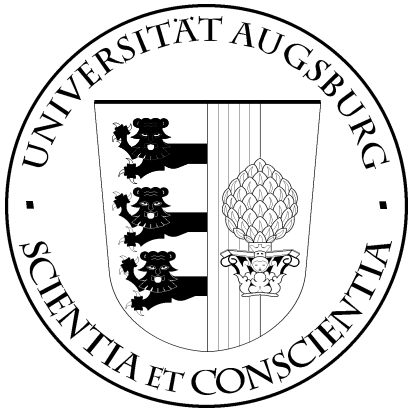
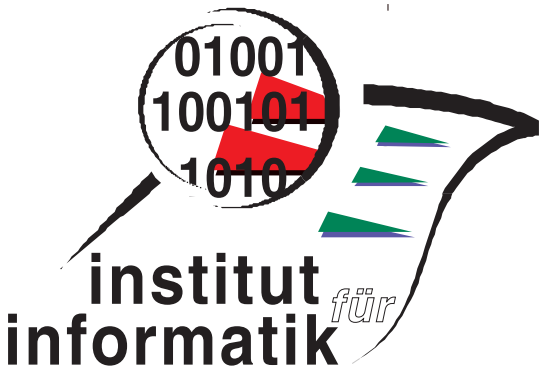
UNIVERSITÄT AUGSBURG



Predictive Modeling for Lossless Audio Compression

Jong-Hwa Kim

Report 2004-5 April 2004



INSTITUT FÜR INFORMATIK
D-86135 AUGSBURG

Predictive Modeling for Lossless Audio Compression

Jong-Hwa Kim

Institute of Computer Science, University of Augsburg
Eichleitnerstr. 30, 86159 Augsburg, Germany
kim@ieee.org
http://mm-werkstatt.informatik.uni-augsburg.de

Abstract. Autoregressive (AR) modeling by linear prediction (LP) provides the basis of a wide variety of signal processing and communication systems including parametric spectral estimation and system identification. Perhaps the greatest success of linear prediction techniques is to be found in speech analysis and audio coding. In this paper, we first reviewed the general frameworks of predictive signal modeling and investigated various prediction filter structures including the modified linear predictor. We then empirically compared the compression performamce of these prediction filters by applying to the lossless audio compression system. We also applied different filter orders and block lengths for each filter to explore their influence on the compression ratio. ...

1 Introduction

Recently a number of new technologies related to the storage capacity and the transmission bandwidth are emerging. From transmission part, for example, ADSL (Asymmetric Digital Subscriber Line) provides several Mbps transmission bandwidth for normal telephone line on down stream side. On storage part, hard disk capacity has being increased dramatically. A new high-density disc such as DVD (Digital Versatile Disc) also provides huge storage capability for audio and video sources. Despite such tremendous growth of the storage capacity and the transmission bandwidth, the demand for higher quality of multimedia associated with audio, image, and video continues to outpace it. For instance, the required data rate satisfying the high-quality audio (more word size, more sample rate, and more channel) will be continuously increased, unless we give up to enjoy the digital audio world. Hence the importance of data compression is not likely to diminish, as a key technology to allow efficient storage and transmission.

The general idea behind data compression is to remove the redundancy present in the data to find more compact representations. Two families of algorithms exist in compression. When the information can be exactly recovered from the bits, the source coding or compression is called *lossless*; otherwise, it is called *lossy*. To achieve higher compression ratios, lossy algorithms remove information from the original in a way that comes *close* to the original or that is not perceptible. In this case, therefore, we allow approximate representations of the original, instead of trying to represent the original exactly, and have only a modified version of the original after transmission. In contrast, lossless algorithms respect the integrity of the original signal. After transmission and reconstruction an exact copy of the original signal is available.

Two main drawbacks are related with lossless audio compression. The first one is the time varying compression ratio, which makes difficult the allocation of a fixed bandwidth to the transmitted audio data and also complicates the editing of compressed material. The second one is the poor compression rate. The lossless audio compression usually achieves compression ratio 2 to 3 without loss any quality, whereas lossy compression can achieve compression ratio 8 to 40 or higher. While achieving higher compression ratios, the lossy audio compression is highly objectionable in high fidelity audio compression applications, due to unexpected artifacts introduced even by the most heavily engineered schemes which use perceptual auditory models. It is more problematical whenever the audio signal undergoes multiple encoding-decoding operations.

The goal of the predictive modeling in lossless audio compression is to reduce the sample magnitudes by making a prediction of the current sample on the basis of previous sample values and by transmitting just the difference between the current sample and the prediction. Several predictive methods exist for exploiting correlation between neighboring samples in a given data stream. The same decorrelation function is used in compression and reconstruction, and this function must take as input a delayed version of the

input sequence. In this paper, we first review the general frameforks of predictive signal modeling and investigate the related works on lossless audio compression using prediction filters. For comparison of the compression performance with various types of prediction filters we developed a prototypical lossless audio compression system in which we apply each prediction filter to the signal decorrelation stage. We will also use the different prediction orders and block lengths for each filter to explore their effect on compression ratio.

2 Linear Prediction Filters

2.1 General expressions

The basic idea behind linear prediction is that a sample of signal can be approximated as a linear combination of previous samples. By minimizing the sum of the squared differences between input samples and linearly predicted ones, a unique set of predictor coefficients can be determined.

From the discrete signal processing viewpoint, a digital filter $H(z)$ is assumed to have p poles and q zeros in the general *pole-zero* case, which means that given the input sequence $x[n]$, its approximation $\hat{x}[n]$ can be modeled by a combination of the q previous output samples and $p+1$ previous input samples in a discrete filter system

$$\hat{x}[n] = \sum_{k=0}^{p} b_k \, x[n-k] + \sum_{k=1}^{q} a_k \, \hat{x}[n-k] \,, \tag{2.1}$$

which is equivalent to

$$H(z) = \frac{\hat{X}(z)}{X(z)} = \frac{\sum_{k=0}^{p} b_k \, z^{-k}}{1 - \sum_{k=1}^{q} a_k \, z^{-k}} \,. \tag{2.2}$$

This is the general description of pth-order *recursive* or *infinite impulse response* (IIR) system. Such model with both poles and zeros is also called an *autoregressive moving average* (ARMA) model. If we set the coefficients $a_1, a_2, \cdots, a_q$ equal to zero, (2.1) becomes a *finite impulse response* (FIR) system, i.e.,

$$\hat{x}[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_p x[n-p] = \sum_{k=0}^{p} b_k \, x[n-k] \,. \tag{2.3}$$

This is an allpole model, also known as an *autoregressive* (AR) model, while the allzero model ($p = 0$) is called a *moving average* (MA) model since the output is a weighted average of the q prior inputs. Figure 2.1 shows the general representation of these filter systems.

As showed in Figure 2.1, the prediction of present value in an FIR system is performed by combination of past values of the input signal, while for an IIR system the estimation of present value depends on the immediate past values of the output and the present and past values of the input. That is, an IIR system involves a recursive process in order to update the output by the present and past values. As the result, the terms of FIR and IIR describe digital filters relative to the length of their sampled response sequences since it is possible to implement an FIR filter in a recursive fashion and an IIR filter in a nonrecursive manner.

Most of linear prediction system in literature assumes to use an allpole FIR filter because of its ability to provide extremely accurate estimate and its relative speed of computation. From (2.3) the system function of a pth-order FIR LP filter is the polynomial

$$P(z) = \sum_{k=1}^{p} b_k \, z^{-k} \,, \tag{2.4}$$

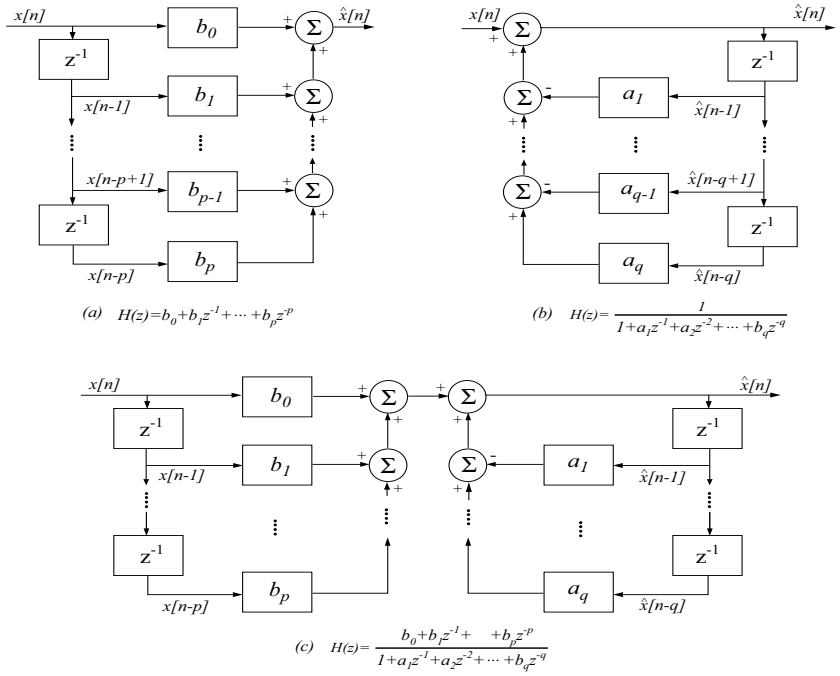


Fig. 2.1. Block diagram representation for three types of filters

where any zeros at $z = 0$ are ignored because such zeros contribute nothing to the spectral magnitude and add only linear phase. Output of this predictive filtering is the difference between the original input signal $x[n]$ and its prediction $\hat{x}[n]$,

$$e[n] = x[n] - \hat{x}[n] = x[n] - \sum_{k=1}^{p} b_k \, x[n-k] \,, \tag{2.5}$$

where the difference signal $e[n]$ is called *residual* or *prediction error*[1]. From (2.5) it can be seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^{p} b_k \, z^{-k} \,, \tag{2.6}$$

which is an inverse filter for the system $H(z)$, i.e., $H(z) = A(z)^{-1}$. The basic problem of LP is to determine a set of predictor coefficients $\{b_k\}$ directly from the signal in such a manner as to ontain a good estimate of the spectral properties of the signal through the use of (2.6).

