prediction for the right channel:

$$\hat{x}_2[n] = \sum_{k=0}^{K_c} c_k x_1[n-k] + \sum_{k=1}^{K_d} d_k x_2[n-k]\cdot \tag{8}$$

Minimization of the variance of the prediction error with respect to c_k and d_k leads to a set of equations similar to (6). These equations can be expressed in a matrix form involving a matrix $\mathbf{R_2}$. As for the left channel, $\mathbf{R_2}$ is symmetric and can be inverted using the *Cholesky* decomposition.

In [5], some compression ratios were given for a lossless audio coder using such forward-estimated joint-channel stereo linear predictors. The prediction coefficients were encoded by scalar quantization with 12 bits per coefficient. This is much more than what is needed to quantize a monophonic linear predictor, using line spectral pairs (LSPs) for example [8]. In fact, simulations showed that the resolution of (6) does not always lead to minimum-phase synthesis filters. Therefore, efficient quantization techniques such as those combining the LSP representation with vector quantization cannot be used. Furthermore, the quantization noise should stay low enough to guarantee the stability of the decoded synthesis filter. Therefore, the bit rate required to transmit forward-estimated joint-channel stereo linear predictors is necessarily high. This puts this technique at a serious disadvantage.

4. BACKWARD ANALYSIS

We saw in the previous section that the transmission of joint-channel predictors estimated using the forward approach poses a number of problems related to efficiency and stability. To get around these problems, we propose to use the backward approach that was first introduced in order to limit the algorithmic delay of a lossy speech coder [2]. In the backward approach, the prediction coefficients are estimated from the past decoded signal, which is available both at the encoder and the decoder. More specifically, the prediction coefficients are estimated on the past frame of audio signal instead of current frame, but the resulting predictor is applied to the current frame. Since the encoder and the decoder can perform exactly the same operation on the past audio signal, there is no need to quantize and transmit the prediction coefficients as in the forward approach. Consequently, there is also no need to limit the prediction orders K_a, K_b, K_c, and K_d as in the forward approach (except of course for the sake of computational complexity).

4.1. Correlations estimates

As mentioned in section 3, two matrices of autocorrelation and crosscorrelation $\mathbf{R_1}$ and $\mathbf{R_2}$ are inverted in order to obtain the stereo linear prediction coefficients. In the forward approach, the correlation matrices are typically estimated on a segment of audio samples weighted by a data window, this window being centered on the analysis frame. In the backward approach, such a weighting puts too much emphasis on the value of older samples [2], and an asymmetric window that favors more recent samples is preferable. In the present work, the hybrid data window represented on Figure 2 is used to compute the correlations. The left part of the window is the first half of a Hamming window. The right part of the window is a quarter period of a cosine function, characterized by a shorter and faster decrease. In our implementation, the data window is 25 ms long and the cosine section takes only 5% of this length. Note that this window is highly asymmetrical, its maximum value being much closer to the end of the window.

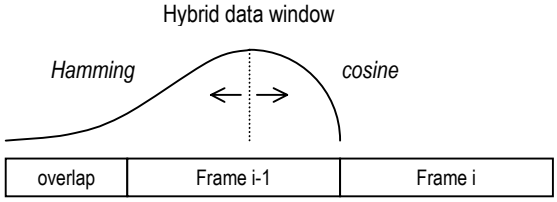

Figure 2: The hybrid data window used for backward LP analysis

As in conventional LP analysis, a lag window is applied to the autocorrelation sequences, and a noise floor at -40 dB is added on both channels in order to make sure that the correlation matrices are invertible.

4.2. Subframes analysis

Since the prediction coefficients are not transmitted anymore in the backward approach, the LP analysis can be performed more frequently than in the forward approach. Frequent adaptation has a positive impact on the compression ratio because the LP model is more closely adapted to the audio signal. The only negative impact is on complexity. In our implementation, the frames are 20 ms long, and the LP adaptation rate is once every 5 ms. The linear predictors are kept constant over a 5 ms subframe.

