LOSSLESS AUDIO CODING

USING ADAPTIVE LINEAR PREDICTION

SU XIN RONG

(B.Eng., SJTU, PRC)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND

COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

ACKNOWLEDGEMENTS

First of all, I would like to take this opportunity to express my deepest gratitude to my supervisor Dr. Huang Dong Yan from Institute for Infocomm Research for her continuous guidance and help, without which this thesis would not have been possible.

I would also like to specially thank my supervisor Assistant Professor Nallanathan Arumugam from NUS for his continuous support and help.

Finally, I would like to thank all the people who might help me during the project.

TABLE OF CONTENTS

SUMMARY

Lossless coding of audio signals attracts more and more interests as the broadband services emerge rapidly. In this thesis, we developed a CODEC, using adaptive linear prediction technique for lossless audio coding. We successfully designed a cascade structure with independently adapting FIR filter in each stage for multistage adaptive linear predictors, which outperform other techniques, such as linear prediction coding (LPC) used in the state-of-the-art lossless audio CODEC. With the adaptive linear prediction, the coefficients of the filter need not to be quantized and transferred as side information, which is obviously an advantage of saving bits compared to LPC. Furthermore, due to the non-stationary of audio signals, it is necessary that the predictor should be adaptive so as to track the local statistics of the signals. Thus adaptive linear prediction technique is an attractive candidate for lossless audio coding.

Meanwhile, we analyze the characteristics and performance of the proposed predictor in theory and get the conclusion that this adaptive linear prediction outperforms the LPC in mean square error (MSE) performance. This is consistent with the simulation results that the prediction gain of the proposed predictor is better than the prediction gain of LPC. The challenge of using adaptive linear predictor is that the convergence speed of the adaptive algorithm must be fast enough so that the average prediction performance is promised.

Moreover, we also provide random access feature in the CODEC while the performance is still guaranteed, although the performance is much dropped by supporting random access due to the transient phase in adaptive linear prediction. In every random

access frame, separate entropy coding scheme is used for transient phase and steady state errors to solve the problem.

With the successful application of adaptive linear prediction for lossless audio coding, by now our CODEC outperforms most of the state-of-the-art lossless audio CODECs for most digital audio signals with different resolutions and different sampling rates.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Objectives

During many years, audio in digital format has played an important role in numerous applications. However with the constrained bandwidth and storage resources, such as internet music streaming and portable audio players, uncompressed audio signals must be a heavy burden. For example CD quality stereo digital audio with 44.1 kHz sampling rate and 16 bit quantization, will consume 1.41 Mbps bandwidth easily.

In response to the need of compression, much work has been done in the area of lossy compression of audio signals. Such as the MPEG Advanced Audio Coding (AAC) technology, can allow compression ratios to range up to 13:1 and higher. However, lossy audio coding algorithms get the high compression at the cost of quality degradation.

Obviously, the lossy audio coding technology is not suitable for applications which require lossless quality. These applications can be in recording and distribution of audio, such as distribution of high quality audio, audio data archiving, studio operations and collaborative work in professional environment. For these applications, lossless audio coding, which enables the compression of audio data without any loss, is the choice. For example, with the lossless audio coding technology, the Internet distribution of exact CD quality music becomes possible. It may not be accepted for customers to use their high-fidelity stereo system to play the AAC or MP3 music.

With the continuing growth of capacities of storage devices, bandwidth of internet and emergence of broadband wireless networks, it can be expected that lossless compression technology will be used much wider in the future. Therefore, recent years more and more interests have been focused on this technology. However, compared with the area of lossy coding, much less work has been done for lossless audio coding. And to make an international standard also becomes necessary.

The standardization body ISO/IEC JTC 1/SC29/WG11, known as the Moving Pictures Experts Group (MPEG) has started to work on defining lossless audio coding technology for ISO/IEC 14496-3:2001 (MPEG-4 Audio) standard [1]. They have issued a Call for Proposal (CfP) on lossless audio compression [2] in July 2002. The CfP requires high lossless compression efficiency for PCM audio signals at sampling rates from 44.1 kHz to 192 kHz and word lengths of 16, 20 and 24 bits. Moreover, the CODEC is also required to provide means for editing, manipulations and random access to the compressed audio data.

Considering the increasing application and MPEG's CfP of lossless audio coding, this project is to develop an efficient lossless audio CODEC which should outperform most of the state-of-the-art CODECs and make contributions to MPEG-4 standardization activities.

1.2 Major Contributions of the Thesis

In this project, we developed a lossless audio CODEC, with high compression performance for audio signals with sampling rates up to 192 kHz and resolutions up to 24 bits. Moreover, the proposal has been submitted to MPEG for evaluation in Oct. 2004.

The major contributions of this thesis are as follows:

1) Digital audio signal (low and high sampling rate) modeling techniques with adaptive filters in cascade structure;
2) Theoretical study of the characteristics of the cascaded adaptive linear predictor for audio signals;
3) Theoretical study of the performance bound of the cascaded adaptive linear predictors;
4) Successful application of the novel cascaded adaptive linear prediction technique in lossless audio coding;
5) Techniques to improve the compression performance in Random Access coding with the adaptive linear prediction technique.

With above efforts, the proposed CODEC can obtain higher compression ratio than most of the state-of-the-art CODECs for MPEG-4 test audio signals by now.

1.3 Organization of the Thesis

The following chapter reviews the background of lossless audio coding, including fundamentals of source compression, basic principles of audio coding, the entropy coding algorithms (Rice and Block Gilbert-Moore Coding) and linear prediction technique which is widely used in audio and speech coding. Two state-of-the-art lossless audio coding systems will be reviewed as well.

The structure overview of the proposed lossless audio coding system will be described in Chapter 3. Among all of the parts in the structure, this thesis focuses on the predictor mainly, which is discussed in Chapter 4. We propose the adaptive linear prediction technique which will be discussed in detail. The adaptive prediction filters in cascade structure will be used as the adaptive linear predictor for audio signals.

For wider and more practical applications, the feature of random access to the compressed audio signals is required by the CfP of MPEG. In Chapter 5, random access (RA) will be discussed in detail and implemented successfully in the proposed audio CODEC. With the adaptive linear prediction, we make the pioneer contribution to this topic in lossless audio coding. Finally, a conclusion of the thesis is given in Chapter 6, with recommendations for future work.

CHAPTER 2

BACKGROUND

This chapter will review the background of lossless audio coding, including some fundamentals of source coding, basic principles of audio coding, linear prediction coding techniques and several entropy coding algorithms.

At the end of this chapter, two state-of-the-art lossless audio coding systems will be briefly discussed. One is Monkey's audio coding [3], which is taken as a benchmark in MPEG's CfP [2]. Another is from Technical University of Berlin (TUB) [4], which is chosen as a reference model for MPEG-4 Audio Lossless Coding (ALS), attaining working draft status in July 2003.

2.1 Digital Audio Signals

In this thesis, the source signals discussed are the audio signals in digital format. During the last decades, analog signal processing has been replaced by digital signal processing (DSP) in many areas of engineering due to the development of digital techniques. In the real world, the physical audio signal is in analog format. Therefore the real signal must be converted to digital data format before processing, which is called analog-to-digital (A/D) conversion.

Fortunately, Claude Shannon had developed a theory which points out that a signal band limited to w Hertz can be exactly reconstructed from its samples when it is periodically sampled at a rate $f_s \geq 2w$ [5].

Human hearing's sensitive range is between 20 Hz and 20 kHz. That is why the sampling rate 44.1 kHz and 48 kHz are most commonly used currently as the sampling rate in high fidelity audio applications, e.g. the CD quality music is sampled at 44.1 kHz. However, with the requirement increasing for digital audio quality MPEG's CfP requires that the proposed CODEC should be able to compress high quality audio data which is sampled at rate from 44.1 kHz to 192 kHz.

During the process of A/D conversion, sampling is the first step. Meanwhile, the amplitude of each sample must be presented with a number of bits. This process is called quantization. Clearly, the number of bits used for each sample determines the quality of digital audio. The more bits are used, the better quality. The quantization resolutions considered are 16, 20 and 24 bits.

In practice, Pulse Code Modulation (PCM) is always used with quantization. That is to present each pulse with a number of bits after normalizing the amplitude. For example, the wave format audio is the PCM data converted from physical audio source.

In conclusion, the source data concerned is the PCM digital audio signal, with sampling rate from 44.1 kHz to 192 kHz, resolution 16, 20 and 24 bits. In general, the mathematical model of digital audio signal $x(n)$ can be given by

$$x(n) = \sum_i A_i(n)\cos(w_i n + \varphi_i) + \varepsilon(n) \tag{2.1}$$

where A_i is the amplitude envelope, φ_i is the phase of each frequency w_i and $\varepsilon(n)$ is the noise.

2.2 Lossless Data Compression

Lossless Data Compression, however, is not a new topic. There are many excellent algorithms in this area, such as Huffman Coding, Arithmetic Coding and Lempel-Ziv Coding [6]. These algorithms are widely used to compress text files, and proved to be very effective for text data.

Shannon's entropy theorem in [5] shows the smallest number of bits needed to encode the information. Let Q be the set of the symbols output by an n bit quantization. The entropy of this source is defined as

$$H(Q) = -\sum_i p_i \cdot \log_2 p_i \tag{2.2}$$

where p_i is the probability of symbol i, $i \in Q$.

The entropy theorem gives the bound for data compression. The problem of data compression is to encode information with as few bits as possible, e.g. to associate shorter codewords to messages of higher probability. In section 2.3.3, we will discuss an example of entropy coding, namely, Rice Coding, because it is widely used in lossless audio coding.

However, applying entropy coding methods directly to the audio signal is not efficient due to the long time correlations in audio signal. Therefore, it is necessary to design coding algorithms specifically for digital audio signals.

2.3 Lossless Audio Coding

2.3.1 Basic Principles

It is well known that conventional lossless compression algorithms (e.g. Huffman Coding) always fail to compress audio signal effectively, because of the large source alphabet and long term correlation of the audio samples. In recent years, a number of new algorithms have been developed for lossless audio coding [7]. All of the techniques are based on the principle of first losslessly reducing the long term correlation between audio samples and then encoding the residual error with an efficient entropy code. Fig. 2.1 shows the scheme for compressing audio signal.

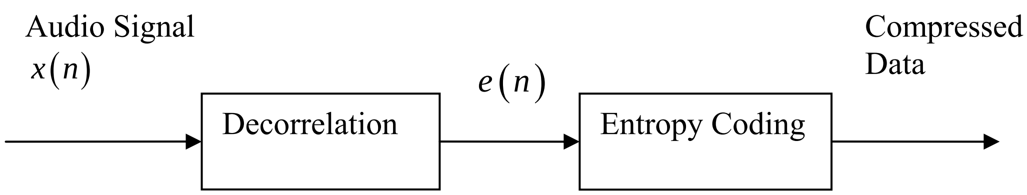


Fig. 2.1: The principle of lossless audio coding

For intra-channel de-correlation, there are two basic approaches, which remove redundancy by de-correlating the samples. The most popular method is to exploit the correlation between samples by using some type of linear predictor [3, 4, 8-12]. Another approach is to use linear transform, where the audio input sequence is transformed into the frequency domain. This method always plays a role as a bridge between lossless and lossy audio coding. The idea is to obtain the lossy representation of the signal, then losslessly compress the difference between the lossy data and the original signal [13-16]. In this thesis, we will only focus on the first approach, i.e. linear prediction for de-correlation. The concept will be discussed in section 2.3.2.

After de-correlation, some proper entropy coding is applied to further reduce the redundancy of the residual signal. Entropy coding is a process to convert symbols into bit streams according to a probability distribution function (pdf). Good compression performance will be expected if the estimated mathematical pdf is close to the true pdf of the signal. In section 2.3.3, Rice coding will be introduced.

2.3.2 Linear Prediction

It is well known that linear prediction is widely used in speech and audio processing [17] [18]. It is used to predict a value using the preceding samples in the time domain. For example, the signal sequence is $x(n), x(n-1), x(n-2), \cdots, x(n-N)$, the linear prediction of $x(n)$ which is at instance n, can be given as

$$\hat{x}(n) = \sum_{k=1}^{N} w(k) x(n-k) \tag{2.3}$$

where $w(k)$ is the coefficient of the linear predictor, which is always determined by the criterion of Minimum Mean Square Error (MMSE). Then we get the residual error $e(n)$ as

$$e(n) = x(n) - \hat{x}(n) \tag{2.4}$$

It has been found that most speech and audio signals are Laplacian distributed [19] [20]. Therefore the residual signal of the linear predictor is still Laplacian distributed, i.e. residual signal $e(n)$ can be approximately modeled with a Laplacian distribution. The probability density function (pdf) of Laplacian distribution is given as

$$p(f) = \frac{\lambda}{2} e^{(-\lambda|f|)} \tag{2.5}$$

2.3.3 Entropy Coding

We discuss a widely used entropy coding, Rice coding in this section. Rice coding [21] is a special case of Golomb coding [22] for data with a Laplacian probability distribution function. As the prediction residual signal $e(n)$ is Laplacian distributed, Rice coding is efficient, thus it is widely used in this application [3, 4, 8, 9, 14, 23, 24].