The optimal predictor coefficients b_k are chosen to minimize the energy (average) in residual signal which is defined as

$$E = \sum_{n=-\infty}^{\infty} e^2[n] \tag{2.7a}$$

$$= \sum_{n=-\infty}^{\infty} (x[n] - \hat{x}[n])^2 \tag{2.7b}$$

$$= \sum_{n=-\infty}^{\infty} \left[x[n] - \sum_{k=1}^{p} b_k x[n-k] \right]^2 . \tag{2.7c}$$

[1] In the information theory $e(n)$ is often called the innovation sequence.

By setting $\partial E/\partial b_k = 0$, $k = 1, 2, 3, \ldots, p$, as a necessary condition for minimum energy, we obtain p linear equations

$$\sum_{n=-\infty}^{\infty} x[n-i]x[n] = \sum_{k=1}^{p} b_k \sum_{n=-\infty}^{\infty} x[n-i]x[n-k], \quad i = 1, 2, 3, \ldots, p\,. \tag{2.8}$$

If we define

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} x[n-i]x[n-k]\,, \tag{2.9}$$

then the linear equations (2.8) can be written more compactly as

$$\sum_{k=1}^{p} b_k\, \phi(i,k) = \phi(i,0), \quad i = 1, 2, 3, \ldots, p\,. \tag{2.10}$$

From (2.7), (2.8) and (2.10) the minimum mean-squared prediction error can be shown to be

$$E = \sum_{-\infty}^{\infty} x^2[n] - \sum_{k=1}^{p} b_k \sum_{-\infty}^{\infty} x[n]x[n-k] \tag{2.11a}$$

$$= \phi(0,0) - \sum_{k=1}^{p} b_k \phi(0,k) \tag{2.11b}$$

There are various algorithms to solve the linear equations (2.10), involving a special recursive algorithm for symmetric Toeplitz matrices. We review these algorithms in next section.

2.2 Estimation of the predictor coefficients

Autocorrelation method: To solve (2.10), the limits of summation in (2.7) and (2.8) must be over a finite interval. One of the basic approaches to determination of the limits is the well-known autocorrelation method. This approach assumes that the signal $x[n]$ has finite duration with length of N samples [2], i.e., $x[n] = 0$ outside the range $0 \le n \le N-1$. In this case, the residual signal $e[n]$ obtained through pth-order LP filter will be nonzero over the interval $0 \le n \le N-1+p$, and therefore the residual energy and (2.9) can be written as

$$E = \sum_{n=0}^{N+p-1} e^2[n]\,, \tag{2.12a}$$

$$\phi(i,k) = \sum_{n=0}^{N+p-1} x[n-i]x[n-k]\,, \tag{2.12b}$$

where $i = 1, 2, \ldots, p$ and $k = 0, 1, \ldots, p$. Since $\phi(i,k)$ of (2.12b) is identical to the autocorrelation function evaluated for $(i-k)$, the linear equations (2.10) can be expressed as a normal equations, called Wiener-Hopf equations[3] [1],

$$\sum_{k=1}^{p} a_k\, R(|i-k|) = R(i), \quad i = 1, 2, \ldots, p\,, \tag{2.13}$$

[2] Normally, a Hamming or similar time window is used to segment the signal

[3] Note that (2.13) is identical to the Yule-Walker equations obtained for an AR(p) model. Thus the LP filter can be considered as a whitening filter. The residual sequence $e[n]$ is white and efficient codeable with lower entropy than the entropy of $x[n]$, only if the input sequence $x[n]$ is an AR(p) process and pth-order LP filter is optimal.

which is in matrix form,

$$\begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ R(2) & R(1) & \dots & R(p-3) \\ \vdots & \vdots & \vdots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} \tag{2.14}$$

Similarly to (2.11), the minimum mean squared prediction error becomes

$$E = R(0) - \sum_{k=1}^{p} a_k R(k)\,, \tag{2.15}$$

where $R(k)$ is the kth correlation of the source and $R(0)$ is equal to the source variance σ_x^2. In (2.14), the $p \times p$ matrix of autocorrelation values is a Toeplitz matrix; i.e., it is symmetric and all the elements along a given diagonal are equal. This special property will be exploited to obtain an efficient algorithm for the solution of (2.13).

The Wiener-Hopf equations can be efficiently solved by the well-known Levinson-Durbin recursive procedure, in which the following operations are performed recursively for $m = 1, 2, \dots, p$:

$$k_m = \frac{\left[R(m) - \sum_{k=1}^{m-1} a_{m-1}[k]\, R(m-k) \right]}{E_{m-1}}\,, \tag{2.16a}$$

$$a_m[m] = k_m\,, \tag{2.16b}$$

$$a_m[k] = a_{m-1}[k] - k_m a_{m-1}[m-k], \qquad 1 \le k \le m-1\,, \tag{2.16c}$$

$$E_m = (1 - k_m^2) E_{m-1}\,, \tag{2.16d}$$

where initially $E_0 = R(0)$ and $a_0 = 0$. The *reflection coefficients* k_m guarantees a stable LP filter $H(z)$ so that at each cycle m the coefficients $a_m[k]$ describe the optimal mth-order linear predictor and the minimum error E_m is reduced by the factor $(1 - k_m^2)$. Therefore, the necessary condition for stable system is given by $|k_m| \le 1$. Particularly, the negatives of the k_m are called *partial correlation* (PARCOR) coefficients.

Covariance method: Autocorrelation method is usually preceded by a windowing of input signal, so that $x[n]$ values at the beginning and end of a block are tapered to zero. Thus, the autocorrelation solution introduces distortion into the spectral estimation procedure. The covariance method avoids this distortion since it does not need to window the input signal, but residual signal is weighted uniformly in time by a simple rectangular window of length N;

$$E = \sum_{n=0}^{N-1} e^2[n]\,. \tag{2.17}$$

Then the covariance function for $x[n]$ becomes

$$\phi(i,k) = \sum_{m=0}^{N-1} x[n-i]x[n-k] \tag{2.18a}$$

$$= \sum_{n=-k}^{N-k-1} x[n]x[n+k-i], \quad 1 \le i \le p,\ \ 0 \le k \le p\,. \tag{2.18b}$$

Setting $\partial E / \partial a_k = 0$ again to zero leads to p linear equations

$$\sum_{k=1}^{p} a_k\, \phi(i,k) = \phi(i,0), \quad i = 1, 2, \dots, p\,. \tag{2.19}$$

Since the evaluating $\phi(i,k)$ requires the input values of $x[n]$ in the interval $-p \le n \le N-1$, it does not need to taper the segment of input signal to zero at the ends as in the autocorrelation method. In matrix form, (2.19) becomes

$$\begin{bmatrix} \phi(1,0) \\ \phi(2,0) \\ \phi(3,0) \\ \vdots \\ \phi(p,0) \end{bmatrix} = \begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,p) \\ \phi(3,1) & \phi(3,2) & \dots & \phi(3,p) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(p,1) & \phi(p,2) & \dots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} \tag{2.20}$$

This covariance matrix is symmetric but not Toeplitz. As a result, the covariance method leads to a function that is not a true autocorrelation function, but rather, the cross-correlation between two very similar finite length segments of the input signal. From the difference of limitations in (2.12b) and (2.18a), the linear equations (2.19) have significantly different properties that strongly affect the method of solution and the properties of the resulting optimum predictor.

One of the well-known methods to solve the set of equations (2.19) is called the Cholesky decomposition (also referred as the square root method). In matrix notation, (2.19) can be written by

$$\mathbf{\Phi a} = \psi \,, \tag{2.21}$$

where $\mathbf{\Phi}$ is a positive definite symmetric matrix with (i,j)th element $\phi(i,j)$, and $\mathbf{a}$ and ψ are column vectors with elements a_j, and $\phi(i,0)$ respectively. For the Cholesky decomposition the matrix $\mathbf{\Phi}$ is given by

$$\mathbf{\Phi = VDV^T} \,, \tag{2.22}$$

where $\mathbf{\Phi}$ is a lower triangular matrix whose main diagonal elements are all 1's, and $\mathbf{D}$ is a diagonal matrix. The elements of the matrices $\mathbf{V}$ and $\mathbf{d}$ are readily determined by

$$V_{ij} d_j = \phi(i,j) - \sum_{k=1}^{j-1} V_{ik} d_k V_j k, \quad 1 \le j \le i-1 \,, \tag{2.23a}$$

$$d_i = \phi(i,i) - \sum_{k=1}^{i-1} V_{ik}^2 d_k, \quad d_1 = \phi(1,1), \quad i \ge 2 \,. \tag{2.23b}$$

The column vector $\mathbf{a}$ can be solved by using a simple recursion of the forms

$$Y_i = \psi_i - \sum_{j=1}^{i-1} V_{ij} Y_j, \quad p \ge i \ge 2 \,, \tag{2.24}$$

where

$$\mathbf{Y = DV^T a}, \quad Y_1 = \psi_1 \,, \tag{2.25}$$

and then

$$a_i = Y_i/d_i - \sum_{j=i+1}^{p} V_{ji} a_j, \quad 1 \le i \le p-1 \,, \tag{2.26}$$

with initial condition

$$a_p = Y_p/d_p \,. \tag{2.27}$$

The index i in (2.26) proceeds backwards from $i = p-1$ down to $i = 1$.