The encoder and decoder use exactly the same scheme to compute the predictors on each channel.

5. RESULTS

In this section, the impact of backward analysis on the prediction gain and on the compression ratio of a lossless stereo encoder is quantified.

Table 1 gives the prediction gains obtained by the forward and backward approaches on various audio signals. The autoprediction and interprediction orders are K_a=K_d=20 and K_b=K_c=10 for both approaches. The forward prediction coefficients are computed using a symmetric Hamming window centered on the frame, and they are updated once every 20 ms. Note that they are *not* quantized in that experiment. This means that the prediction gains reported for the forward approach are the best achievable gains for the orders under consideration. It is clear that quantizing the predictors would result in a reduction of the prediction gain, the magnitude of which depends on the number of bits used.

J.S. Bach [14]	**Forward**	**Backward**
Left channel	30.68 dB	30.54 dB
Right channel	31.83 dB	31.69 dB
S. Vega [20]		
Left channel	22.44 dB	22.12 dB
Right channel	30.08 dB	29.93 dB
E. Clapton [16]		
Left channel	25.38 dB	24.98 dB
Right channel	27.71 dB	27.16 dB

Table 1: Prediction gains for the forward and backward approaches

Table 2 gives the compression ratios obtained by the forward and backward approaches on the same signals. In this experiment, the forward prediction coefficients are scalarly and uniformly quantized using 11 bits per coefficient. Since there are 60 prediction coefficients, 11*60 = 660 bits must be sent for each 20 ms frame. This represents a significant overhead for the forward approach, but as it was said above this overhead is quite unavoidable.

	Forward	Backward
J.S. Bach	2.00	2.10
S. Vega	2.38	2.50
E. Clapton	1.52	1.57

Table 2: Compression ratios obtained by the forward and backward approaches

Table 1 shows that the prediction gain is slightly lower with the backward approach than it is with the forward approach. However, Table 2 also shows that the bit rate reduction due to not transmitting the coefficients in the backward case counterbalanced the decrease in predictor performance involved by the backward approach. Hence, the compression ratios are in the end improved by backward estimation.

A clear advantage of the backward approach is that, since it is not necessary anymore to transmit the prediction coefficients, one may set higher prediction orders without increasing the bit rate. Simulations were performed with different orders on a track extracted from [20]. The results presented in Table 3 show that:

- Increasing the prediction orders improves the compression ratio;
- Above a certain order, this improvement is not significant anymore and is obtained at the cost of a higher computational complexity (the algorithm must invert a larger matrix).

This validates our choice to use 20 for the autoprediction orders and 10 for the interprediction orders.

Prediction order $(K_a=K_d)$ / $(K_b=K_c)$	**Compression ratio**
20/10	1.4511
30/10	1.4525
30/30	1.4534
40/40	1.4544

Table 3: Influence of the prediction order on the compression ratio

Finally, let us compare the performances of our lossless audio coder with the performances of some state of the art algorithms on various types of audio signals. We choose to use LPAC [12] and Monkeys audio [13] as reference codecs, because both of them achieve the highest compression ratios on most audio signals. Table 4 shows again that the backward approach outperforms the forward approach in terms of compression ratio. Table 4 also shows that the results obtained by our coder using backward estimation of the joint-channel stereo linear predictors are comparable to the results obtained by the reference codecs.

	LPAC	Monkeys	Forward	Backward
Luka [20]	1.422	1.490	1.422	1.452
Born in the USA [19]	1.424	1.461	1.390	1.418
Training [18]	1.582	1.621	1.532	1.568
Cosmic girl [17]	1.521	1.578	1.510	1.540
Polonaise [15]	2.610	2.700	2.200	2.284

Table 4: Comparison with some state-of-the-art lossless audio coders

Note that our lossless audio coder is composed only of a joint-channel stereo linear predictor followed by a Rice code. The frame length is fixed, and the Golomb parameter is updated once per subframe. Furthermore, our coder uses a very straightforward implementation of the Rice code. The two only special features that we implemented are:

- Coding of zero subframes (an all-zero subframe is indicated by a zero Golomb parameter);
- Coding of prediction residuals exceeding the 16 bits limits (when the absolute value A of a residual sample is above 2^{16}, it is transmitted as the code for 2^{16} followed by the code for A-2^{16}.