The idea of Rice coding is to decompose the code (the signed integer residual in lossless audio coding) into 3 parts:

1. One sign bit.
2. Lower part with length L bits.
3. Higher part presented with a series of 0s and terminated by 1.

We note that Rice coding is characterized by one parameter L. The sign bit can be 1 for negative, 0 for positive. If the code value is n, the lower part is the L least significant bits of n. In the higher part, the number of 0s is equal to the result by truncating the L least significant bits from n. Denote the number of 0s by N_h, which can be calculated as follows

$$N_h = n \gg L \tag{2.6}$$

where operator $\gg$ is the operation of bit shift. The parameter L is found by means of a full search, or estimated by the following equation, first given in [23]

$$L = \log_2\left(\ln(2)E\left(|e(n)|\right)\right) \tag{2.7}$$

where $E\left(|e(n)|\right)$ is the expectation of the absolute value of $e(n)$.

Table 2.1 gives the examples of Rice coding with $L = 4$.

Table 2.1 Rice Coding Example for $L = 4$

Number $e(n)$	$\lvert e(n)\rvert$ **in binary**	**Sign bit**	L **lower bits**	**Number of 0s**	**Full code**
0	0	0	0000	0	00001
-20	10100	1	0100	1	1010001
50	110010	0	0010	3	000100001

2.4 State-of-the-art Lossless Audio Coding

2.4.1 Monkey's Audio Coding

Monkey's Audio Coding has high compression ratio, which is therefore taken as a benchmark in MPEG's CfP. In its extra high mode, it adopts 3-stage predictor [3]. The first stage is a simple first-order linear predictor. Stage 2 is an adaptive offset filter. Stage 3 uses neural network filters. To reduce the redundancy of residual error further, Rice coding is used for entropy coding.

Because the neural network algorithm is used to adapt the coefficients, a long input sequence is needed to complete the learning process while encoding. This results in high complexity, moreover random access feature is not supported.

2.4.2 TUB ALS

This coder developed by TUB, chosen as the reference model for MPEG-4 ALS, is based on the Linear Prediction Coding (LPC) technique and Rice coding [4] [25].

In the current version, two alternative entropy coding schemes are available to process the prediction residual. The first scheme uses simple and fast Rice codes, while the second one employs block Gilbert-Moore codes (BGMC) [26] together with Rice coding, which offers an improvement in compression at the expense of increased processing time. Similar to TUB's CODEC, the proposed coding system provides the two alternative coding schemes as well. We use the latter coding scheme for our simulations, since it gives better compression performance.

As for the LPC predictor, the Durbin-Levinson algorithm is used for coefficients calculation [27] and decoding is straightforward with the coefficients quantized and transmitted. In general, it processes high compression and moderate complexity.

However, we find that LPC technique is not the optimal prediction solution in lossless audio coding. Moreover, using LPC the bit-stream must contain quantized LPC coefficients. Therefore we propose an adaptive linear predictor to replace LPC in lossless audio coding.

CHAPTER 3

OVERVIEW OF THE PROPOSED ALS SYSTEM

In the current MPEG-4 ALS CODEC, LPC is used to reduce the bit rate of audio clips in PCM format [2] and the Levinson-Durbin algorithm is used to find the optimal linear predictor according to the MMSE criteria. It is well known that the longer the linear predictor, the smaller the mean square error (MSE) of the predictor. However, the estimated optimum predictor coefficients for each block of input sequence should be quantized and transmitted as side information. Thus, the performance of this kind of CODECs in terms of compression ratio is trade-off between the prediction order and the MSE.

To overcome the drawback of LPC, an adaptive linear predictor is used because this sort of CODEC need not transmit the prediction coefficients, thus they can construct a high-order FIR filter to model more accurately the ample and harmonic components of general audio signals than the relative low-order linear prediction coding technique. In this thesis, we propose a stable adaptive linear predictor, which leads to a better compression ratio compared to that of the TUB optimal CODEC which is with high predictor order.

3.1 Big Picture

An overview of the proposed encoder is depicted in Fig. 3.1 and each part is described in the following sections. Fig. 3.2 is the overview of the corresponding decoder, which reconstructs the original signal perfectly using the same adaptive prediction algorithm as in the encoder. Therefore, the complexity of the adaptive predictors in both encoder and decoder are identical.

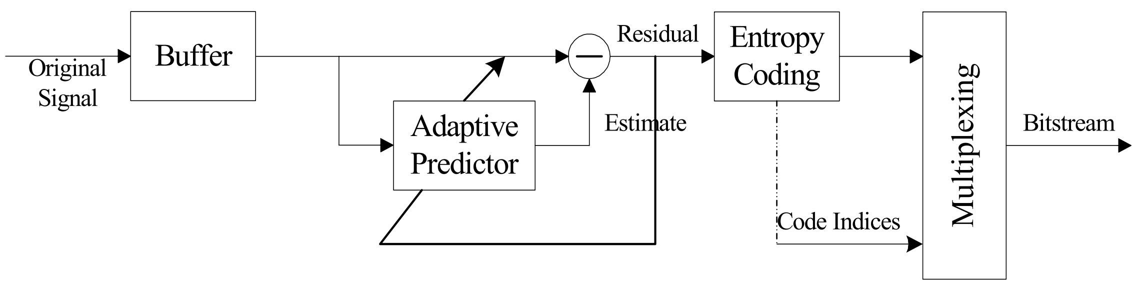


Fig. 3.1: Lossless audio coding encoder

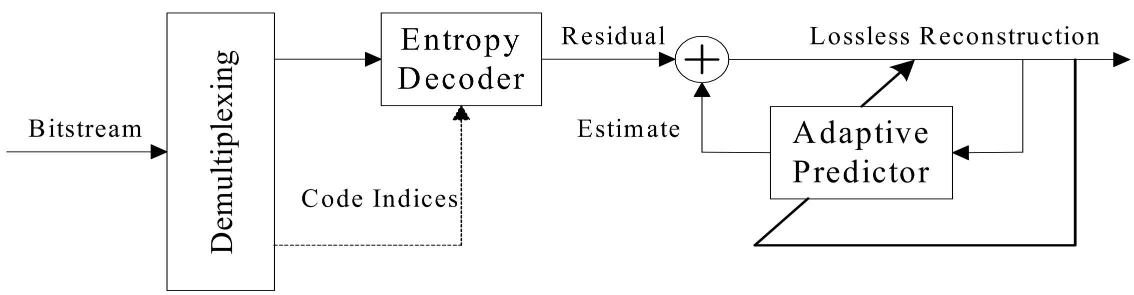


Fig. 3.2: Lossless audio coding decoder

3.2 Framing

First of all, the input signal of adaptive linear predictor is operated by framing, i.e. the input sequence is processed block by block. The framing operation is an important property for audio CODECs and necessary for most applications where it is required to quickly and simply access or edit the compressed audio bit stream. For example, the framing is required for random access, which will be discussed in detail in Chapter 5.

3.3 Adaptive Linear Predictor

Many audio signals, like music which is of the most interesting in lossless audio coding, contain abundant tonal and harmonic components. It requires a large predictor order to reduce the energy and correlation of the signal effectively. The adaptive linear predictor should be an ideal candidate for this requirement because its coefficients need not to be contained and transmitted in bit stream.

Moreover, considering the non-stationary property of audio signals, an ideal predictor should be adaptive and possess tracking capabilities to capture the local statistics of the signal, so that high prediction gain can be obtained.

Therefore we propose the adaptive linear predictor in the system for audio lossless coding. However lots of methods are available to design the adaptive predictor. In this thesis, we will discuss some adaptive filter algorithms, such as Least Mean Square (LMS) and Recursive Least Square (RLS) algorithms.

While designing and implementing the adaptive linear predictor, the random access function is also considered. We will discuss some solutions for this issue in a separate chapter focused on Random Access (RA).

3.4 Entropy Coding

In almost all of the coding systems, some kind of entropy coding is employed to reduce the redundancy and energy of residual signals after prediction. As discussed in Chapter 2, Rice coding is a popular entropy coding algorithm for this application.

However, a more efficient and complex entropy coding scheme is applied in the proposed coding system, namely, Block Gilbert-Moore Codes (BGMC), which works together with Rice coding [25].

CHAPTER 4

ADAPTIVE LINEAR PREDICTOR

We will study and design an optimal adaptive linear predictor, which outperforms the LPC predictor for lossless audio coding.

It is well known that the original digital audio signal is generally compressible because it possesses considerably high redundancy between samples. That is, the samples are highly correlated and non-uniformly distributed. Most lossless audio coding algorithms employ a pre-processor to exploit and remove the redundancy between signal samples, and then code the output or residual signal with an efficient entropy coding scheme [7]. In such a coding approach, the pre-processor is a predictor, which plays a dominant role in lossless audio coding. In general, better prediction results in higher compression performance.

Obviously, to achieve optimal compression performance, the predictor should be designed to remove correlation of the signal as much as possible so that the resulting prediction residual error can be coded at the lowest possible rate. We have discussed that in most coding systems, the digital audio signals are described by some sort of parametrical model, e.g. the Laplacian distribution. For such a model, the optimal predictor can be designed based on the least mean square criterion, so that the output generated has the smallest variance. The low complexity solution, which is already widely used in this area, is LPC technique based on Levinson-Durbin algorithm. However, the coefficients of LPC have to be quantized and transmitted as side information. For bit savings, a trade-off must be made between predictor order and mean square error (MSE), i.e. the length of order is limited in LPC. However, considering the characteristics of audio signals, a high-order predictor is always needed to reduce the large energy effectively.

Therefore, instead of LPC, adaptive linear predictor seems a good alternative, which does not need to transfer coefficients, promising potential bit savings and high predictor order. Furthermore, as the audio signals are non-stationary, it is necessary that the predictor should be adaptive and is capable to track the local statistics of the signals. A number of adaptive algorithms can be used to design an adaptive linear predictor such as the Least Mean Square (LMS) algorithm and the Recursive Least Square (RLS) algorithm. The LMS is widely used in practical application due to its robustness, efficiency and low complexity. However, the LMS suffers from slow convergence speed for highly correlated input signals with large eigenvalue spread, which leads to poor prediction performance. Although the RLS is much less sensitive to the eigenvalue spread of the input, its considerable complexity makes it impractical to be applied in a high-order predictor.

The LMS algorithm is an attractive candidate for the adaptive linear predictor. Several methods have been proposed to improve the convergence performance of the LMS algorithm. Most of them adopt a two-step approach, where the input is de-correlated using either a suitable transform or an adaptive pre-whitener before the LMS filter. Examples include the frequency domain based FFT-LMS and DCT-LMS adaptive filters [27], improving the convergence at the cost of large misadjustment of the filter coefficients and complexity. In the time domain, an FIR cascade structure with independently adapting and low-order LMS filter in each stage, has been reported for speech prediction [28].

In this chapter, we present a cascade structure, with an independently adapting FIR filter in each stage, to counteract the slow convergence problem. Moreover, the proposed structure exhibits lower overall MSE which results in better prediction gain than LPC. Although any adaptive FIR can be applied in each stage, e.g. the RLS can be used in low-order stage, for simplicity and stability, we use the LMS in every stage in our study.

4.1 Review of Adaptive Filter Algorithms

Before we study the adaptive linear predictor, let us review the widely used RLS and LMS algorithms in this section. With $x(n)$ denoting the input to the predictor, the residual error $e(n)$ of the RLS or LMS predictor is given by

$$e(n) = x(n) - \mathbf{w}^T(n)\mathbf{x}(n) \tag{4.1}$$

where T denotes matrix transposition, $\mathbf{x}(n) = \left[x(n-1), x(n-2), \cdots, x(n-N)\right]^T$, and the filter tap weights $\mathbf{w}(n) = \left[w_1(n), w_2(n), \cdots, w_N(n)\right]^T$.