2.3 Lattice structure of LP coefficients

It is possible to convert any digital filter to a corresponding lattice filter [2] [3]. The prediction in the LP systems discussed above is based on p previous samples of $x[n]$, i.e. *forward prediction*. In lattice LP

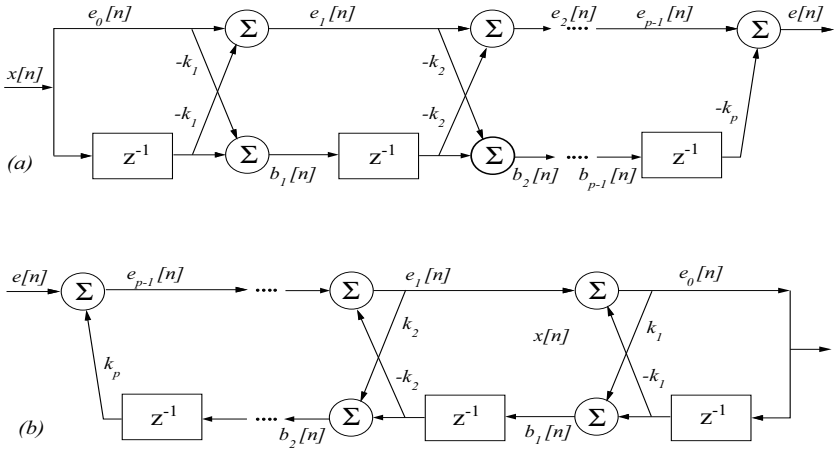


Fig. 2.2. Lattice Filters: (a) inverse filter $A(z)$, which generates both forward and backward error signals at each stage of the lattice; (b) synthesis filter $1/A(z)$.

model, the p ensuing samples are also used in a form of *backward prediction*. Figure 2.2 shows a general lattice structure which involves both the forward and backward prediction.

The lattice structure is a direct consequence of the Levinson-Durbin's recursion with the PACOR coefficients k_m. Applying (2.16c), we see that the prediction error sequence $e_m[n]$ can be expressed as

$$e_m[n] = e[n] * a_{m-1}[n] - k_m x_m * a_{m-1}[m-n]\,, \tag{2.28}$$

where * denotes convolution. The first term in (2.28) is the forward prediction error from an $(m-1)$th predictor and the second term is a parallel backward error. Lattice methods of linear prediction are order-recursive. That is, the optimal coefficients are first solved for the first stage of the filter, then the prediction error signals are computed for the next stage and so on. Assigning $b_m[n]$ to this backward prediction error yields a recursion formula:

$$e_m[n] = e_{m-1}[n] - k_m b_{m-1}[n-1]\,, \tag{2.29}$$

where

$$b_m[n] = x[n] * a_m[m-n] = \sigma_{l=n-m}^{n} x[l] a_m[m-n+l] \tag{2.30a}$$

$$= x[n-m] - \sum_{l=1}^{m} a_m[l] x[n-m+l]\,, \quad a_m[0] = 1 \tag{2.30b}$$

$$= b_{m-1}[n-1] - k_m e_{m-1}[n]\,. \tag{2.30c}$$

(2.29) and (2.30c) define the forward and backward prediction error sequences for an mth order predictor in terms of the corresponding prediction errors of an $(m-1)$th order predictor (see Figure 2.2, with initial conditions of $e_0[n] = b_0[n] = x[n]$). This block-based method is called *Burg Algorithm* [4]. In this method, the PACOR coefficients k_m can be directly related to the forward and backward prediction errors. The relationship in form of normalized cross-correlation function is

$$k_m = \frac{\displaystyle\sum_{n=0}^{N-1} e_{m-1}[n] b_{m-1}[n-1]}{\left[\displaystyle\sum_{n=0}^{N-1} (e_{m-1}[n])^2 \sum_{n=0}^{N-1} (b_{m-1}[n-1])^2\right]^{1/2}} \tag{2.31}$$

The reflection coefficients k_m have many interesting properties. In [5], these coefficients were derived directly from a non-uniform acoustic tube model, where the coefficients, as the name indicates, are reflection coefficients of individual tube elements. Therefore, the reflection coefficients and the lattice structure

have firm physical interpretations. Their goal was to find a representation for LP filter coefficients that is more robust to quantization. Reflection coefficients also act in a reasonable way in temporal interpolation of coefficients between frames. In addition, if all the reflection coefficients obey $|k_m| < 1, m = 1, 2, \ldots, p$, the synthesis filter is stable. Therefore, lattice methods of linear prediction also give direct means to check and guarantee the stability of the estimated model.

2.4 Determination of LP filter order

Finding an optimal LP filter order p is crucial to achieving optimal compression. It is not just a matter of decoder and encoder complexity because there is a tradeoff between the lower variance of the residual sequence and the increasing overhead due to larger predictor orders. On one hand, larger orders can capture the dynamics of a richer class of signals. On the other hand, larger orders also require proportionally larger data sets for the parameters to be accurately estimated and transmitted.

Obviously, determination of LP filter order is to find an order which minimizes the variance of zero-mean residual sequence, i.e.,

$$\sigma_e^2(p) = \frac{1}{N-p-1} \sum_{k=p}^{N-1} e^2[k] \,. \tag{2.32}$$

Simply, we might find an optimal LP filter order by incrementing p from $p = 1$ until the residual variance $\sigma_e^2(p)$ reaches a minimum. Another method, called the Akaike information criteria (AIC) [6], involves minimizing the following function

$$AIC(p) = N \ln \sigma_e^2(p) + 2p \,, \tag{2.33}$$

where $2p$ serves to penalize for unnecessarily high predictor orders. The AIC, however, has been shown to be statistically inconsistent, so the minimum description length (MDL) criterion has been formed [7] [8],

$$MDL(p) = N \ln \sigma_e^2(p) + p \ln N \,, \tag{2.34}$$

which implies to minimize the number of bits that would be required to describe the data. A method proposed by Tan [9] involves determining the optimal number of bits necessary to code each residual,

$$\mathcal{B}(p) = 2^{-1} \log_2 \sigma_e^2(p) \,. \tag{2.35}$$

In this case, p is increased until the following criterion is no longer true,

$$(N-p)\Delta B(p) > \Delta\beta(p) \,, \tag{2.36}$$

where $\Delta\mathcal{B}(p) = -[\mathcal{B}(p) - \mathcal{B}(p-1)]$ and $\Delta\beta(p)$ denotes the increase in overhead bits for each successive p. There are several other methods of order determination that are often used in practice include the Bayes information criterion (BIC) [10], which is equivalent to the MDL in many settings and the predictive least-squares (PLS) principle for sequential coding.

2.5 Design of polynomial LP filter

With smooth signals, piecewise polynomial signal model can be employed. Filters with polynomial responses arise e.g. in unbiased extrapolation of polynomial signals with maximal noise attenuation. The goal of polynomial prediction filter design is to design such FIR coefficients $h(k)$, $k = 1, 2, \ldots, p$, where p is FIR length, that a piecewise polynomial input signal is exactly predicted. Thereafter, noise gain of the FIR is minimized.

$$NG = \sum_{k=1}^{p} h(k)^2 \tag{2.37}$$

With the last input sample taken at time $n-p$, prediction of an input signal $x[n]$ is generally given by

$$x[n+i] = \sum_{k=1}^{p} h[k]\, x[n-k]\,. \tag{2.38}$$

A polynomial with order p can be found that passes through the previous p data points $x[n-1], x[n-2], \ldots, x[n-p]$. This polynomial can be evaluated at the nth sample time in a restrictive form of the linear predictor to obtain the predicted value $\hat{x}[n]$. In this respect, a simple adaptive prediction method using only integer coefficients was first proposed for lossless audio compression in Shorten [13]. For example, we can obtain the estimates,

$$\hat{x}_0[n] = 0 \tag{2.39a}$$
$$\hat{x}_1[n] = x[n-1] \tag{2.39b}$$
$$\hat{x}_2[n] = 2x[n-1] - x[n-2] \tag{2.39c}$$
$$\hat{x}_3[n] = 3x[n-1] - 3x[n-2] + x[n-3] \tag{2.39d}$$

These polynomials and the predicted values that they produce are illustrated in Fig. 2.3 for typical set of previous samples.

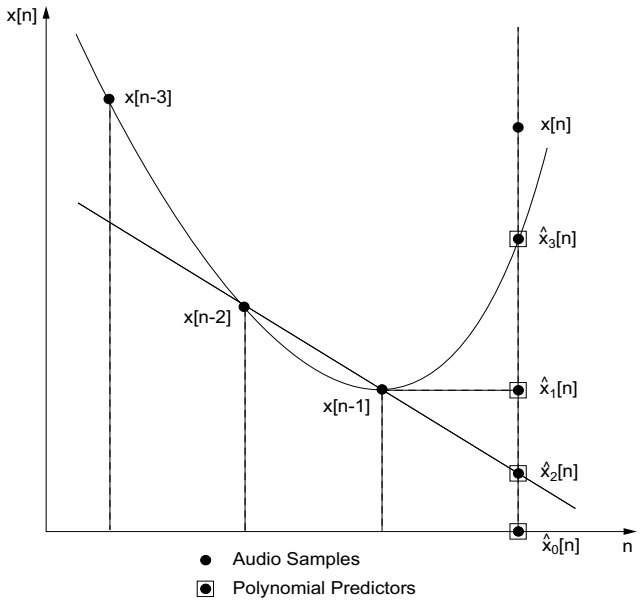


Fig. 2.3. Four polynomial approximations of $x[n]$.(After [14])

An interesting property of these polynomial approximations is that the resulting residual signals, $e_p[n] = x[n] - \hat{x}_p[n]$, can be efficiently computed without any multiplications in the following recursive manner:

$$\begin{aligned}
e_0[n] &= x[n] && \text{(2.40a)}\\
e_1[n] &= e_0[n] - e_0[n-1] \\
&= x[n] - x[n-1] && \text{(2.40b)}\\
e_2[n] &= e_1[n] - e_1[n-1] \\
&= (x[n] - x[n-1]) - (x[n-1] - x[n-2]) \\
&= x[n] - (2x[n-1] - x[n-2]) && \text{(2.40c)}\\
e_3[n] &= e_2[n] - e_2[n-1] \\
&= (x[n] - 2x[n-1] + x[n-2]) \\
&\quad - (x[n-1] - 2x[n-2] + x[n-3]) \\
&= x[n] - (3x[n-1] - 3x[n-2] + x[n-3]) && \text{(2.40d)}
\end{aligned}$$

The equations (2.40) mean also that the sum of absolute values is linearly related to the variance. This may be used as the basis of predictor selection. In our experiment, the four prediction residuals are computed at every sample in the frame, and the absolute values of these residuals are averaged over the complete frame. The residual with the smallest sum of magnitudes over all samples in the frame is then defined as the best approximation for that frame, and that predictor is used for that frame. The information on which predictor was used can be coded with two bits of information, and that becomes part of the overhead information for the frame.