Obviously, there is still plenty of room for improvement in our coder.

6. CONCLUSION

In this paper, we have shown that backward estimation of the LP coefficients solves most of the problems associated with the use of joint-channel stereo linear prediction in a lossless audio coder. In the backward approach, the prediction coefficients are estimated on the past decoded signal. Since this operation is possible both at the encoder and the decoder, no bit rate is dedicated to their transmission. Therefore, there is no need anymore to make a compromise between the bit rate, the order of the predictors (which conditions their gain), and the stability of the decoded synthesis filters. As a consequence, the compression ratios obtained by the lossless audio coder are improved.

7. REFERENCES

[1] P. Cambridge and M. Todd, "Audio data compression techniques", presented at the 94th AES Convention, Berlin, Germany, 1993 March 16-19.

[2] J.-H. Chen et al., "A low-delay CELP coder for the CCITT 16 kb/s speech coding standard", IEEE J. Select. Areas Commun., vol. 10, no. 5, pp. 830-849 (1992 June).

[3] S.W. Golomb, "Run-Length Encodings", IEEE Trans Info. Theory, vol. 12, n. 3, pp. 399-401 (1966 July).

[4] M. Hans and R.W. Schafer, "Lossless compression of digital audio", IEEE Signal Processing magazine, vol. 18, no. 4, pp. 21-32 (2001 July).

[5] T. Liebchen, "Lossless audio coding using adaptative multichannel prediction", presented at the AES 113th Convention, Los Angeles, CA, USA, 2002 October 5-8.

[6] T. Liebchen, "MPEG-4 Lossless Coding for High-Definition Audio", presented at the AES 115th Convention, New York, NY, USA, 2003 October 10-13.

[7] S. L. Marple Jr., Digital Spectral Analysis with Applications, Prentice-Hall, Englewood Cliffs, N.J., 1987.

[8] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", IEEE Trans. on Speech and Audio Processing, vol. 1, no. 1, pp. 3-14 (1993 January).

[9] R.F. Rice, Some practical universal noiseless coding techniques, Tech. Rep. JPL-79-22, Jet Propulsion Laboratory, Pasadena, CA, March 1979.

[10] T. Robinson, Simple lossless and near lossless waveform compression, Tech. Rep. CUED/F-INFENG/TR.156, Cambridge University Engineering Department, Cambridge, UK, December 1994

[11] A. Wegener, "MUSICompress: Lossless, low-MIPS audio compression in software and hardware", in Proc. Int. Conf. Signal Processing Applications and Technology, San Diego, CA, USA, 1997 September 14-17.

[12] T. Liebchen, The LPAC Homepage. Software, http://www.nue.tu-berlin.de/wer/liebchen/lpac.html

[13] Monkeys Audio Homepage. Software, http://www.monkeysaudio.com/

[14] Johann Sebastian Bach, Violin Concertos No. 1&2 - Concerto for 2 Violins (Christopher Hogwood), 1981.

[15] Frédéric Chopin, Polonaises (Maurizio Pollini), Label: Deutsche Grammophon, 1976.

[16] Eric Clapton, Unplugged, Label: Reprise, 1992.

[17] Jamiroquai, Travelling Without Moving, Label: Sony Music, 1997.

[18] Michel Petrucciani, Trio In Tokyo, Label: Dreyfus Jazz, 1999.

[19] Bruce Springsteen, Greatest Hits, Label: Columbia, 1995.

[20] Suzanne Vega, Solitude Standing, Label: A&M, 1987.