With the RLS algorithm, the filter weights $\mathbf{w}(n)$ are updated as follows,

$$\mathbf{K}(n)=\frac{\lambda^{-1}\mathbf{Q}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^{T}(n)\mathbf{Q}(n-1)\mathbf{x}(n)} \tag{4.2}$$

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mathbf{K}^{T}(n)e(n) \tag{4.3}$$

and

$$\mathbf{Q}(n)=Tri\left\{\lambda^{-1}\mathbf{Q}(n-1)-\lambda^{-1}\mathbf{K}(n)\mathbf{x}^{T}(n)\mathbf{Q}(n-1)\right\} \tag{4.4}$$

where λ is a positive real-valued constant that is slightly smaller than 1, and the operator $Tri\{\ \}$ signifies the calculation of the upper or lower triangular part of the matrix $\mathbf{Q}(n)$ to improve the computational efficiency of this algorithm. Initialize the algorithm by setting $\mathbf{Q}(0)=\delta^{-1}\mathbf{I}$ and $\mathbf{w}(0)=\mathbf{0}$, where δ is a small positive real-valued constant.

With the LMS algorithm, the filter weights $\mathbf{w}(n)$ are updated as follows,

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\frac{\mu}{1+\mathbf{x}^{T}(n)\mathbf{x}(n)}\mathbf{x}(n)e(n) \tag{4.5}$$

where $0<\mu<2$ is the adaptation step size of the LMS algorithm.

When we use the RLS or LMS algorithm in audio signal de-correlation, we need to choose the proper parameters, μ for LMS, λ and δ for RLS. According to the principles of LMS and RLS algorithm, these parameters should be selected properly, based on the statistical properties of audio signals.

4.2 The Cascade Structure

In this thesis, we study a cascade structure for the adaptive linear predictor, with an independently adapting filter, e.g. an LMS filter, in each stage.

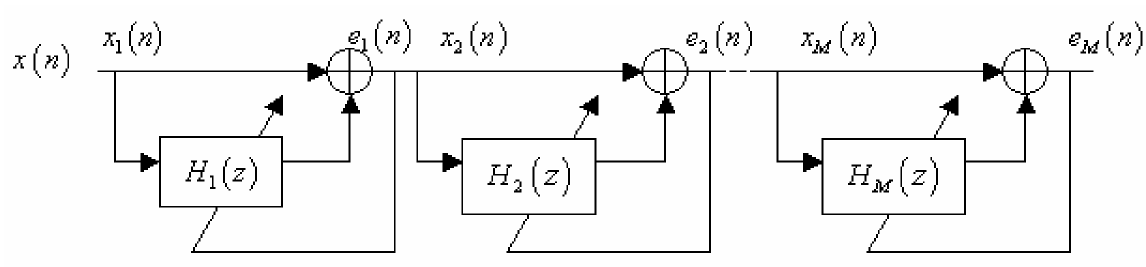


Fig. 4.1: Structure of cascaded predictor

The general structure of the cascade for the linear prediction can be shown in Fig. 4.1. In the cascade structure, each stage of the *M* sections uses an independently adapting FIR predictor of order l_k , $k=1,...,M$. Let $x_k(n)$ and $e_k(n)$ be the input and corresponding prediction error sample of stage k, respectively, with the latter being given by

$$e_k(n)=x_k(n)-\sum_{m=1}^{l_k}h_k^{(m)}(n)x_k(n-m) \tag{4.6}$$

where $h_k^{(1)}(n),...,h_k^{(l_k)}(n)$ are the time-varying taps of the k th predictor. Each stage of the cascade structure satisfies $x_{k+1}(n)=e_k(n)$; $x_1=x(n)$, where $x(n)$ is the input audio signal. The error of the last stage, $e_M(n)$ is the final prediction error of the cascade structure. After convergence, $h_k^{(m)}(n) = h_k^{(m)}$, where $h_k^{(m)}$ is a constant. The overall transfer function of the cascaded predictor can be expressed as

$$\hat{H}(z) = \prod_{k=1}^{M} H_k(z) \tag{4.7}$$

where

$$H_k(z) = 1 - \sum_{m=1}^{l_k} h_k^{(m)} z^{-m} \tag{4.8}$$

In fact, the FIR predictor in such a cascade structure is inadequate for general input signals since the resulting prediction filter has only strictly real zeros. Craven et al and C. Montgomery have shown that an Infinite Impulse Response (IIR) predictor may potentially perform better [29] [30]. However, IIR predictors have not been widely used in lossless audio coding because the general solution for the MSE predictor is much more complicated in this case. Recently, there is a report that the speed of convergence of the cascaded FIR filter with LMS adaptation is faster and its initial MSE is usually smaller than those of equivalent-order LMS and lattice LMS predictors [28]. For each stage, the cost function is defined as

$$J_k = E\left[e_k^2(n)\right] = E\left[x_{k+1}^2(n)\right] \tag{4.9}$$

In the experiments, the most successful experiments employed long filters in the middle stage and low-order filters in the preceding and subsequent stages. We will discuss it in follows.

4.3 Characterization of a Cascaded Linear Predictor

4.3.1 The Performance of LMS Predictor with Independence Assumption

Before developing a theoretical characterization of the cascade structure, we need to review the MSE performance of the LMS predictor. In the cascade structure, each stage performs prediction by passing past values through an l_k-tap FIR filter, where the filter weights are updated through the LMS weight update equation

$$\mathbf{h}_k(n+1) = \mathbf{h}_k(n) + \mu e_k(n)\mathbf{x}_{l_k}(n) \tag{4.10}$$

where $\mathbf{x}_{l_k}(n) = \left[x(n-1), x(n-2), \cdots, x(n-l_k)\right]^T$, and $\mathbf{h}_k(n) = \left[h_k^{(1)}(n), h_k^{(2)}(n), \cdots, h_k^{(l_i)}(n)\right]^T$.

The weight update equation is derived through a minimization of the mean-square error (MSE) between the desired signal and the LMS estimate, namely,

$$E\left[e_k^2(n)\right] = E\left[\left(x_k(n) - \hat{x}_k(n)\right)^2\right] \tag{4.11}$$

For simplicity, the performance of the LMS predictor can be analyzed with the independence assumption [27] which is described as follows,

1. the composite (desired signal and input vector) vectors $\left[\hat{x}_i(n)\mathbf{x}_{l_i}^T(n)\right]^T$, and $\left[\left[\hat{x}_i(n)\mathbf{x}_{l_i}^T(n)\right]^T, \cdots, \left[\hat{x}_i(-\infty)\mathbf{x}_{l_i}^T(-\infty)\right]^T\right]^T$ are independent of each other;
2. $\hat{x}_i(n)$ is dependent on $\mathbf{x}_{l_i}^T(n)$;
3. $\mathbf{x}_{l_i}^T(n)$ and $\hat{x}_i(n)$ are mutually Gaussian.

The performance of the LMS predictor can be bounded by that of the finite Wiener filter, where the filter weights are given in terms of the autocorrelation matrix of the reference signal $\mathbf{R}_k$, and the cross-correlation vector between the past value and desired signals $\mathbf{r}$. Explicitly, the weights are

$$\mathbf{h}_k(n) = \mathbf{R}_k^{-1}\mathbf{r}_k \tag{4.12}$$

where $\mathbf{R}_k = E\left[\mathbf{X}_{l_k}(n)\mathbf{X}_{l_k}^H(n)\right]$, $\mathbf{r}_k = E\left[\mathbf{X}_{l_k}(n)x_{l_k}(n)\right]$ and H denotes the conjugate transpose operator.

The MSE of the LMS predictor under these assumptions is therefore bounded by the MSE of the finite Wiener filter, which is

$$E\left[e_k^2(n)\right] = E\left[\left(x_k(n) - \hat{x}_k(n)\right)^2\right] = E\left[x_k^2(n)\right] - \left(\mathbf{R}_k^{-1}\mathbf{r}_k\right)^T \mathbf{r}_k \tag{4.13}$$

Referring to Equation (4.9), we are able to write the above equation in terms of the power spectral density function of $\{h_k(n)\}$ as

$$J_{k_{opt}} = \int_{-\pi}^{\pi} S_{x_k x_k}(\lambda)\, d\lambda - \int_{-\pi}^{\pi} \left|H_k(\lambda)\right|^2 S_{x_k x_k}(\lambda)\, d\lambda \tag{4.14}$$

Obviously, with the independence assumption the performance bound of the LMS predictor is the infinite Wiener filter. Actually, we will discuss it further without this assumption and obtain a lower performance bound in section 4.4.

4.3.2 Characterization of the Cascade Structure

In this section, we try to prove that the cascaded adaptive FIR filter operates as a linear prediction in terms of successive refinements. The cascaded adaptive FIR operation can be described in the following theorem.

Theorem 1:

In the cascaded FIR filter structure, each stage attempts to cancel the dominant mode of its input signal, i.e. to place its zeros close to the dominant poles of the Autoregressive (AR) model. It performs linear prediction with a progressive refinement strategy, i.e.

$$J_M\left(h_M\right) \le J_{M-1}\left(h_{M-1}\right) \le \cdots \le J_1\left(h_1\right) \tag{4.15}$$

Proof: Assuming N to be the minimum description length (MDL) of the AR model, the time series $x(n), x(n-1), \cdots, x(n-N)$ can be realized by an AR model of order N as it satisfies the difference equation

$$x(n) + a_1^* x(n-1) + \cdots + a_N^* x(n-N) = v(n) \tag{4.16}$$

where $a_1, \cdots a_N$ are complex-valued constants, * denotes the conjugate operator and $v(n)$ is white noise. The corresponding system generates $x(n)$ with $v(n)$ as input, whose transfer function is

$$H(z) = \frac{1}{\sum_{i=0}^{N} a_i^* z^{-i}} \tag{4.17}$$

This function is completely defined by specifying the location of its poles, as shown by

$$H(z)=\frac{1}{\left(1-p_1z^{-1}\right)\left(1-p_2z^{-1}\right)\cdots\left(1-p_Nz^{-1}\right)} \tag{4.18}$$

The parameters $p_1, p_2, \cdots, p_N$ are the poles of $H(z)$; they are defined by the roots of the characteristic equation

$$1+a_1^*z^{-1}+\cdots+a_N^*z^{-N}=0 \tag{4.19}$$

For the system to be stable, the roots of the characteristic Equation (4.19) must all lie inside the unit circle in the z-plane, e.g., $|p_k|<1$, for all $k=1,\cdots,N$. The FIR filter of order N in each stage contributes to estimate weights $w_1^*,\cdots,w_N^*$ in the linear prediction problem

$$x(n)=w_1^*x(n-1)+\cdots+w_N^*x(n-N)+v(n) \tag{4.20}$$

such that $w_k=-a_k$. The analyzer function $H(z)$ can be expressed in cascade form

$$\begin{aligned}H(z)&=1-w_0^*z^{-1}-w_1^*z^{-2}-\cdots-w_N^*z^{-N}\\&=\left(1-\sum_{m=1}^{l_1}h_1^{(m)}z^{-m}\right)\left(1-\sum_{m=1}^{l_2}h_2^{(m)}z^{-m}\right)\cdots=\prod_{k=1}^{M}H_k(z)\end{aligned} \tag{4.21}$$

where $\sum_{k=1}^{M}l_k=N$. We have the output $e_1(n)$ of the first stage of the cascaded FIR structure when the LMS predictor converges to its steady-state value,

$$
\begin{aligned}
e_1(n) &= \sum_{m=1}^{M} w_m^* x(n-m) + v(n) - \sum_{m=1}^{l_1} h_1^{(m)}(n) x(n-m) \\
&= \underbrace{\sum_{m=1}^{l_1} \left(w_m^* - h_1^{(m)} \right)(n-m)}_{e_0(n)} + \sum_{m=l_1}^{M} w_m^* x(n-m) + v(n)
\end{aligned}
\tag{4.22}
$$

The cost function at the first stage becomes

$$
\begin{aligned}
J_1(n) &= E\left[\left| e_1(n) \right|^2 \right] \\
&= E\left[\left| \begin{aligned} & e_0^2(n) + v^2(n) + 2e_0(n)v(n) \\ & +2\sum_{m=l_1}^{N} w_m^* x(n-m)\left(e_0(n) + v(n)\right) \\ & + \left(\sum_{m=l_1}^{N} w_m^* x(n-m) \right)^2 \end{aligned} \right| \right]
\end{aligned}
\tag{4.23}
$$

According to the principle of orthogonality, in the steady-state, $E\left[e_0(n) v(n) \right] = 0$ and $E\left[x(n-m)\left(e_0(n) + v(n)\right) \right] = 0$. The cost function becomes

$$
J_1(n) = E\left[\left| e_0(n) \right|^2 + \sigma_v^2(n) + \left| \sum_{m=l_1}^{M} w_m^* x(n-m) \right|^2 \right] \tag{4.24}
$$

where $\sigma_v^2(n)$ is the variance of the white noise $v(n)$. We see that $J_1(n)$ achieves its minimum, if and only if, the following two terms are minimal

$$J_{1_{e_0}}(n) = E\left[\left|e_0(n)\right|^2\right] \tag{4.25}$$

and

$$J_{1_w}(n) = E\left[\left|\sum_{m=l_1}^{M} w_m^* x(n-m)\right|^2\right] \tag{4.26}$$

It means that the first stage attempts to cancel the dominant mode of its input signal, i.e. to place its zeros close to the dominant poles of the AR model.