3 Nonstationary Signal Modeling

The techniques and concepts discussed in previous sections are all based on an assumption about stationarity of the input signal. In practical LP algorithms, the filter coefficients are time-varying, i.e., parameters of a nonstationary signal model. Cremer [15] showed that Wold decomposition principle applies also to nonstationary signal models. However, there is no unique solution for the optimal time-varying coefficients $a_k[n]$. The coefficient evolutions must be restricted somehow in order to find one of the least-square optimal solutions to the coefficients. It is usually assumed that the signal is locally stationary, or the coefficients are smoothly time-varying [16]. These are conceptually two different approaches. The local stationarity assumption is used in conventional frame-based and continuously adaptive techniques. Smooth coefficient evolution is assumed in smoothness priors techniques [17], and in techniques where the coefficient evolutions are restricted to a class of functions which can be expressed as linear combinations of predefined basis functions [18]. The latter approach is sometimes called deterministic regression approach of time-varying autoregressive modeling.

3.1 General approaches

Frame-based filter design: The basic technique to obtain a nonstationary signal model is to perform linear predictive analysis in frames such that the signal is assumed to be stationary within each analysis frame. In a long time scale, this means that the signal model for linear predictive coding is actually given by

$$x[n] = \sum_{k=1}^{p} a_k[n]x[n-k] + e[n] = \hat{x}[n] + e[n]\,, \tag{3.1}$$

where filter coefficients $a_k[n]$ are now also functions of time n.

Audio and speech coding algorithms usually process the input signal in frames. For example, in LP-based speech coders, the frame-length is typically 10-20 ms. The frames are usually overlapping and, in the

case of the autocorrelation method, some window function is applied to each signal frame before analysis. The filter coefficients corresponding to each frame are coded and transmitted along with excitation data. A direct application of this procedure would produce discontinuities to the coefficient trajectories in frame borders, which may produce unwanted artifacts. It is a common practice to interpolate filter coefficients smoothly from one frame to another. Therefore, the signal model which is considered in these algorithms is essentially given by (3.1) even if the spectral model is estimated in locally stationary frames.

It is possible to increase the amount of overlapping in the analysis so that the coefficients are estimated more frequently. An extreme example is a sliding window formulation of linear predictive modeling where coefficients are solved at each time instant. However, this is computationally expensive and leads to an increased number of filter coefficients to be transmitted. Barnwell [19] has introduced a computationally efficient method for computation of adaptive autocorrelation. In this method, the correlation terms in (2.14) are computed recursively using a leaky integrator. This is a version of the autocorrelation method of linear prediction where the window function is actually defined as an impulse response of a low-order IIR filter.

Deterministic regression time-varying LP: It was proposed by Subba Rao [18] that the time-varying coefficient evolutions $a_k[n]$ could be expressed by

$$a_k[n] = \sum_{l=0}^{M} c_{kl}\phi_l[n]\,, \tag{3.2}$$

where $\phi_l[n]$ are a set of M predefined basis functions. For this system it is possible to formulate normal equations where the least squares optimal coefficients c_k can be solved directly. Typically, basis functions are some elementary mathematical functions such as the Fourier basis, Gaussian pulses, or prolate spheroidal sequences [20].

Adaptive filtering: While the method of frame-based linear prediction is effective, it suffers from the problem of finding a solution to the Yule-Walker equations, which becomes increasingly computationally expensive with large block sizes. Stochastic gradient methods for adaptive filtering also follow from a local formulation of the prediction problem. Here, the coefficients are not solved directly for a long signal frame but adjusted iteratively such that the filter coefficients converge towards optimal values. In this sense, these techniques are time-recursive. A classical example is least mean square (LMS) algorithm in which the coefficients of a direct form filter are adjusted using a simple gradient rule.

A backward adaptive formulation of linear predictive coding was introduced in [21] which is closely related to backward adaptive quantization methods presented in [22]. Here, the spectral model is not formed from the original input signal but from the already coded and transmitted signal. Since the same model can be computed at the decoder and the spectral model is completely estimated from the signal already transmitted, there is no need to code and transmit filter coefficients. Several different adaptive filtering techniques were compared in [23].

3.2 Normalized least mean square algorithm

Adaptive FIR filters using normalized least mean square (NLMS) have been proposed and used successfully [24] [25]. From (3.1) the signal $x[n]$ and the time-varying filter coefficients $a[n]$ can be represented by the column vector, i.e., $\mathbf{x}[n] = (x[n-1], \cdots, x[n-N])^{\mathbf{t}}$ and $\mathbf{a}[n] = (a_0[n], \cdots, a_{N-1}[n])^{\mathbf{t}}$. Then a time-varying prediction error is given by

$$e[n] = x[n] - \mathbf{a}^T\mathbf{x}[n]\,, \tag{3.3}$$

If two fixed parameters, a smoothing parameter β and a convergence parameter u, are specified, then $\mathbf{a}[n]$ can be computed iteratively as following

$$\mathbf{a}[n+1] = \mathbf{b}[n] + \mu[n]e[n]\mathbf{x}[n]\,, \tag{3.4}$$

where

$$\mu[n] = \frac{u}{\sigma_N^2[n]} \tag{3.5}$$

$$\sigma_N^2[n] = \beta\sigma_N^2[n-1] + (1-\beta)(e^2[n-1])\,. \tag{3.6}$$

Original signal can be exactly reconstructed by using the inverse of the algorithm

$$x[n] = e[n] + \mathbf{b^t x}[n]\,. \tag{3.7}$$

Eq. (3.7) shows that only $\mathbf{a}[0]$, $\mathbf{x}[0]$, and $e[n]$ are needed to reconstruct the original signal $x[n]$. This means that it is not necessary to transmit the coefficients $\mathbf{a}[n]$ and therefore to segment the signal in blocks. This algorithm requires less overhead than the standard LP method, but the coefficients should be updated very frequently.

3.3 Gradient adaptive lattice filter

In the gradient adaptive lattice (GAL) [26][27], the coefficients k_m are updated using the approximation of the error energy of the mth order predictor, i.e.,

$$k_m[n+1] = k_m[n] - \mu_m \frac{\partial \hat{E}_m(n)}{\partial k_m[n]}\,, \tag{3.8}$$

where μ_m are gradient weights and

$$\hat{E}_m(n) = e_m^2[n] + b_m^2[n]\,. \tag{3.9}$$

Applying the recursions after (2.29) (2.30c) for the current time index n leads to

$$\frac{\partial \hat{E}_m(n)}{\partial k_m[n]} = -2\big[e_m[n]b_{m-1}[n-1] + b_m[n]e_{m-1}[n]\big]\,. \tag{3.10}$$

From (3.8) and (3.10) we obtain a sample-by-sample update for the PACOR coefficients.

The gradient weights μ_m can be obtained by

$$2\mu_m = \frac{\alpha}{D_m(n)}\,, \tag{3.11}$$

where $D_m(n)$ is the expectation value of the sum of the forward and backward prediction error energies, i.e.,

$$D_m(n) = \lambda D_m(n-1) + (1-\lambda)\left(e_{m-1}^2[n] + b_{m-1}^2[n-1]\right),\ \text{with } 0 < \lambda < 1\,, \tag{3.12}$$

and the constant value α is normally chosen to $\alpha = 1 - \lambda$.

Due to the cascaded structure of a lattice filter, the GAL algorithm is both time-recursive and order-recursive. In practice, GAL algorithm is significantly faster in convergence than the conventional LMS algorithm. In adaptive filtering techniques, the gradient update rule can also be interpreted as a method to produce a recursive window function for linear predictive analysis.

4 Modified Linear Prediction Filter

As showed in previous sections, in traditional one-step forward linear prediction an estimate for the current sample value is formed as a linear combination of previous sample values, and the filter is assumed to be a conventional allpole filter. There are infinitely many alternative ways to form a linear combination of signal history and use it to predict the next signal value. The selection for sampling of signal history is not based on any mathematical necessity. An example of a modified formulation of the prediction principle which has been used in speech and audio application is frequency-warped linear prediction [28] [29]. Warped linear predictive coding is an alternative for conventional LP in speech and audio coding applications, especially for the perceptual audio coding system.

4.1 Generalized form of linear prediction filter

Recall that the classical linear prediction for a sample value $x[n]$ is given by

$$\hat{x}[n] = \sum_{k=1}^{p} a_x \, x[n-k] \, . \tag{4.1}$$

The z-transform of (4.1) is

$$\widehat{X}(z) = \left[\sum_{k=1}^{p} a_k z^{-k} \right] X(z) \, . \tag{4.2}$$

This scheme can be generalized by replacing the unit delay z^{-k} with allpass filter $D^k(z)$ to obtain

$$\widehat{X}(z) = \left[\sum_{k=1}^{p} a_k D^k(z) \right] X(z) \, . \tag{4.3}$$

In time domain one may write

$$\hat{x}[n] = \sum_{k=1}^{p} a_x d_k \big[x[n] \big] \, , \tag{4.4}$$

where $d_k \big[x[n] \big] = x[n-k]$. The mean square error of the estimate can now be written as

$$MSE = E \left\{ \left| x[n] - \sum_{k=1}^{p} a_k d_k \big[x[n] \big] \right|^2 \right\} \, , \tag{4.5}$$

where $E\{\cdot\}$ is expectation. A conventional minimization procedure leads to a system of normal equations

$$E\big\{ d_j \big[x[n] \big] d_0 \big[x[n] \big] \big\} \sim \sum_{k=1}^{p} a_k \, E\big\{ d_k \big[x[n] \big] d_j \big[x[n] \big] \big\} = 0 \tag{4.6}$$

with $j = 0, \ldots, p-1$. Because $D(z)^{-1} = D(z)^{-1}$ for an allpass filter, Parceval's theorem can be applied so that

$$\begin{aligned} E\big\{ d_{j+l} \big[x[n] \big] d_{k+l} \big[x[n] \big] \big\} &\sim \sum_{n=-\infty}^{\infty} d_{j+l} \big[x[n] \big] d_{k+l} \big[x[n] \big] \\ &= \frac{1}{i2\pi} \oint_C D(z)^{j+l} X(z) D(z^{-1})^{k+l} X(z^{-1}) \, \frac{dz}{z} \\ &= \frac{1}{i2\pi} \oint_C D(z)^{j+l-k-l} X(z) X(z^{-1}) \, \frac{dz}{z} \\ &= \sum_{n=-\infty}^{\infty} d_j \big[x[n] \big] d_k \big[x[n] \big] \sim E\big\{ d_j \big[x[n] \big] d_k \big[x[n] \big] \big\} \, , \end{aligned} \tag{4.7}$$

where k, j and l are any integers and $\sim$ indicates that the normalization of the expectation is omitted to simplify notation. The equation (4.7) states that the same correlation values appear in both terms of the left-hand side of (4.6). Therefore, (4.6) can be seen as a generalized form of the Wiener-Hopf equations, and the optimal coefficients a_k can be obtained by using, for example, the Levinson-Durbin algorithm just like in the conventional autocorrelation method of linear prediction.

4.2 Warped linear prediction (WLP)

A set of orthogonal polynomial functions are given by [30]

$$D(z)^k = \frac{\sqrt{1-|\lambda_k|^2}}{1-\lambda_k z^{-1}} \prod_{p=1}^{k} \frac{z^{-1}-\lambda_p}{1-\lambda_p z^{-1}} \,. \tag{4.8}$$

In equation (4.8), if $\lambda_k = \lambda_p = 0$ $(\forall k, p)$, this reduces to a traditional FIR filter, and if $\lambda_k = \lambda_p$ $(\forall k, p)$, this is the *Laguerre model* [31] which is a long tradition in the theory of signal processing. A simplified version of (4.8) given by

$$D(z)^k = \prod_{p=1}^{k} \frac{z^{-1}-\lambda_p}{1-\lambda_p z^{-1}} \tag{4.9}$$

is called a *frequency-warped filter*, and the corresponding modified LP scheme is called warped linear predictive coding (WLPC). From (4.7), the filter coefficients can be computed by the autocorrelation method of LP. In Fourier domain the equation (4.8) can be written as

$$\begin{aligned} D(e^{-i\omega}) &= \prod_{p=1}^{k} e^{-ip\left(\omega + 2\arctan\left(\frac{\lambda \sin(\omega)}{1-\lambda\cos(\omega)}\right)\right)} \\ &= \prod_{p=1}^{k} e^{-ip\nu(\omega)} \,, \end{aligned} \tag{4.10}$$

where $\nu(\cdot)$ is the frequency warping function. Therefore the spectral mapping is determined by the phase function of $D(z)^k$ which can be controlled by the value of λ.

The transfer function of a warped linear predictor is then given by

$$A(z) = 1 - \sum_{k=1}^{p} a_k D(z)^k \,, \tag{4.11}$$

where $D(z)$ is an allpass filter

$$D(z) = \frac{z^{-1}-\lambda}{1-\lambda z^{-1}} \,. \tag{4.12}$$

The current sample is estimated from samples of the frequency warped signal [32], which is formed by the outputs of the allpass filter chain. Figure 4.1(a) gives the structure of an WLP synthesis filter with order 2.

Note that this has a delay-free branch and is therefore not realizable in a practical implementation without modification. A possible modification is given Fig. 4.1(b). The coefficients g_0 and $b_1, \ldots, b_{p+1}$ can be calculated by using the following equations:

$$g_0 = \frac{1}{1+\sum_{k=1}^{p} a_k(-\lambda)^k} \tag{4.13a}$$

$$b_1 = \sum_{k=1}^{p} a_k(-\lambda)^k \tag{4.13b}$$

$$b_k = \sum_{k=l}^{p} a_k(-\lambda)^{k-l} - \sum_{k=l-1}^{p} a_k(-\lambda)^{k-l+2} \tag{4.13c}$$

$$\text{with} \quad l = 1, \ldots, p+1$$

A warped FIR lattice filter may be derived directly by replacing the unit delays of the conventional lattice structure with first-order allpass elements. This leads to the structure shown in Figure 4.2. The reflection coefficient of the warped structure can be computed from the estimated coefficients of a warped direct-form filter as in the case of a conventional filter.

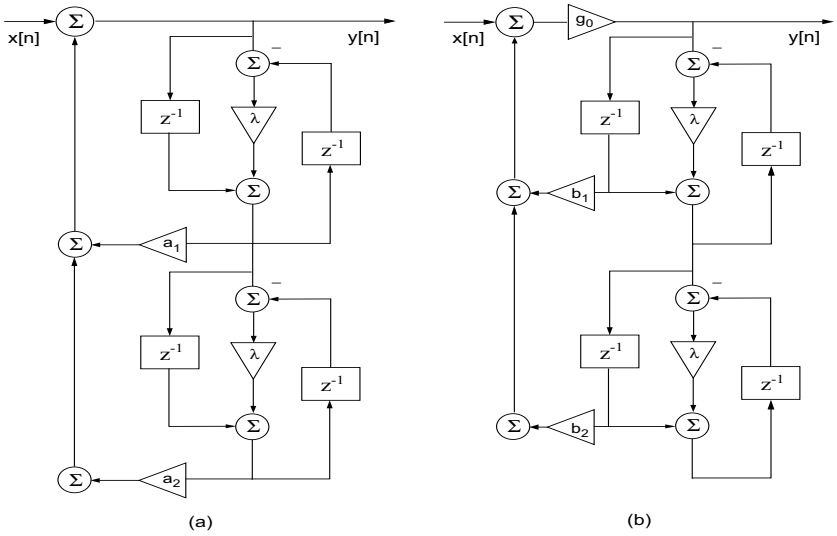


Fig. 4.1. (a) WLP synthesis filter, (b) realizable WLP synthesis filter.

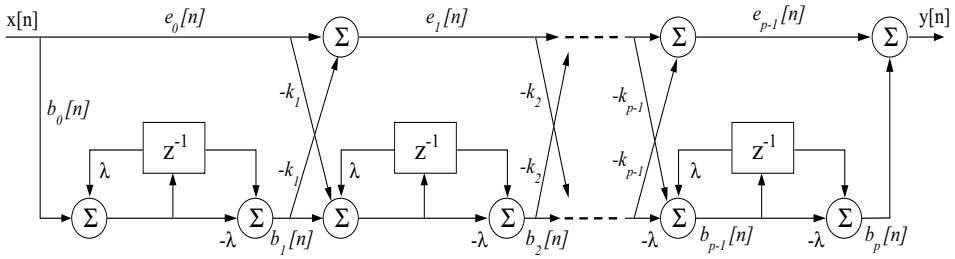


Fig. 4.2. A warped FIR lattice filter structure

Since WLP performs linear prediction in the warped frequency domain, the resulting auto-regressive process has different resolutions in modeling the spectral peaks at different frequencies. The frequency resolution of a WLP model is controlled by the allpass coefficient λ. This is advantageous in speech and audio signal processing because the frequency resolution in the spectral estimate is relatively close to the frequency resolution of human hearing [33]. The performance of conventional and warped LPC algorithms is compared in a simulated coding system using listening tests and bit rate [29].

5 Context-based Modeling

Statistical modeling of the source being compressed plays a central role in any data compression systems. Given a finite source sequence with length N, $x_1, x_2, \ldots, x_N$, the optimal code length of the sequence in bits is

$$H_{opt} = -\log_2 \prod_{n=1}^{N} p(x_n|x^{n-1}) \tag{5.1}$$

where x^{n-1} denotes the sequence $x_{n-1}, x_{n-2}, \ldots, x_1$. The key issue in context modeling for compression of the signal x^N is to estimate $p(x_n|X^{n-1})$, where X^{n-1} denotes a subset of the available past sequence x^{n-1} (causal template), with the assumption that the random process producing x^N is Markovian. The probability of the current symbol is conditioned by the set of past observations X^{n-1} in the form of the conditional probability mass function (pmf), $\hat{p}(x_n|X^{n-1})$, that serves as a statistical model of the source.

In classical universal context tree modeling [34] [35], for example, the different contexts of $\{X^{n-1}\}$ are grouped in a context tree, having as nodes as possible values of $\{X^{n-K}\}$, up to a depth level dictated by the affordable memory resources. The tree is grown as the message is encoded, according to the contexts seen so far. Histogram tracking is used for modeling the pmf to each node. The context algorithm can also be used for prediction, by predicting $\hat{x}_n = \arg\max_x p(x_n|X^{n-1})$.