Let us look at the sufficient condition. If $J_{1_{e_0}}(n)$ and $J_{1_w}(n)$ are minimal, the dominant component of the input signal is removed. In fact, $H_1(z)$ can be decomposed as

$$\hat{H}_1(z) = \left(1-\hat{p}_1 z^{-1}\right)\left(1-\hat{p}_2 z^{-1}\right)\cdots\left(1-\hat{p}_{l_1} z^{-1}\right) \tag{4.27}$$

The zeros $\left|\hat{p}_k\right| < 1, k = 1,\cdots,l_1$ are close to the poles $p_k, k = 1,\cdots,l_1$ in Equation (4.18), which dominates the main component of the input. The remaining poles $p_k, k = l_1 + 1,\cdots,N$ contributes to the minor components of the input, resulting in the minimum $J_{1_w}(n)$.

For necessary condition, only if, we can assume that the zeros $\left|\hat{p}_k\right| < 1, k = 1,\cdots,l_1$ are close to the poles $p_k, k = 1,\cdots,l_1$ in Equation (4.18), which are not the dominant component of the input. There are poles among the $p_k, k = 1, \cdots, M$, which give the dominant component of the input. Therefore, there is a subset of $\hat{w}_k$ such that

$$\left| \sum_{m=l_1}^{N} \hat{w}_m^* x(n-m) \right|_i^2 > \left| \sum_{m=l_1}^{N} w_m^* x(n-m) \right|_i^2 \tag{4.28}$$

The $J_{1_w}(n)$ is not minimum. This is contradictory to the initial assumption, i.e. the cost function $J_1(n)$ achieves its minimum, resulting in the minimum $J_{1_w}(n)$. Thus the first stage will attempt to cancel the dominant mode of its input signal, i.e. to place its zeros close to the dominant poles of the AR model. The proof for the second stage is done in same way, and so on.

Referring to Equation (4.9) and Equation (4.14), it is easy to verify that

$$\begin{aligned} J_k(h_k) &= E\left[e_{k-1}^2(n)\right] - \underbrace{\int \left|H_k(\lambda)\right|^2 S_{x_k x_k}(\lambda)\, d\lambda}_{a^2} \\ &= J_{k-1}(h_{k-1}) - a^2 \le J_{k-1}(h_{k-1}) \end{aligned} \tag{4.29}$$

where $a^2 < E\left[e_{k-1}^2(n)\right]$. Therefore, $J_M(h_M) < J_{M-1}(h_{M-1}) < \cdots < J_1(h_1)$, the theorem is proved.

■

With Theorem 1, we can derive the following property of the cascaded LMS predictor.

Lemma 1:

If each stage of the cascade LMS predictor converges to its steady-state value, the cascaded FIR filter structure possesses the following property:

$$\chi\left(R_M\right) < \chi\left(R_{M-1}\right) < \cdots < \chi\left(R_1\right) \tag{4.30}$$

where the eigenvalue spread $\chi\left(R_k\right) = \dfrac{\lambda_{k_{\max}}}{\lambda_{k_{\min}}}$.

Proof: The condition given in Lemma 1 means that the optimum cascaded FIR filter structure satisfies Theorem 1. The output of the first stage $e_1\left(n\right)$ can be characterized by a $\left(N - l_1\right)$ -tap input vector. In other words, after the first stage's adaptation and convergence, the input signal's dynamic range to the second stage is reduced. The ratio between the peak and average of the power spectral density of the input signal is decreased. Thus

$$\chi\left(R_2\right) < \chi\left(R_1\right) \tag{4.31}$$

For the output of the second stage $e_2(n)$, by the same reason, it can be estimated using a $(N-l_1-l_2)$-tap input vector. Therefore, the eigenvalue spread of the input to the third stage satisfies

$$\chi(R_3)<\chi(R_2) \tag{4.32}$$

and so on until the last stage, the input can be estimated using a l_k-tap input vector and satisfies

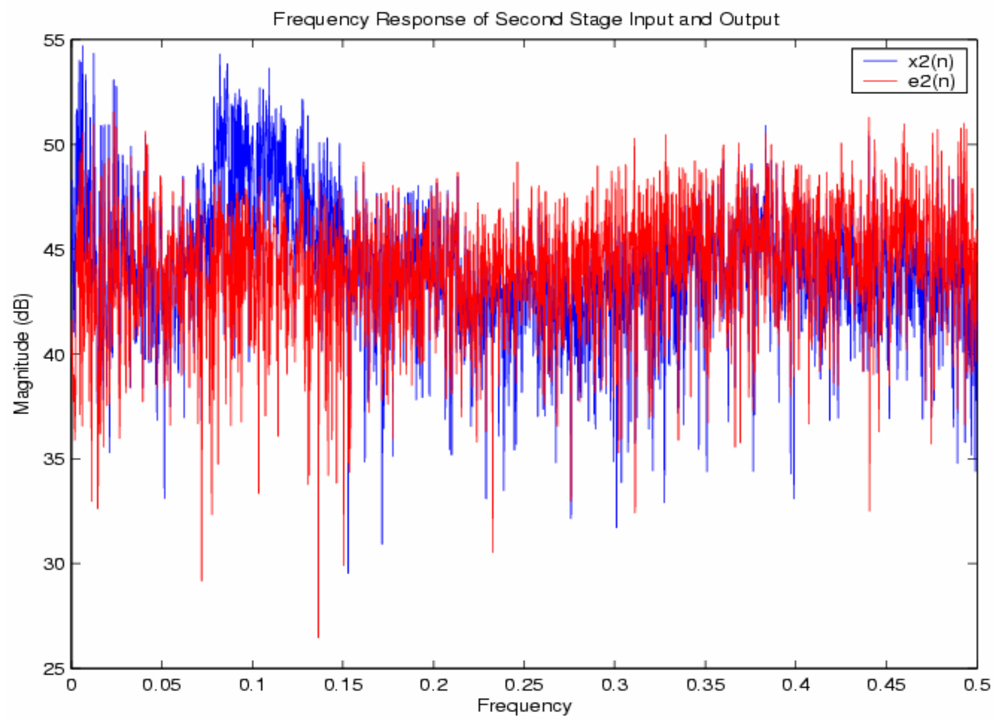
$$\chi(R_M)<\chi(R_{M-1}) \tag{4.33}$$

The lemma is proved.

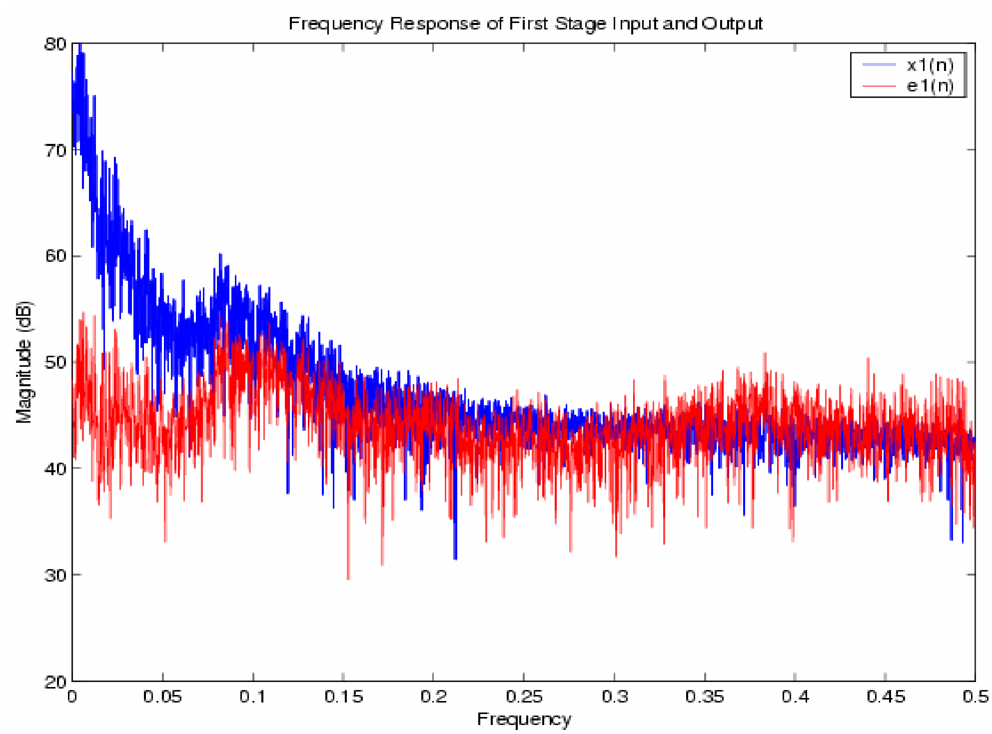
■

4.3.3 Simulation Results

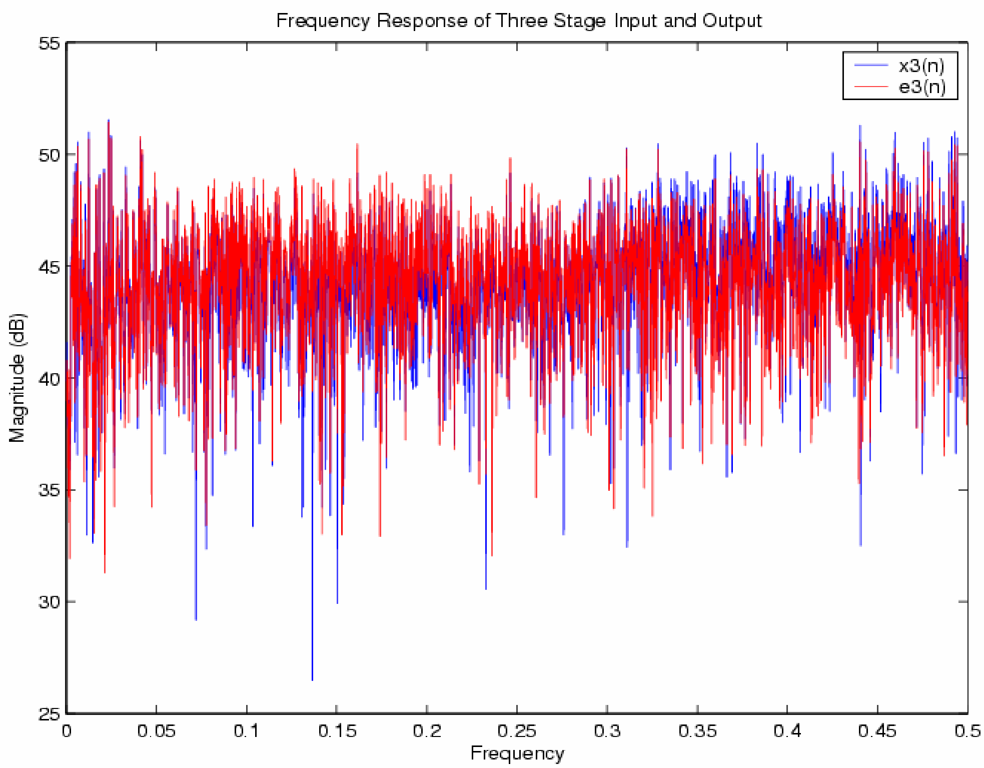
In the demonstration of the theorem and lemma above, we assume that each stage of the cascaded LMS predictor converges to its steady-state value. However, LMS convergence speed suffers from both the length of the filter and the eigenvalue spread of the input covariance matrix. In this simulation, the first stage uses a low-order filter as a pre-whitening adaptive filter to reduce the eigenvalue spread. The second stage adopts a long LMS predictor, which works well for general signals and it is different from the cascaded low-order filters [28].


Frequency Response of Second Stage Input and Output
x2(n)
e2(n)
Magnitude (dB)
Frequency
55
50
45
40
35
30
25
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

(a)


Frequency Response of First Stage Input and Output
x1(n)
e1(n)
Magnitude (dB)
Frequency
80
70
60
50
40
30
20
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

(b)



(c)

Fig. 4.2: Frequency response of a 3-stage cascaded LMS predictor: (a) First stage $x1(n)$ and $e1(n)$; (b) Second stage $x2(n)$ and $e2(n)$; (c) Third stage $x3(n)$ and $e3(n)$

We carried out the simulation for real audio signals to evaluate the behavior of the cascaded LMS predictor. To support the theoretical analysis, the prediction results of the 3-stage cascaded LMS predictor are shown in Fig.4.2, from which it can be observed that the cascaded LMS predictor removes the dominant component of input by successive refinements at each stage.