The algorithm is optimal in encoding Markovian sources, but becomes extremely complex for large size alphabets, such as the audio signal sampled at 16 bits/sample or higher. In the case of large size alphabets, the Markov finite state machine (FSM) becomes extremely large,[4] incurring two major problems; the estimating of the conditional probabilities for each context,[5] and the excessive memory requirement to store the large number of all possible states (contexts). As a result, the context model can not learn the source statistics fast enough to estimate accurately the conditional probability distribution, leading to the so called "context dilution" [36]. Hence, the main challenge in audio compression applications is to find a compromise between the context size and compression performance and to mimic the main principles of context algorithm for sources with large size of the alphabet. Several different approaches to the reduced context size are proposed such as context quantization [37], histogram bucketing [38] [39], or context tree modeling [34].

5.1 Design of statistical prediction filter

An adaptive context algorithm for prediction modeling is proposed [40], using the combination of L-predictor [41] and the FSM context modeler. The L-predictor is defined as a linear combination of the order statistics

$$\hat{x}_n = \sum_{k=1}^{K} b_k X_k \ , \tag{5.2}$$

where X_k denote the k-th order statistics in the set $\{X_1, ..., X_K\}$. The conditional information $X^{n-1} = [X_1, \ldots, X_K]$ is obtained by ordering the x^{n-1} increasingly. This means that the prediction is itself the main feature extracted from the contextual information. The prediction error in this case is defined as

$$e_n = x_n - \hat{x}(n|X_1, X_2, \ldots, X_K) \, . \tag{5.3}$$

For the given alphabet interval $[0, M-1]$, (5.2) can be parameterized by setting $X_0 = 0$ and $X_{K+1} = M - 1$;

$$\hat{x}_n = a_0 + \sum_{k=1}^{K+1} a_k (X_k - X_{k-1}) \, . \tag{5.4}$$

This reparameterization in (5.4) has several advantages [40]; the flexibility of the prediction model, for example, that allows us to have a predictor with adapted properties at each different contexts. By selecting the parameters of the model at time n, (5.4) can be written as

$$\hat{x}_n = \hat{x}(n|w_n) = a_0(w_n) + \sum_{k=1}^{K+1} a_k(w_n)(X_k - X_{k-1}) \, , \tag{5.5}$$

where w_n denotes the suitable functions of the context at that time.

In [42], for example, two contexts are selected by context tree modeling and Hasse diagram. With these contexts the prediction value of current sample is computed using the adaptive linear prediction model similar to (5.5). First the contextual information is selected by a context mask containing the most recent N_c samples $x^{n-1} = [x_{n-1}, \ldots, x_{n-N_c}]$ and most recent N_e prediction errors $e^{n-1} = [e_{n-1}, \ldots, e_{n-N_e}]$, that is, the prediction errors from (5.3) are also used as part of the contexts. A recursive least squares (RLS) algorithm with forgetting factor w is used to perform the adaptive linear prediction;

$$\hat{x}_n = \sum_{k=1}^{p} w_{S_m^{(n-1)}}(k) \, x_{n-k} \, , \tag{5.6}$$

[4] e.g., given a sample value with Z bits resolution, there are 2^{ZK} different contexts.

[5] because even a very large signal does not provide sufficient samples to reach good estimates

where $S_m \in \{1, \ldots, N_m\}$ is the main context, obtained by using a tree classification procedure $S_m^{n-1} = \mathcal{T}(x^{n-1}, e^{n-1})$. A secondary context $S_s \in \{0, \ldots, N_s - 1\}$ is obtained by use of a Hasse diagram selection $S_s^{n-1} = \mathcal{H}(x^{n-1})$. The Hasse cube forms the state transition diagram of the finite state machine. Using the main and secondary contexts, the Eq. (5.6) can be extended with the intercept ρ,

$$\hat{x}_n = \rho_{S_m^{n-1}, S_s^{n-1}} + \sum_{k=1}^{p} w_{S_m^{n-1}}(k)\, x_{n-k}\,, \tag{5.7}$$

where the intercept depends both on main and secondary contexts, while the parameters $w_{S_m^{n-1}}(k)$ depend only on the main context.

5.2 Context-based error modeling

To achieve high compression ratio by entropy coding, e.g., by arithmetic or Golomb-Rice coding, the residual signal consisting of prediction errors should be similar to stationary white noise as much as possible. As a matter of fact, however, the residual signal has ample redundancy and especially is still nonstationary. The better the prediction, however, the more noise like prediction errors will be. An improvement of coding rate can be obtained by applying the context algorithm to make use of the dependency between the residuals. It is showed in [34] [43] that the context algorithm in conjunction with arithmetic or Huffman coding provides to produce good results in lossless image and audio compression.

In image compression, context algorithm for modeling of the prediction errors or the transform coefficients is highly studied in order to condition the distribution of the prediction errors [44] [43]. In these cases, the prediction errors are arranged into a predefined number of statistically homogeneous classes based on their spatial context, i.e., prediction-based context modeling. If such classes are statistically discriminated, then the zeroth-order entropy of a *context-conditioned* model of prediction errors will be lower than that derived from a stationary memoryless model of the decorrelated source.

The cost of context modeling is proportional to the number of the parameters in the statistical model and could then offset the entropy savings. It means also that a direct implementation of the classical universal context algorithm to the high quality audio, such as DVD audio sampled by 96 kHz at 20 or 24 bits/sample, does not provide a good solution. Although the universal algorithms are proven to be asymptotically optimal for stationary sources, the complexity of their underlying models will add an extra term to the best achievable coding rate. Hence a key objective in a context modeling scheme for high quality audio compression is to reduce the number of parameters defining the coding distribution at each context and the number of contexts.

6 Experimental Comparison of Prediction Filters

A block diagram of the lossless predictive coding system for experimentation is shown in 6.1. The linear prediction filters with different structures, i.e., FIR, IIR, lattice, and polynomial approximation (PAP) are tested with various block lengths and different filter orders. In addition, an efficient context-based error modeling (FIR-CM) is also experimented in conjunction with FIR linear predictor and Golomb-Rice coding.

6.1 Test audio materials

Nine audio materials are chosen for our experiment; six materials from SQAM-CD [45] [6] and two from published music CDs. All materials are sampled at 44.1kHz, 16 bits step size, and stereo channel (except for speech). There is provision to take advantage of dependency between the two stereo channels, but this does not provide a significant improvement except for the mono recorded test material Nr. 3 (female speech).

[6] SQAM(Sound Quality Assessment Material), European Broadcasting Union

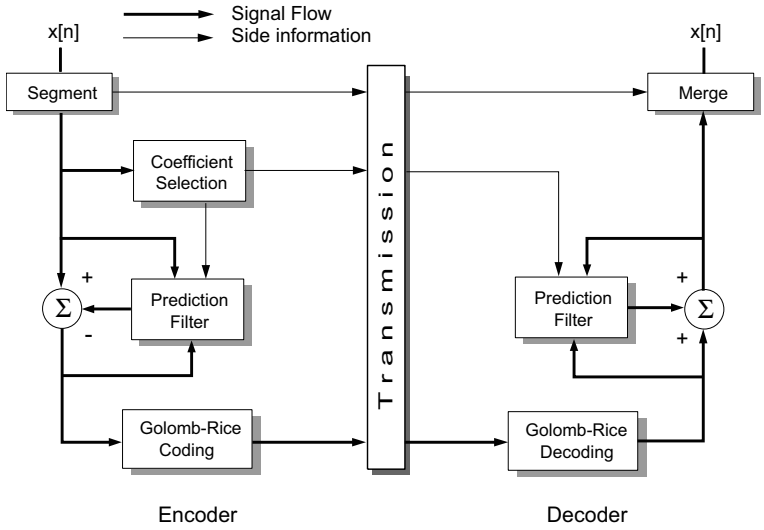


Fig. 6.1. Block diagram of lossless audio coding system for the experiments.

Nr.	Length	Description
1	1:11	SQAM Track 8, Violin, Arpeggio and Melodious Phrase
2	0:46	SQAM Track 13, Flute, Arpeggio and Melodious Phrase
3	0:21	SQAM Track 53, Female Speech, German (Mono)
4	1:32	SQAM Track 60, Piano, Schubert
5	1:22	SQAM Track 67, Wind Ensemble, Mozart
6	0:33	SQAM Track 69, ABBA, *Pop*
7	0:21	SQAM Track 70, Eddie Rabbitt, *Country*
8	0:29	Def Leppard "Adrenalize", Track 1 "Let's get rocked", Bludgeon Riffola Ltd, *Metal Rock*
9	0:29	Stan Getz "The Artistry of Stan Getz", Track 10 "Litha", Polygram Records, *Soft Jazz*

Table 1. Description of test audio materials

6.2 Description of compression system

Segmentation: The input audio signal is divided into the blocks (frames), and each block is treated separately during the coding operation. A simple rectangular windowing is used for the segmentation without overlapping between frames and padding operation. We experimented the compression system with various block lengths being a power of two, 2^N, with N between 8 to 12, i.e., 256, 512, 1024, 2048, and 4096 samples.

Prediction filters: Four types of the linear prediction filters, i.e., FIR (Fig. 2.1.a), IIR (Fig. 2.1.b), lattice (Fig. 2.2), and polynomial approximation predictor (PAP) with order $p \leq 4$ using Eq. (2.39), are tested to compare the efficiencies of the filters. Except for PAP, all filters are tested by the prediction order up to 10. The coefficients for FIR, IIR, and lattice filter are found using the standard Levinson-Durbin's recursion algorithm (2.16). First the autocorrelation coefficients of each block are computed and converted to prediction filter coefficients by using Levinson-Durbin algorithm.

Residual modeling and entropy coding: For entropy coding, we use Golomb-Rice code with fixed parameter k, since Golomb-Rice code is optimized for a block of signals having a Laplacian probability

density (double-sided exponential distribution), which is found to be a good approximation for the distribution of the prediction residual samples resulted from the decorrelation operations. The estimation of optimal parameter k is linearly related to the variance of the signal. The Laplacian distribution is defined by

$$p\,(x) = \frac{1}{\sqrt{2}\sigma}\, e^{\frac{-\sqrt{2}}{\sigma}|x|}\ , \tag{6.1}$$

where σ^2 is the variance of the distribution. An expectation of the absolute value of x can be given as following

$$\begin{aligned} E(|x|) &= \int_{-\infty}^{\infty} |x|\, p(x)\, dx && (6.2)\\ &= \int_{0}^{\infty} x\, \frac{\sqrt{2}}{\sigma}\, e^{\frac{-\sqrt{2}}{\sigma}x}\, dx && (6.3)\\ &= \int_{0}^{\infty} e^{\frac{-\sqrt{2}}{\sigma}x}\, dx\ - \left[x\, e^{\frac{-\sqrt{2}}{\sigma}x}\right]_0^{\infty} && (6.4)\\ &= \frac{\sigma}{\sqrt{2}}\,. && (6.5) \end{aligned}$$

Since the optimal parameter k means that half the samples lie in the range $\pm 2^{\,k}$, the code word length of integer n is optimal when $k+1$ bits are for probability 0.5 and $k+n+1$ bits for probability $2^{\,-(k+n)}$. It leads to that

$$\begin{aligned} \frac{1}{2} &= \int_{-2^k}^{2^k} p(x)\, dx && (6.6)\\ &= \int_{-2^k}^{2^k} \frac{1}{\sqrt{2}\sigma}\, e^{\frac{-\sqrt{2}}{\sigma}|x|}\, dx && (6.7)\\ &= -e^{\frac{-\sqrt{2}}{\sigma}2^k} + 1\,. && (6.8) \end{aligned}$$

Therefore, the expectation function for the parameter k is given by

$$\begin{aligned} k &= \log_2\Big(\log_e(2)\frac{\sigma}{\sqrt{2}}\Big) && (6.9)\\ &= \log_2\big(\log_e(2)\, E(|x|)\big)\ . && (6.10) \end{aligned}$$

In PAP implementation k is obtained using the Eq. (6.10). For the other schemes with FIR, IIR, and Lattice predictor, the expectation of residual samples $E(|e[n]|)$ is derived from σ, which is computed for the calculation of predictor coefficients.

Context-based error modeling (FIR-CM): It is common that the residual signal has still redundancy even after the decorrelation by prediction. In FIR-CM scheme, a context algorithm for error modeling is embedded into the standard FIR prediction scheme in order to make use of the dependency between the residuals. We used the similar context selection algorithm proposed in [42]. After the context indexing, the error remapping algorithm used in LOCO-I [43] is performed to improve the Golomb-Rice coding performance. The error signal $e[n]$ is remapped to positive integer using the revertible remapping;

$$e' = \begin{cases} 2e & \text{if } e \geq 0\,, \\ 2|e| - 1 & \text{otherwise}\,. \end{cases} \tag{6.11}$$

Materials	PAP[1]	FIR	IIR	Lattice	FIR-CM	Gzip
Nr. 1 violin, solo	7.56 (4096,3)[2]	7.46 (4096,3)	**7.23** (4096,10)	7.59 (4096,3)	7.59 (2048,9)	12.05
Nr. 2 flute, solo	**5.76** (4096,3)	6.12 (4096,2)	5.77 (4096,10)	5.86 (4096,3)	6.11 (2048,2)	11.04
Nr. 3 speech, fem.	6.02 (256,2)	6.06 (1024,5)	5.96 (1024,9)	6.26 (512,2)	**5.92** (1024,4)	8.62
Nr. 4 piano, solo	4.23 (1024,3)	4.51 (4096,2)	**4.13** (4096,5)	4.45 (4096,3)	4.39 (2048,2)	10.43
Nr. 5 classic, orch.	**5.73** (1024,3)	6.07 (2048,3)	5.92 (4096,4)	5.90 (4096,3)	5.79 (2048,4)	12.79
Nr. 6 pop, abba	7.13 (512,2)	7.05 (2048,5)	**6.87** (2048,7)	7.21 (1024,2)	6.93 (2048,3)	11.76
Nr. 7 country	6.38 (4096,1)	6.35 (1024,5)	**6.28** (1024,4)	6.57 (1024,1)	6.37 (1024,3)	9.65
Nr. 8 rock, metal	12.16 (512,2)	11.40 (2048,10)	**11.31** (4096,10)	12.23 (4096,2)	11.49 (2048,9)	14.86
Nr. 9 jazz, soft	7.64 (512,2)	7.89 (4096,2)	7.73 (4096,10)	7.80 (2048,2)	**7.57** (2048,2)	13.93
Average	6.96 (1792,2.3)	6.99 (2731,4.1)	**6.80** (3186,7.7)	7.10 (2786,2.3)	6.91 (1820,4.2)	11.68

[1] prediction order $p \leq 4$, for the other filters $p \leq 10$
[2] (block length, prediction order)

Table 2. Test results with compression rates (bits/sample)

6.3 Test results and discussion

Table 2 shows the test results for comparison of different prediction filters. The IIR prediction filter with large block length and higher prediction order has superior compression result for nearly all kind of audio samples over other prediction filters. Especially for the music samples with high treble energy and wide dynamic range (e.g., Nr. 6, 7, 8), IIR prediction scheme provides an acceptably stable compression ratio. It was clearly proven in our experiment that IIR prediction filter has more potential for improving the compression rate than FIR prediction filter, as discussed in [46]. As expected from its simple integer algorithm, PAP has lowest complexity with relative efficient compression performance. Using context-based error modeling, FIR-CM improved the compression rate 2-6% compared with that of FIR. In fact, this result of FIR-CM is less than we expected. Using all nodes in the Hasse diagram instead of middle layer nodes could have the potential of more improving the compression rate with several percents, but unfortunately with the cost of a twofold will increase in the overall complexity of the algorithm.

The choice of block length is an important consideration in implementation of most LP filter systems. In general, a longer block length may require an unreasonably high amount of computation and reduce the editibility. As showed in the results, for the nonstationary signal with highly localized statistical behavior like the speech (in our case Material Nr. 3) the block length should be small enough. Figure 6.2 shows the compression ratio with respect to the different block legnths. In our experiment, the efficient block length was between 2048 and 4096 samples for all LP filters.

The order of the prediction filter is also set as a tradeoff. For large predictor orders, the number of bits necessary to send the side information will be large, but the quality of the prediction will be better, and therefore the residuals will be encoded with a small number of bits. For small predictor orders, the side information will be encoded using a small number of bits, but encoding of residuals will require a large amount of bits. However, there are reasons for keeping the filter order as low as possible. It is not just a matter of decoder and encoder complexity, as the coefficients of the filter have to be transmitted periodically, possibly with initialization data for the associated state variables, and the overheads of transmitting the side information can be significant and increase with filter order. Figure 6.3 shows the compression rate with respect to the prediction orders. Compared with the case of the block length in Figure 6.2, the effi-

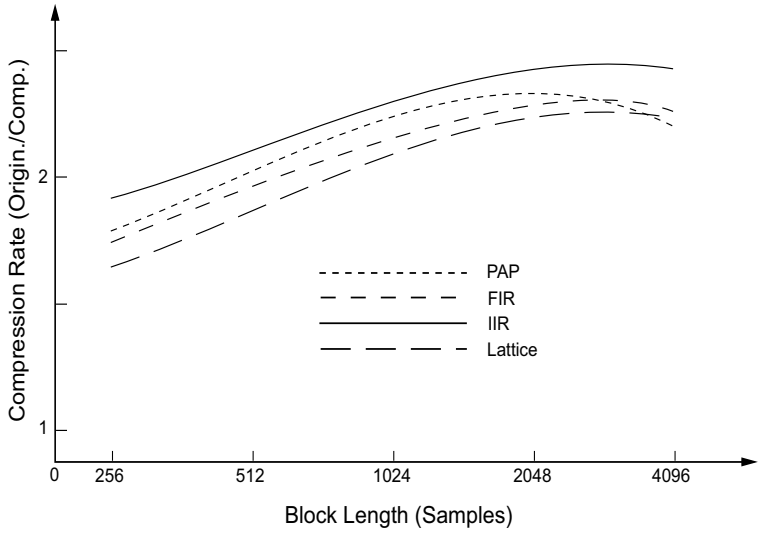


Fig. 6.2. Comparison between block length and averaged compression ratios of all test materials

cient order of the prediction filter varies with the characteristics of each prediction filter. In our experiment, predictor order of 6 to 10 provided the best results from IIR filter, and 2 to 4 from the others.