We gave a formal proof that the cascaded adaptive linear predictor performs linear prediction in terms of successive refinement, under the assumption that each stage of the cascade predictor converges to its steady state. Because of the successive refinement, i.e.

the lower MSE and faster convergence stage by stage, the adaptive linear predictor may lead to better predictive gain than LPC technique [31]. We will study the performance bound of this structure in the next section.

4.4 A Performance Bound for a Cascaded Linear Predictor

4.4.1 Performance Bound

As discussed in section 4.3.1, the performance of the LMS predictor can be bounded by that of the finite Wiener filter with the independence assumption [27] for simplicity. In this section we will discuss it further without the independence assumption.

Assuming that the initial weight vector at time $n = -\infty$ is the all-zero vector, we can write the LMS predictor as a nonlinear function $f(\ )$ of semi-infinite set of input signal, as well as of past values of the desired signal,

$$\begin{aligned} \hat{x}_k(n) &= \mu \sum_{i=-\infty}^{n-1} e(i)\, x_{l_k}^T(n)\, x_{l_k}(n) \\ &= f\left(x_k(n), x_k(n-1), \cdots, \hat{x}_k(n-1), \cdots\right) \end{aligned} \tag{4.34}$$

The performance of any predictor can be bounded by that of the optimal predictor. The optimal MSE predictor is given by the mean of the desired signal, conditioned on the knowledge of all information available to the predictor. Examining Equation (4.34) for the LMS predictor, the optimal predictor is given by

$$\hat{x}_k(n) = E\left\{x_k(n), x_k(n-1), \cdots, \hat{x}_k(n-1), \cdots, \hat{x}_k(-\infty)\right\} \tag{4.35}$$

Actually, to solve the problem it is required to know the statistics of the signals $\hat{x}_k(n)$ and $x_k(n)$. In the analysis, the third assumption is kept. The mutually Gaussian assumption is similar to that for the optimal LMS estimator in [32]. Including the entirety of both processes, the optimal predictor of k th stage is given by

$$\hat{x}_k(n) = \sum_{i=-\infty}^{n-1} h_x(n-i)\, x_k(i) + \sum_{i=-\infty}^{n-1} h_{\hat{x}}(n-i)\, \hat{x}_k(i) \tag{4.36}$$

where the impulse responses of the causal linear predictor for $x_k(n)$ and causal linear predictor for $\hat{x}_k(n)$ are given as $h_x(n)$ and $h_{\hat{x}}(n)$, respectively. The MSE of the optimal predictor, and thus a bound on the LMS predictor's performance is

$$E\left[e_{k_{opt}}^2\right] = E\left[\left|x_k(n) - \hat{x}_k(n)\right|^2\right] \tag{4.37}$$

As the same analysis process in [32], we are able to write above equation in terms of the spectral density functions as

$$E\left[e_{k_{opt}}^{2}\right]=\int_{-\pi}^{\pi}S_{x_k x_k}\left(\lambda\right)d\lambda-\int_{-\pi}^{\pi}\left|H_{\hat{x}_k}\left(\lambda\right)\right|^{2}S_{\hat{x}_k\hat{x}_k}\left(\lambda\right)d\lambda$$
$$-2R\left(\int_{-\pi}^{\pi}e^{-k\lambda}H_{\hat{x}_k}\left(\lambda\right)H_{x_k}^{*}\left(\lambda\right)S_{\hat{x}_k x_k}\left(\lambda\right)d\lambda\right) \tag{4.38}$$
$$-\int_{-\pi}^{\pi}\left|H_{x_k}\left(\lambda\right)\right|^{2}S_{x_k x_k}\left(\lambda\right)d\lambda$$

Equation (4.38) is a bound on the performance of the LMS predictor under a mild set of assumptions which does not exclude data contributions available to the predictor. Therefore, the performance bound of a cascaded LMS predictor can be described in the following theorem.

Theorem 2:

In an M-stage cascaded LMS predictor, each stage attempts to cancel the dominant mode of its input signal in a successive refinement strategy. The performance of such predictor is bounded by,

$$J_{opt}\le J_{M}=\int_{-\pi}^{\pi}S_{x_M x_M}\left(\lambda\right)d\lambda-\int_{-\pi}^{\pi}\left|H_{\hat{x}_M}\left(\lambda\right)\right|^{2}S_{\hat{x}_M\hat{x}_M}\left(\lambda\right)d\lambda$$
$$-2R\left(\int_{-\pi}^{\pi}e^{-M\lambda}H_{\hat{x}_M}\left(\lambda\right)H_{x_M}^{*}\left(\lambda\right)S_{\hat{x}_M x_M}\left(\lambda\right)d\lambda\right) \tag{4.39}$$
$$-\int_{-\pi}^{\pi}\left|H_{x_M}\left(\lambda\right)\right|^{2}S_{x_M x_M}\left(\lambda\right)d\lambda$$

4.4.2 Simulation Results

The performance of the cascaded LMS predictor has been bounded with Equation (4.39) for the class of signals satisfying assumptions in [32]. In this section, we desire to demonstrate this bound for the example where the LMS as well as the cascaded LMS predictor outperform the finite Wiener filter. To do this, we use stable Autoregressive Moving Average (ARMA) processes generated by the following system whose poles are restricted to be close to the unit circle meeting assumptions in [32],

$$\text{poles} \begin{cases} -0.3 \\ 0.75 \pm j0.5809, \\ -0.5 \pm j0.8307 \end{cases} \qquad \text{zeros} \begin{cases} 1.05 \\ -0.2 \\ 0.9 \pm j0.4472 \end{cases} \tag{4.40}$$

when driven by white noise, normalized to yield $r_0 = 1$.

The MSE performance of the LMS predictor was evaluated through Monte Carlo simulations. The performance of the finite Wiener filter using Levinson-Durbin algorithm was evaluated numerically.

Fig. 4.3 shows the learning curve, the average MSE of LMS predictor and the MSE of the finite Wiener filter using Levinson-Durbin algorithm. In this case, the average MSE of LMS predictor and the MSE of Wiener predictor are 5.2892dB and 6.4265dB, respectively. The LMS predictor performs better than Wiener predictor.

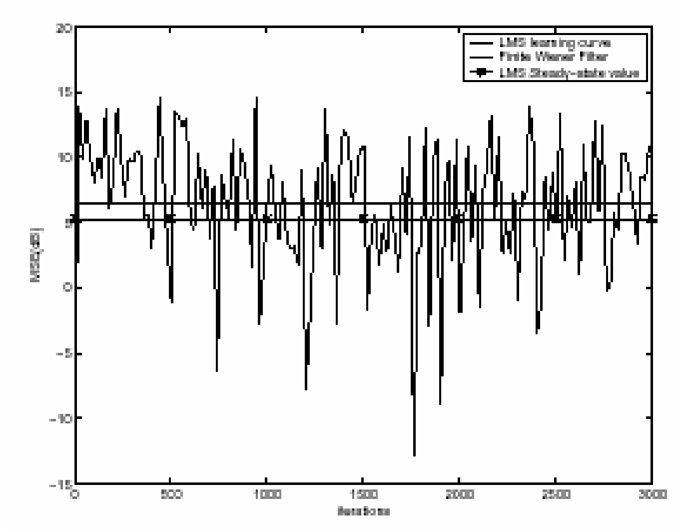


Fig. 4.3: MSE of the LMS predictor and the LPC based predictor

Fig. 4.4 shows the learning curves of an LMS predictor and a three-stage cascaded LMS predictor for the above ARMA process. We observe that the cascaded LMS predictor behaves faster convergence than the single stage LMS predictor. It means that the cascade FIR structure using LMS algorithm leads to better prediction than the traditional LPC technique.

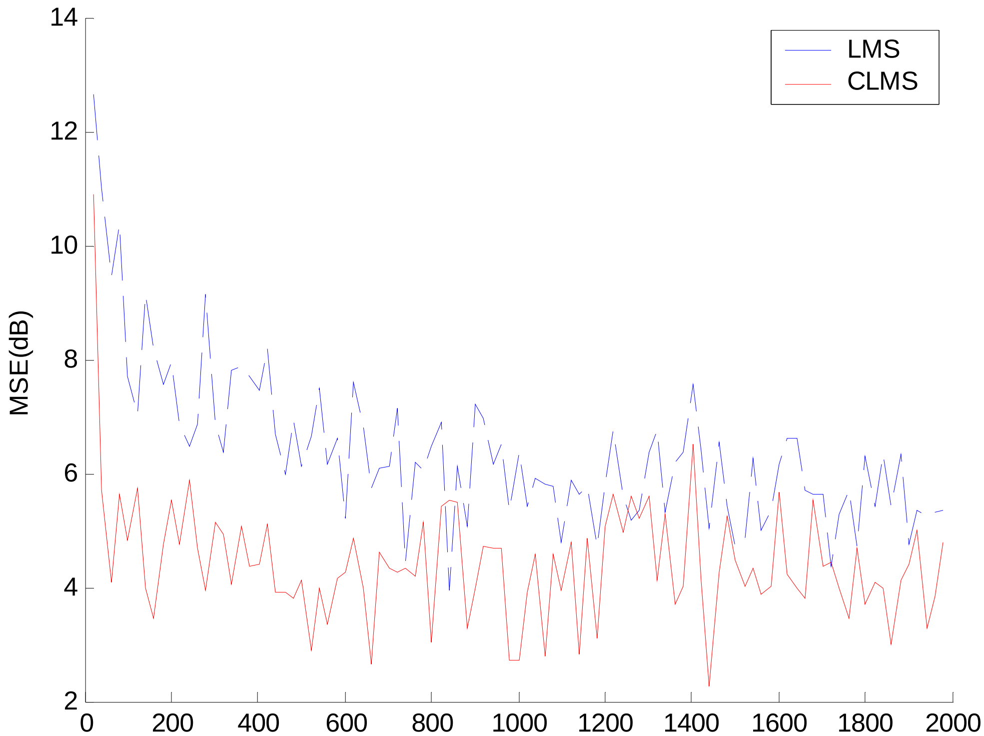


Fig. 4.4: The learning curves of the LMS predictor and the cascaded LMS predictor

Fig. 4.5 shows the learning curves of each stage in a three-stage cascaded LMS predictor. The three stages' average MSE are 5.39 dB, 4.30 dB, and 4.20 dB, respectively, calculated after 400 iterations. The results confirm Theorem 1, while each stage converges to its optimal value, and the cascaded LMS predictor behaves in a successive refinement, i.e.

$$J_3 < J_2 < J_1 \tag{4.41}$$

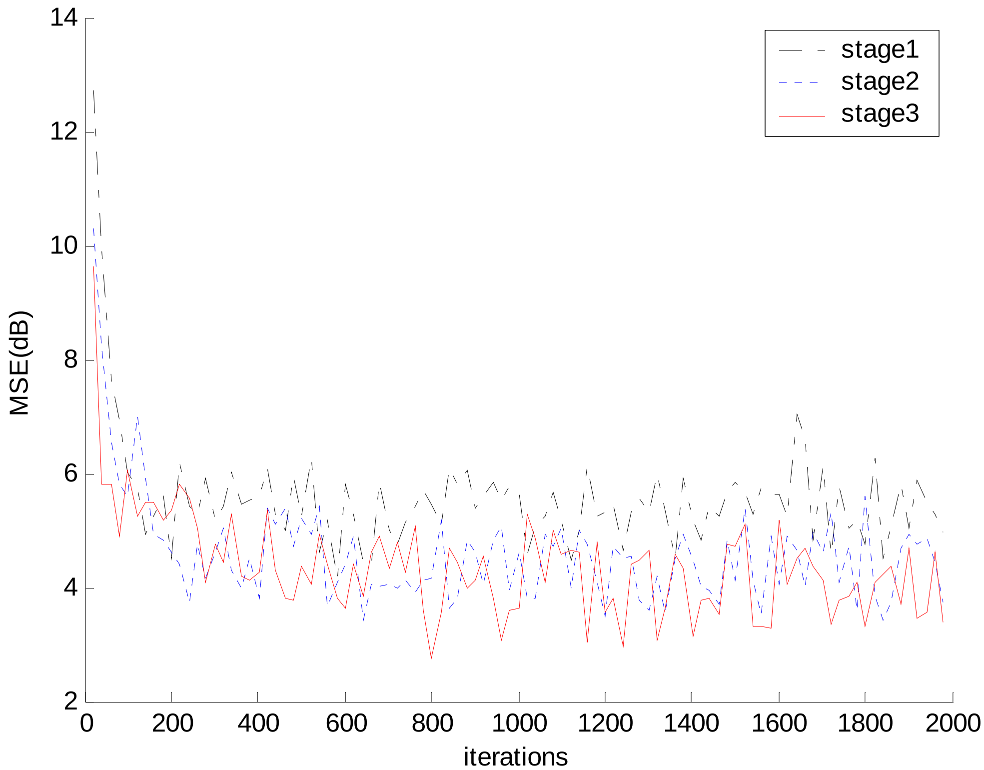


Fig. 4.5: The leaning curves of each stage in three-stage cascaded LMS predictor

We bounded the performance of the cascaded LMS predictor without using the independence assumption where the cascaded LMS predictor outperforms the finite Wiener predictor, which has been known as a bound of the LMS predictor under independence assumption. The conclusion is that the performance of the cascaded LMS predictor can be better than the traditional LPC technique for synthetic or real audio signals [33].

4.4.3 Challenge

In section 4.3 and 4.4, we analyzed the characteristics and performance bound of the cascaded adaptive linear predictor in theory. It is pointed out that this cascaded predictor can perform better than LPC technique. In [28], Prandoni and Vetterli have proposed a second-order cascaded predictor, which gives a good performance for speech signal prediction. However, can this low-order cascade structure still work well for audio signal prediction? Although we proved the advantage of cascade structure in theory, we face the challenging task in practical application. For example, for audio signal prediction, what kind of the cascade structure should be to surpass the LPC technique? We will study this in following sections.

4.5 An Adaptive Cascade Structure for Audio Signals Modeling

4.5.1 Signal Models

The prediction efficiency of an adaptive linear predictor requires fast convergence as well as low MSE. In order to design a structure suitable for modeling all signals, we investigate the transient behaviour and the MSE performance of the cascaded LMS predictor for three classes of signals through simulations. The three models we chose are described as follows,

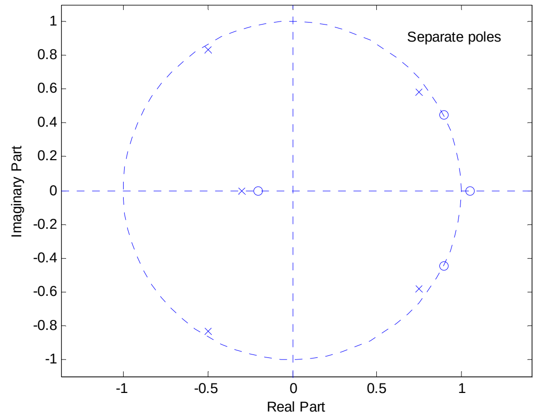
(a) Signals with well separate poles;

(b) Signals with frequency spectra symmetric around $\frac{\pi}{2}$;

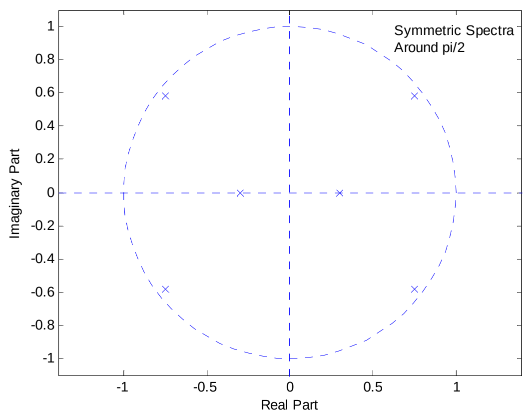
(c) Signals with clustered poles.

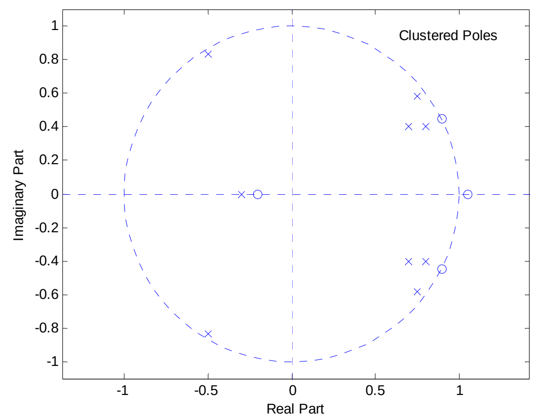
It is well known that most signals can be modeled by model (a), i.e. with well separate poles. Therefore, the basic requirement is that the good predictor should give a superior performance for such signals. However, the challenging task is to get the uniform cascade structure to model any audio signal efficiently. We find that even if the structure can work perfectly for signals with well separate poles, it is possible that such structure would still fail in general audio signal modeling. That is because the models for audio signals, normally composed of abundant poles and zeros, are always complex and diversified. Therefore, to get the universal structure for audio signal modeling, other typical signal models must be considered as well, such as model (b) and (c). The optimal structure should work well for all classes of signals in average sense. We will discuss it in details in following sections.

In simulation, the zero-pole positions for different models are shown in Fig. 4.6 respectively. We use an ARMA model (with 5 poles and 4 zeros) for signals with well separate poles, an AR model (with 6 poles and without zero) for signals with frequency spectra symmetric around $\pi/2$ (whose autocorrelation is zero at odd lags), and an ARMA model (with 9 poles and 4 zeros) for signals with clustered poles. The input simulation signals are obtained by filtering unit variance Gaussian white noise through the ARMA or AR modeling filter.



(a)





(b) (c)

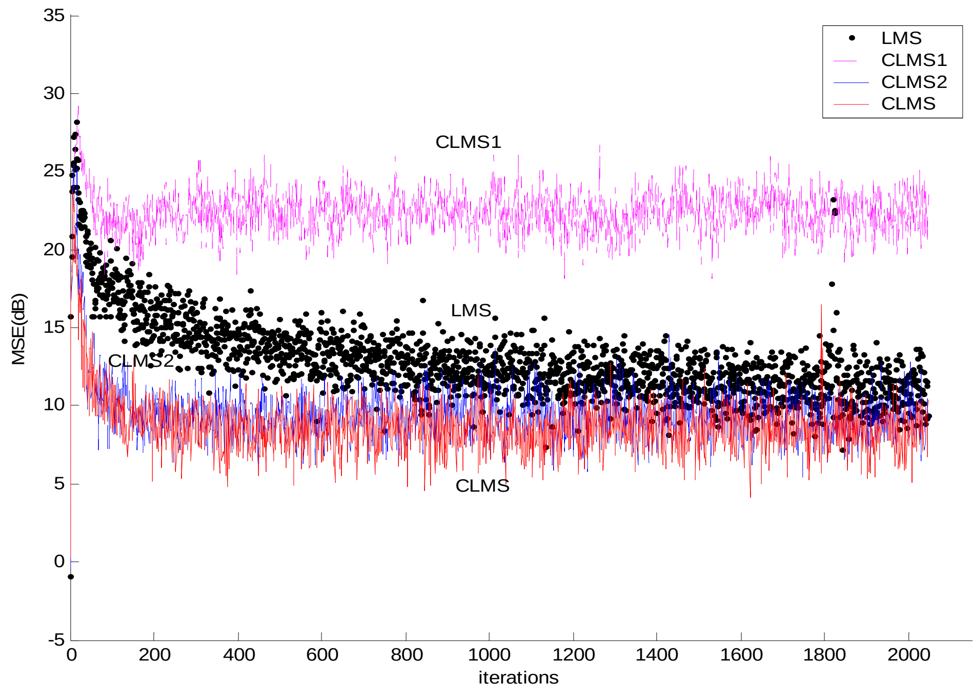
Fig. 4.6: Zero-pole position diagram: (a) ARMA (5 poles and 4 zeros); (b) AR (6 poles); (c) ARMA (9 poles and 4 zeros)

4.5.2 A Cascade Structure for Signals Modelling

In the experimentation, we compare the performance of standard LMS predictor, one-tap, two-tap cascaded LMS predictors [28] and the proposed cascaded LMS predictor, in which low-order LMS predictors are preceding and subsequent to a long LMS predictor.

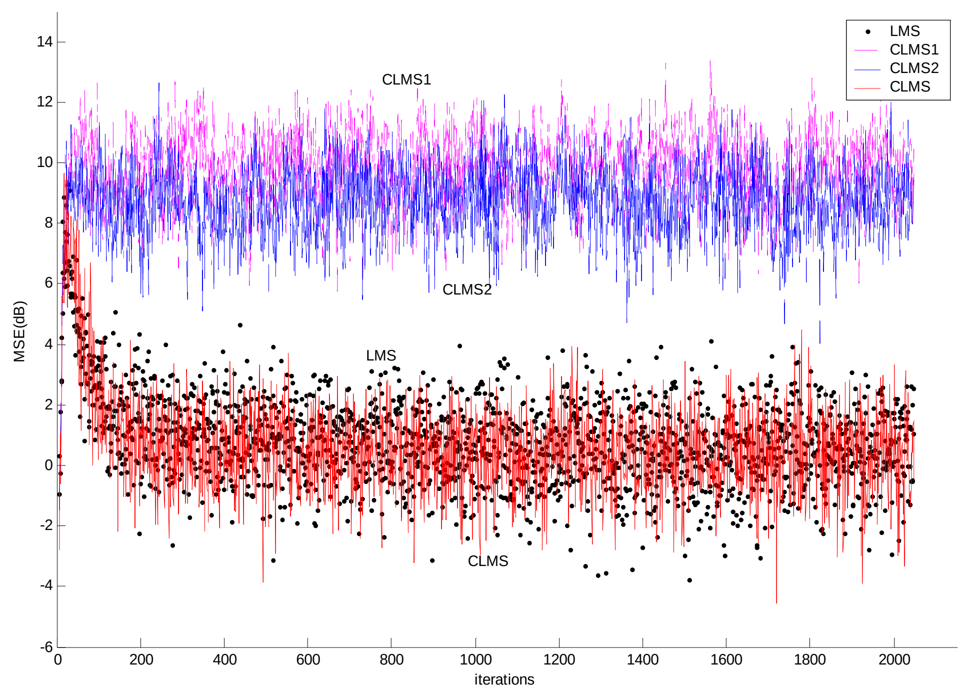
The standard LMS filter, i.e. one stage LMS filter will be labelled as ”LMS”. The one-tap cascaded LMS filter which contains N stages, will be labelled as ”CLMS1”. And a $N/2$ stages second-tap cascaded LMS filter will be labelled as ”CLMS2”. For simplicity, the length of the standard LMS filter is selected as 8 taps, i.e. $N=8$, for simulation. In such case, the proposed cascaded LMS predictor can be designed as 2-tap filter for first stage, 4-tap filter for second stage and 2-tap filter for third stage, which is labelled as ”CLMS”. Now these four predictors have the equivalent order length.

The results for three classes of signals are shown in Fig. 4.7(a), (b) and (c) respectively. In Fig. 4.7(a), it can be seen that the best to the worst performance order is CLMS, CLMS2, LMS, and CLMS1, which confirms our analysis. In fact, for signals with well separate poles, 2-tap cascaded LMS filter is enough to successfully model it. However the proposed cascade structure can get better performance than two-tap cascade structure. Because most signals are in such model, both of the proposed structure and the two-tap cascade structure can be the candidates. However, the one stage LMS is not the good choice because it fails in modelling such signals well.



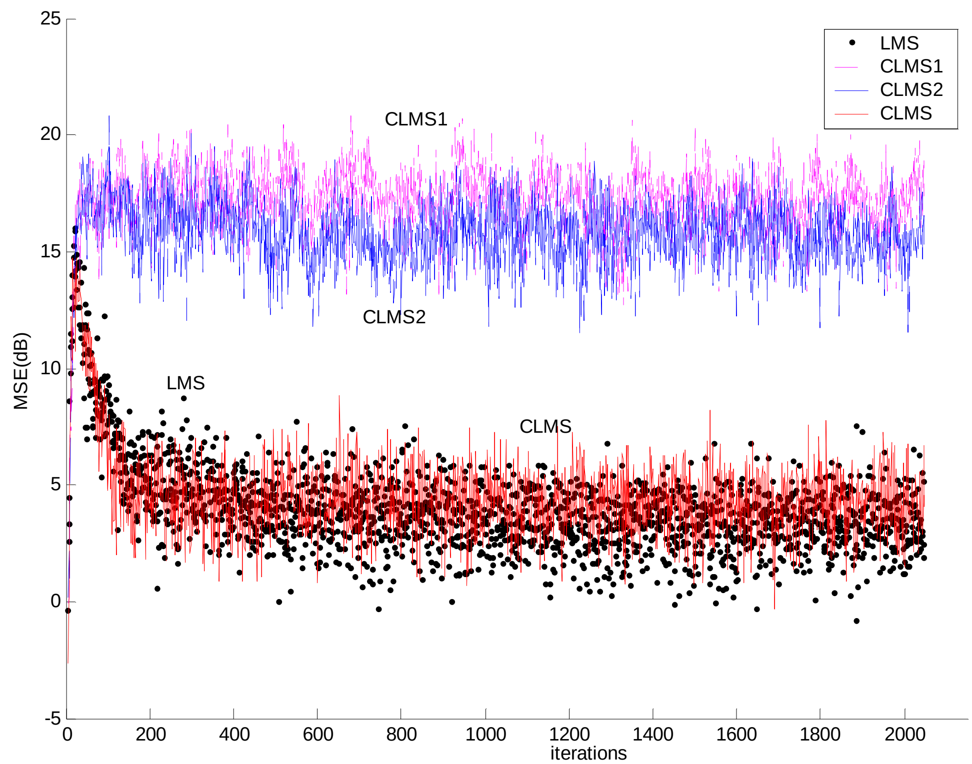
LMS
CLMS1
CLMS2
CLMS
CLMS1
LMS
CLMS2
CLMS
MSE(dB)
iterations
35
30
25
20
15
10
5
0
-5
0
200
400
600
800
1000
1200
1400
1600
1800
2000