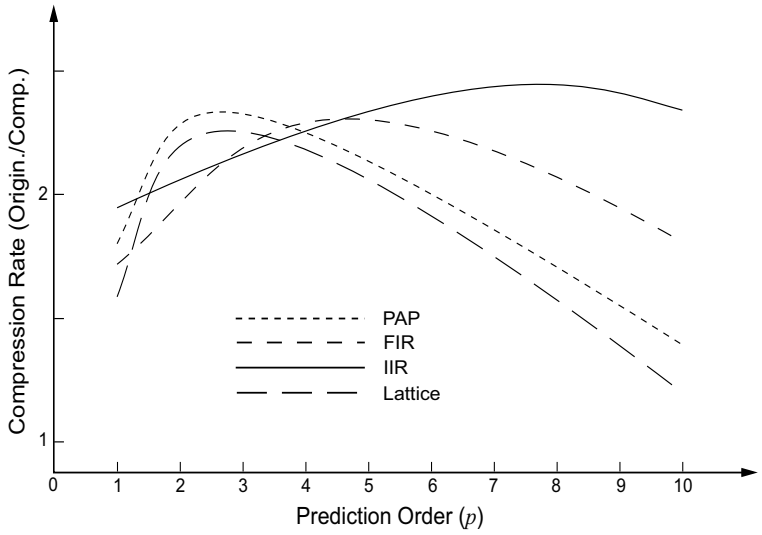


Fig. 6.3. Comparison between prediction order and averaged compression rate of all test materials

7 Conclusion and Future Work

We have touched on a range of issues in predictive modeling for lossless audio compression. In modeling a signal, it is of primary importance that the model be well approximated to the signal in question. Otherwise, the model will not necessarily provide a meaningful decorrelation performance in the compression scheme. The goal of the predictive modeling is to reduce the sample magnitudes by making a prediction of the current sample based on previous sample values. Various prediction filters that are characterized by their prediction structures and coefficients are compared by applying to lossless audio compression. To

explore the influence of the filter parameters on compression performance, various block lengths and prediction orders are tested for each prediction filter. Complexity is also an important factor for the evaluation of the prediction filters. With its lowest complexity, the polynomial predictor (PAP) provides efficient compression performance for certain of the audio materials. On the other side, applying the context-based error modeling into the compression scheme does not specifically improve the compression ratio compared to its very high complexity. It is clearly shown through the test results that the selection of the prediction filter and the parameters depends strongly on the audio signal content. In general, the lattice structure offers a powerful characterization in both filter design and implementation with the independently variable coefficients. However, for the compression applications, it may be not good choice because of the increasing side information and complexity. IIR prediction filter with large block length (2048 to 4096 samples) and high prediction order ($p = 10$) has superior compression result for nearly all kind of audio samples over other prediction filters. Furthermore, for the music samples with high treble energy and wide dynamic range, the IIR prediction filter provides an acceptably stable compression ratio. In contrast with the FIR filter, the IIR filter offers more possible ways to efficiently design or modify the filter structure because of its backward adaptation.

In fact, the works done in this paper and other state-of-the-art lossless audio coders show us that the lossless audio compression seems to be already reached its limit of compression ratio (i.e., max. 3 to 4), as often said in the literature, no matter whether using a simple prediction method or a complex transform to decorrelate the audio signal. However, there are still some points that could potentially pave the way to overcome the limit. First, parametric signal segmentation will improve compression performance. The choice of effective block length is a critical factor for designing signal compression system. In prediction method the decorrelation procedure is performed in each block separately. In this case, a cross correlation between the blocks is generally ignored. The musical signal is quasi-periodic signal because of its repeating rhythm and harmonic progress. If we segment the signal according to the period of rhythm and beat and take a cross correlation between the blocks into account, it is feasible to enhance compression ratios. As a starting point, one might incorporate a beat tracking stage into the compression system to estimate the parameters for the effective block length. Another possibility for improving prediction precision is to develop prediction method with blockwisely varying prediction order. As we know, efficient decorrelation of signal with high treble portion and wide dynamic range needs in general higher prediction order than the case of the smooth signal. Therefore it might be useful to estimate the prediction order for each block by calculating variance of the blocks.

References

1. A. Papoulis, “Maximun entropy and apectral estimation: A review,” *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 29, pp. 1176–1186, Dec. 1991.
2. F. Itakura and S. Saito, “On the optimum quantization of feature parameters in the PACOR speech synthesizer,” in *Proc. Conf. Speech Commun. and Processing*, pp. 434–437, 1972.
3. R. W. Schafer and J. D. Markel, *Speech Anaysis*. New York: Selected Reprints Series, IEEE Press, 1979.
4. J. Makhoul, “Linear prediction: A tutorial review,” in *Proceedings of the IEEE*, vol. 63, pp. 561–580, Apr. 1975.
5. B. Atal and S. L. Hanauer, “Speech anaysis and synthesis by linear prediction of the speech wave,” *Journal of Audio Eng. Soc.*, vol. 50, pp. 637–655, 1971.
6. H. Akaike, “A new look at the statistical model identification,” *IEEE Tran. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.
7. J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1978.
8. A. Barron, J. Rissanen, and B. Yu, “The minimum description length principle in coding and modeling,” *IEEE Trans. Inform. Th.*, vol. 44, pp. 2743–2760, Oct. 1998.
9. L.-Z. Tan, *Theory and Techniques for Lossless Waveform Data Compression*. PhD thesis, The University of New Mexico, 1992.
10. G. Schwarz, “Estimating the dimension of a model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.
11. J. Rissanen, “A predictive least squares principle,” *IMA J. Math. Contr. Inform.*, vol. 3, no. 2-3, pp. 221–222, 1986.
12. M. Wax, “Order selection for ar models by predictive least squares,” *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 36, pp. 581–588, Apr. 1988.
13. T. Robinson, “Shorten: Simple lossless and near-lossless waveform compression,” Tech. Rep. CUED/F-INFENG/TR.156, Cambridge University, UK, Dec. 1994.

14. M. Hans and R. W. Schafer, "Lossless compression of digital audio," *IEEE Sig. Proc. Mag.*, vol. 18, pp. 21–32, July 2001.
15. H. Cremer, "On some classes of nonstattionalry processees," in *Proc. 4th Berkeley Symp. Math. Statist. Probability*, vol. 2, (Berkeley, CA, USA), pp. 57–78, Univ. California Press, 1961.
16. M. B. Priestley, *Spectral Analysis and Time Series*. Vol. 2 of *Probability and Mathematical Statistics*, London: Academic Press Inc., 1981.
17. J. P. Kaipio and M. Juntunen, "Deterministic regression smoothness prior $t$var modelling," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. III, pp. 1693–1696, IEEE, Phoenix, Arizona, USA, 1999.
18. T. Subba Rao, "The fitting of non-stationary signals," *J. R. Statis. Soc.*, vol. B32, pp. 312–322, 1970.
19. T. Barnwell, "Recursive autocorrelation computation for (lpc) analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, (Hartford), pp. 1–4, 1977.
20. D. Slepian, "Prolate spheroidal wave functions, Fourier analysis and uncertainty-V: The discrete case," *Bell System Technical Journal*, vol. 57, no. 5, pp. 1371–1430, 1978.
21. J. D. Gibson, S. K. Jones, and J. L. Melsa, "Sequentially adaptive prediction and coding of speech signals," *IEEE Trans. Commun.*, vol. 22, no. 11, pp. 1789–1797, 1974.
22. N. S. Jayant, "Adaptive quantization with one word memory," *Bell System Technical Journal*, pp. 1119–1144, 1973.
23. J. D. Gibson, Y. C. Cheong, W.-W. Chang, and H. C. Woo, "A comparison of backward adaptive prediction algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 1, (Albuquerque, New Mexico), pp. 237–240, 1990.
24. S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ:Printice-Hall, 1996.
25. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Printice-Hall, 1985.
26. L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, (Hartford, USA), pp. 683–686, 1977.
27. S. J. Orfanidis, *Optimum Signal Processing, An Introduction*. Signapore: McGraw-Hill, second edition ed., 1990.
28. A. Härmä, U. K. Laine, and M. Karjalainen, "WLPAC-a perceptual audio codec in a nutshell," in *AES 102nd Conv.*, (Munich, Germany), p. Preprint 4420, 1997.
29. A. Härmä and U. K. Laine, "A comparison between frequency-warped and conventional linear predictive coding," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 9, pp. 579–588, July 2001.
30. B. Ninness and F. Gustafsson, "A unifying construction of orthogonal bases for system identification," *IEEE Trans. Automatic Control*, vol. 42, no. 4, pp. 515–521, 1997.
31. Y. W. Lee, *Statistical Theory of Communication*. New York: Wiley, 1960.
32. U. K. Laine, M. Karjalainen, and T. Altosaar, "WLP in speech and audio processing," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 3, pp. 349–352, 1994.
33. J. O. Smith and J. S. Abel, "Bark and ERB bilinear transform," *IEEE Trans. Speech, Audio Proc.*, vol. 7, pp. 697–708, Nov. 1999.
34. J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Th.*, vol. IT-29, pp. 656–664, Sept. 1983.
35. M. Weinberger, J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inform. Th.*, vol. IT-3, pp. 643–652, May 1995.
36. M. Weinberger, J. Rissanen, and R. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Proc.*, vol. IP-5, pp. 575–586, Apr. 1996.
37. X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Proc.*, vol. 6, pp. 656–664, May 1997.
38. S. Todd, G. Langdon, and J. Rissanen, "Parameter reduction and context selection for compressing of grey-scale images," *IBM J. Res. Develop*, pp. 188–193, Mar. 1985.
39. I. Tăbuş and J. Astola, "Adaptive Boolean predictive modeling with application to lossless image coding," in *SPIE-Statistical and Stochastic Methods for Image Processing II*, (San Diego, Califonia), pp. 234–245, Oct. 1997.
40. I. Tăbuş, J. Rissanen, and J. Astola, "Adaptive L-predictors based in finite state machine context selection," in *ICIP'97 International Conference on Image Processing*, (Santa Barbara), pp. 401–404, Oct 1997.
41. P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
42. C. D. Giurcăneanu, I. Tăbuş, and J. Astola, "Adaptive context based sequential prediction for lossless audio compression," in *Proc. Eusipco-98, IX European Signal Processing Conference*, (Rhodes, Greece), Sept. 1998.
43. M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A Low Complexity, Context-based Lossless Image Compression Algorithm," in *Proc. Data Compr. Conf.*, vol. 40, (Snowbird, Utah, USA), pp. 140–149, Oct. 1996.
44. X. Wu and N. Memon, "Context-based adaptive lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, April 1997.
45. "Sound Quality Assessment Material CD." Technical Centre of the European Broadcasting Union (EBU), 1988. 4222042.
46. P. Craven and M. J. Law, "Lossless Compression Using IIR Prediction Filters," in *102nd Convention of AES*, (Munich), p. Preprint 4415, Mar. 1997.