(a)



LMS
CLMS1
CLMS2
CLMS
CLMS1
CLMS2
LMS
CLMS
MSE(dB)
iterations
14
12
10
8
6
4
2
0
-2
-4
-6
0
200
400
600
800
1000
1200
1400
1600
1800
2000

(b)



(c)

Fig. 4.7: MSE performance comparison between LMS (dotted line), one-tap cascade LMS (dash-dot), two-tap cascade LMS (dashed) and variant length cascade LMS (solid) in predicting signal in (a) model a; (b) model b; (c) model c

From Fig. 4.7(b) and (c) it can be seen that the proposed structure and the one-stage filter behave the best and get the similar performance. It is also seen that the one-tap cascaded LMS predictor cannot get good performance as expected, and the two-tap cascaded LMS predictor cannot behave better than the standard LMS predictor for these two cases. These simulation results are consistent with the theory. It is well known that the small autocorrelation matrix would not allow the low-order filter to cancel the separate modes of input signal which is generated by the signals with frequency spectra symmetric around $\pi/2$ or with clustered poles. It means each stage of the predictor cannot place its zeros close to the dominant poles of input signal. Therefore the dominant modes cannot be cancelled or weakened much as in predicting signals with well separate poles. As a result, the output error of each low-order stage is unchanged almost compared to the input signal. It is one of reasons for which the low-order cascade structure fails to work well in these two classes of signals. However, the proposed cascaded LMS predictor, with short filters and long filters, still gets the satisfied performance for these classes of signals.

Obviously, from the simulation results, the optimal cascaded filter is CLMS. It is well known that LMS predictor's convergence speed suffers from the big eigenvalue spread of the high correlated input signals. In the experiment, we have not discussed such structure in which a long filter is used as the first stage. Using a long filter in first stage is not recommended, because convergence performance of such structure is heavily affected by high correlated input. In the proposed structure, the first low-order stage can be used as a pre-whitening adaptive filter to reduce the eigenvalue spread. The second stage adopts a long LMS filter, which is required for most audio signals modeled with abundant poles and zeroes.

In this section, we studied the cascaded adaptive linear prediction for audio signal modelling [34]. The simulation results and analysis show that low-order cascade structure, which is good for speech coding, is not applicable for audio signal modelling. It comes to propose the optimal and uniform cascade structure as follows: the optimal cascade structure is composed of high-order filter in middle and low-order filters in previous and followed.

4.6 High Sampling Rate Audio Signal Modeling

4.6.1 Motivation

As mentioned before, the proposed adaptive linear predictor should work well for all of the audio signals with different sampling rates, i.e. not only 48 kHz but also 96 kHz and 192 kHz. If we got the predictor with an optimal cascade structure for audio signals with 48 kHz sampling rate, will the predictor still be applicable for corresponding audio signals with 96 kHz or 192 kHz sampling rate?

In section 4.5, we have already successfully designed and proposed an optimal and universal cascade structure for audio signal modelling. However, the optimal predictor cannot be applicable for signals which are generated by up sampling the corresponding original audio signals. It means that the optimal predictor for audio signals with 48 kHz sampling rate is always not optimal for the signals with 96 kHz or 192 kHz sampling rate. That is because the bandwidth of high sampling rate signals is larger than that of low sampling rate signals. Therefore the wider bandwidth, low pass filter is necessary to get better modelling for high sampling rate signals. Meanwhile, it also raises the challenging task. In this section, the purpose is to propose a uniform cascade structure for high sampling rate audio signal prediction with minimum description length FIR filters [27].

Similarly, we study such cascade structure for high sampling rate audio signals' modeling based on different zero-pole positions of signals such that the stability of the cascade LMS structure can be guaranteed. We will present the cascade LMS structure and the experimental results which are generated by applying this proposed LMS structure in modeling some typical signal models.

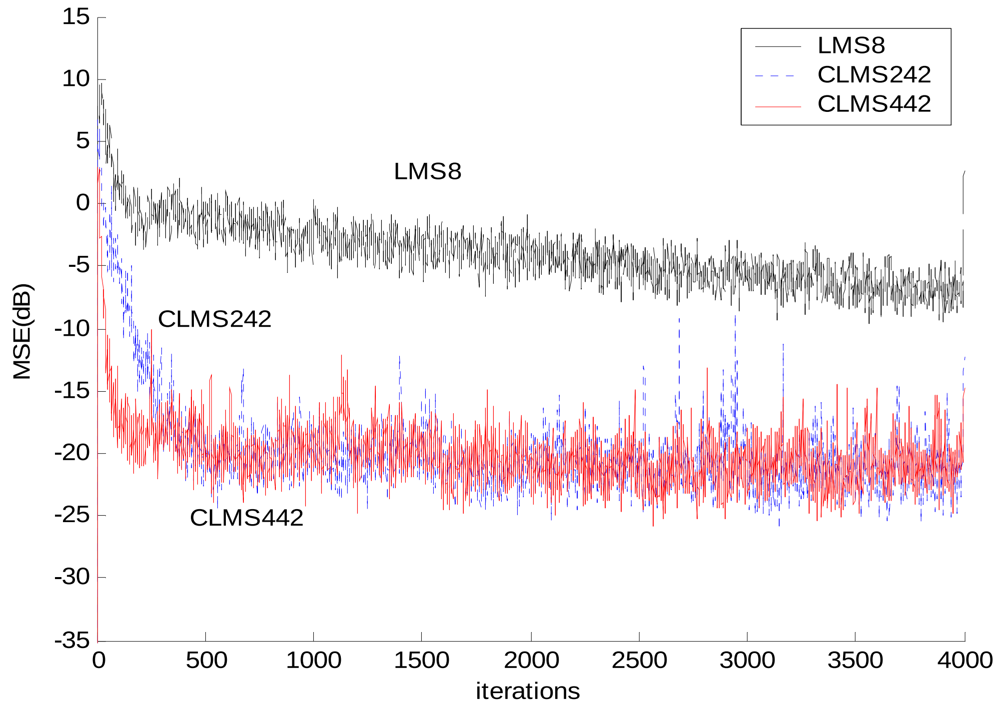
4.6.2 Study for High Sampling Rate Audio Signal Modeling

In order to get the high prediction gain, fast convergence and low MSE are required. In this part, we use three typical signal models to study the behaviours of the cascaded LMS predictor. The purpose is to design a suitable cascade structure for modeling most audio signals.

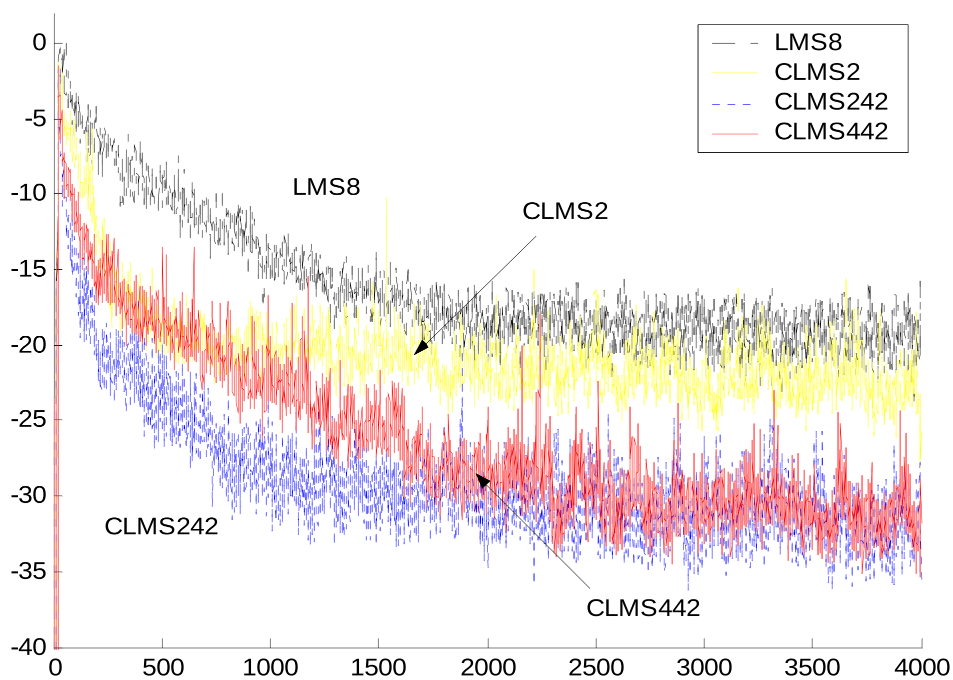
The three models are same with the models in section 4.5. As discussed, the optimal structure should perform the best for all classes of signals in average sense. In simulation, the zero-pole positions for different models are shown in Fig. 4.6 respectively. The input simulation signals are obtained by filtering unit variance Gaussian white noise through the ARMA or AR modeling filter.

With the same signal models as that in section 4.5 we have discussed and proposed a cascade structure for modeling audio signals with low sampling rate, i.e. 48 kHz. The proposed structure is composed of long filter in middle and low-order filters in previous and followed. Importantly, such structure outperforms other cascade LMS structure and LPC technique. However, the structure proposed in section 4.5 may not give good performance for high sampling rate because the designed low pass filter behaves as a relative narrow band filter. For the purpose of studying the modeling of high sampling rate signal, the corresponding signals are operated by up sampling.

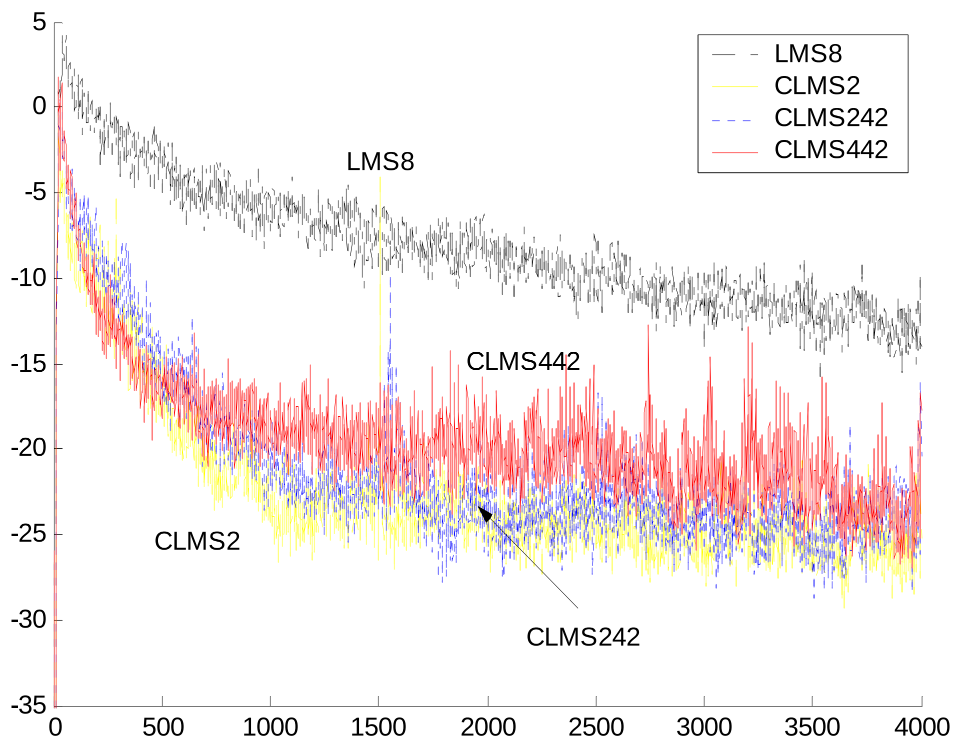
The simulation results are shown in Fig. 4.8 for the behaviours of different structures for above three typical signal models. These figures show the transient phase and the steady-state mean-square errors, which are averaged over an ensemble of 50 trials. We can observe the performance of different cascade structures for signal modeling. The standard LMS predictor with 8 taps is labelled as "LMS8". The learning curves for three different cascade structures are illustrated. One labelled as "CLMS242" is with 2-tap filter at first, 4-tap filter in middle, and 2-tap filter at last. Another is labelled as "CLMS442" with two 4-tap and one 2-tap filters in order. The third one is labelled as "CLMS2" with 4 stage 2-tap filters. Clearly, the one stage LMS performs the worst among them.


LMS8
CLMS242
CLMS442
LMS8
CLMS242
CLMS442
MSE(dB)
iterations
15
10
5
0
-5
-10
-15
-20
-25
-30
-35
0
500
1000
1500
2000
2500
3000
3500
4000

(a)


LMS8
CLMS2
CLMS242
CLMS442
LMS8
CLMS2
CLMS242
CLMS442
0
-5
-10
-15
-20
-25
-30
-35
-40
0
500
1000
1500
2000
2500
3000
3500
4000

(b)



(c)

Fig. 4.8: MSE performance comparison between conventional LMS (dash-dot), CLMS2 (dash), CLMS242 (dotted) and CLMS442 (solid) for (a) model a; (b) model b; (c) model c

In Fig. 4.8(b), it can be seen that the performance of low-order cascaded LMS predictor [28] is very poor. This result is consistent with the fact that low-order cascaded predictor cannot perform effectively in audio signal modeling, as discussed in section 4.5. Although it works well in Fig. 4.8(c), it cannot be the good structure because it fails to be applied for some signals with certain models, especially the model (b) studied. For audio signal modeling, it is essential to get the uniform structure to model all classes of audio signals well so as to get the best performance on average. Therefore, it is necessary to use other typical models (b) and (c) to study the cascade structure besides the model with well separate poles.

Since low-order stages cascade structure is improper, let us look at the structures CLMS242 and CLMS442. In this simulation, the high sampling rate signal is obtained by up sampling the original signal by four times. The CLMS242 is the optimal structure for signal modeling when the simulation signal is not processed by up sampling. It can be seen that such structure still works well to modelling signals with up sampling. However, CLMS442 behaves faster convergence speed in Fig. 4.8(a) than CLMS242 which is the optimal cascade structure for original signal modeling. As mentioned above, because most signals are modeled with well separate poles, the structure CLMS442 obviously has an advantage in high sampling rate signals' modeling since it performs the best for such signals. Moreover, its performances in Fig. 4.8(b) and Fig. 4.8(c) are still acceptable and similar with CLMS242.

Importantly, the optimal structure is the structure which performs the best for all the signal models on average. From the analysis above, among the structures the optimal one obviously should be CLMS442, which is different but related with the structure CLMS242. From the simulation results and analysis, for high sampling rate audio signal modeling, we come to propose the uniform cascade structure as follows: similar with the low sampling rate signal modeling, the optimal cascade structure is composed of high-order filter in middle and low-order filters in previous and followed. The difference is that, for high sampling rate, the signals can be better modeled by increasing the order length in the low-order stage filters properly and purposely.

It is well known that LMS predictor convergence speed suffers from the big eigenvalue spread of the high correlated input signals. Long filter in first stage is not recommended, because convergence performance of such structure is heavily affected by high colored input. In our proposed structure, the first low-order stage can be used as a pre-whitening adaptive filter to reduce the eigenvalue spread. The second stage adopts a long LMS filter, which is required for most audio signals modeled with abundant poles and zeroes. In this section, we concern the high sampling rate audio signals mainly. Clearly, by up-sampling the correlation between signal samples is much increased compared with the correlation of the original signal. Therefore, to increase the order of low-order filters properly is reasonable for better modeling purpose, since the low-order filters act as the whitening filters to reduce the eigenvalue spread.

Moreover, it is proved that in cascaded FIR predictor, each stage attempts to cancel the dominant mode of its input and performs a linear prediction with a successive refinement strategy. Consequently, the eigenvalue spread becomes smaller in each stage, which results in the faster convergence. Basically, the simulation results confirm the analysis in theory.

In this section, we studied the cascaded adaptive linear predictor for high sampling rate audio modeling [35]. For lossless audio coding purpose, we proposed a uniform cascade LMS structure which can model all high sampling rate audio signals well. As similar as the low sampling rate signal modeling, the optimal cascade structure is composed of high-order filter in middle and low-order filters in previous and followed.

However, the signals with high sampling rate can be better modeled by increasing the order length in the low-order filters properly and purposely.

4.7 Application for Prediction of Audio Signals

Based on the theoretical and experimental study, we apply the proposed cascade LMS structure to audio signals sampled at different rates (48 kHz, 96 kHz and 192 kHz). We tested 51 audio clips provided by MPEG-4 with different predictors of lossless audio CODEC, which are Monkey's 3.97, cascaded LMS and TUB LPC predictor. The cascaded LMS predictor is designed with low-order and high-order adaptive filters, while using longer filters in the low-order filter stages for high sampling rate audio signals.

Table 4.1 SNR for Different Lossless Predictors

Track	**Monkey 3.97 (dB)**		**Cascade LMS (dB)**		**TUB LPC (dB)**	
	L	**R**	**L**	**R**	**L**	**R**
48kHz/16bit	28.8	29.3	29.5	30.1	28.1	28.6
48kHz/24bit	28.9	29.5	30.0	30.6	28.6	29.1
96kHz/24bit	51.5	52.0	53.9	54.8	53.5	54.2
192kHz/24bit	63.4	63.0	65.4	64.9	65.3	64.7

The average results of Signal to Noise Ratio (SNR) for different predictors are shown in the Table 4.1. From the Table 4.1, it can be seen that the SNR of the cascaded LMS is about 1dB over that of LPC used by TUB for 48 kHz. For 96 kHz and 192 kHz, the SNR of the cascaded LMS predictor is about 0.1~0.4dB over LPC of TUB. It shows that the proposed predictor in such cascade structure gives the best predictive gain among them for both low and high sampling rate audio signals. It supports our theoretical analysis that cascaded LMS predictor can outperform the LPC technique which is using Levinson-Durbin algorithm.

In fact, experimental study above gave us the guideline to design optimal cascaded LMS predictor. The principle of structure selection is to design the structure with short filters and long filters for audio signals' modeling. The number of filters and the step size selection depend on the input signals. To study the design of cascade structure, e.g. the order selection in each stage in theory should be the future work.

4.8 Summary

In this chapter, we studied theoretically an adaptive linear predictor in detail. Furthermore, according to the experimental study and theoretical analysis, we proposed a cascade structure to design the optimal adaptive linear predictor. The adaptive linear predictor with a cascade structure performs a linear prediction in terms of successive refinement in each stage, which results in fast convergence and low final MSE. Most importantly, it is given out that the MSE performance of adaptive filter, e.g. LMS, can surpass the performance of LPC, which is always considered as the performance bound of LMS.

In practical application, it has been seen the proposed predictor with cascade structure can work better than the single stage adaptive predictor and maybe the predictors in other cascade structures. Most importantly, it is observed that the adaptive linear predictor outperforms the LPC predictor on average for the prediction of audio signals sampled in different sampling rates.

CHAPTER 5

RANDOM ACCESS FUNCTION IN ALS

5.1 Introduction

Random access (RA) enables random and fast access to any part of the encoded audio signal without costly decoding of previous parts. Obviously, this is an important and practical function for audio CODEC. For example, RA enables customers to edit and play back the compressed audio signals. This function is also required in MPEG-4's CfP of lossless audio coding [2]. Therefore, in this chapter, we will discuss and implement the random access function in our CODEC.

In general, to allow random access the encoder inserts random access frames at intervals of several frames. The number of frames between two random access frames is given by

$$N_{RA\_RA} = \left\lfloor \frac{f_s \cdot t_{RA}}{N_f} \right\rfloor \tag{5.1}$$

where f_s is the sampling rate of PCM wave, such as 48kHz, t_{RA} is the random access interval in second, N_f is the frame length and operator $\lfloor\ \rfloor$ is the integer operation. The random access frame can be decoded without decoding previous frames. In other words, in those random access frames, no samples and no information from previous frames are used for prediction. Moreover, at the start of each random access frame the encoder also need insert several bytes field which specifies the distance to the next random access frame, thus enabling a fast search inside the compressed file. An example for the general bit stream structure of a compressed file with random access is shown in Fig. 5.1. The field “R” appears only at the beginning of random access frames (e.g. Frame 1, Frame 4 and each third frame) and specifies the distance (in bytes) to the next random access frame.

Header	R	Frame 1	Frame 2	Frame 3	R	Frame 4	Frame 5	···	Frame X	Non-audio	CRC

Fig. 5.1: General bit stream structure of a compressed file with random access

It is well known that samples of most audio signals have strong correlation which can be reduced considerably by linear predictor. Then the output residual error of predictor is coded by an entropy coding technique, such as Rice coding. In order to get the satisfied compression performance, the correlation must be reduced as much as possible. However, to support the random access function, the CODEC has to reconstruct perfect signals from a random access point without using any of the previous signal information. It means that to support random access for linear prediction, the loss of prediction gain has to be introduced because a number of samples at the beginning cannot be predicted sufficiently without the information of previous frame. In such case, for the traditional LPC technique, a method called “progressive prediction” has been proposed in order to enhance the compression performance in [36].

In previous chapter, we have provided an adaptive linear predictor for lossless audio coding, which can get better prediction gain than LPC technique. Therefore, we will see that with the adaptive linear prediction, the audio CODEC can get a satisfied compression performance without supporting random access. However, in random access mode, the compression performance is dropped heavily because the adaptive linear prediction suffers from more loss of performance than LPC technique in this case. The method progressive prediction which works well for LPC predictor, is not applicable for adaptive linear predictor due to the special properties of the adaptive filter, which will be discussed in following section.

In random access frame, the initial values of the adaptive linear predictor should be reset such that the decoder can reconstruct the signal without any extra information from previous frames. As the cascaded adaptive predictor does instance-based analysis and its convergence behaviour depends tightly on the initial values, e.g. the previous samples values which are used to predict the first samples of RA frames and the initial weight information of the adaptive filter, if the initial values are reset to zeros, the adaptive linear predictor need a transition period to converge to its steady state. In other words, there is existing a transition period which always contains the large errors in each RA frame, which degrades the performance of the adaptive linear predictor heavily. The proposed adaptive linear predictor is designed with a cascade structure in order to increase the convergence speed, i.e. to reduce the transition period. However, in RA mode, the problem resulting from the transition period becomes more vital for the performance. Therefore, although we have proposed an optimal adaptive linear predictor, we still have to do more to improve the compression performance while realizing the random access because of the unavoidable transition period of adaptive linear predictor.

5.2 Basic Ideas

5.2.1 Improvement of Adaptive Linear Predictor for RA mode

In Chapter 4, we have proposed an adaptive linear predictor. Meanwhile, it is also pointed out that most adaptive algorithms can be used for the single stages. For simplicity and stability, LMS algorithm is adopted in each stage in Chapter 4. However, it is well known that LMS filter suffers slow convergence for inputs with large eigenvalue spread.

In RA mode, to increase the convergence speed becomes more important since this problem exists in each RA frame. Considering RLS algorithm is less sensitive to the eigenvalue spread of input, in this section, we use the low-order RLS filter in the first stage, thus the slow convergence problem can be mitigated.

5.2.2 Separate Entropy Coding Scheme

The separate entropy coding scheme is proposed to improve the compression efficiency for RA frames in lossless audio coding with the proposed adaptive linear predictor.

In general, the error values in transient phase are always larger than the error values in steady state. The residual error of predictor will be coded by an entropy coding scheme, which is an improved Rice coding technique in our CODEC system. In section 2.3.3, it has been discussed that Rice coding is the most efficient algorithm for the signals which are Laplacian distributed. Because the error signals in steady state can be modeled with Laplacian distribution function, Rice coding technique is widely used in this application. Obviously, the error signals which are larger in transient phase can distort the probability distribution, which results in less efficiency of Rice coding. In other words, if we code the output signals of the predictor as a whole, the efficiency of Rice coding should be reduced because the supposed Laplacian distribution is distorted. Therefore, to separate the output signals as transient phase and steady state, and coding them respectively is a reasonable method to improve the compression performance for RA frames.

5.3 Separate Entropy Coding Scheme

5.3.1 A Simplified DPCM Prediction Filter

For some audio signals, except for the larger values in transient phase, an explosive divergence may happen in some RA frames at the beginning of transient phase due to the adaptive iterations [27]. It leads to the higher bit rate than in the case of continuous prediction even a separate entropy coding is used for transient phase and the steady-state residual error. Therefore, we propose a simplified DPCM prediction filter for the signals in transient phase. The purpose is to use forward prediction method to reduce the error values so as to reduce the variance and energy level of the error signals in transient phase. The prediction is given as

$$e_e(n) = e(n) - a \cdot e(n-1) \tag{5.2}$$

where $e(n)$ is the error signal in transient phase, $e_e(n)$ is the output of DPCM filter and a is the coefficient. For simplicity, we set $a = 1$ in application.

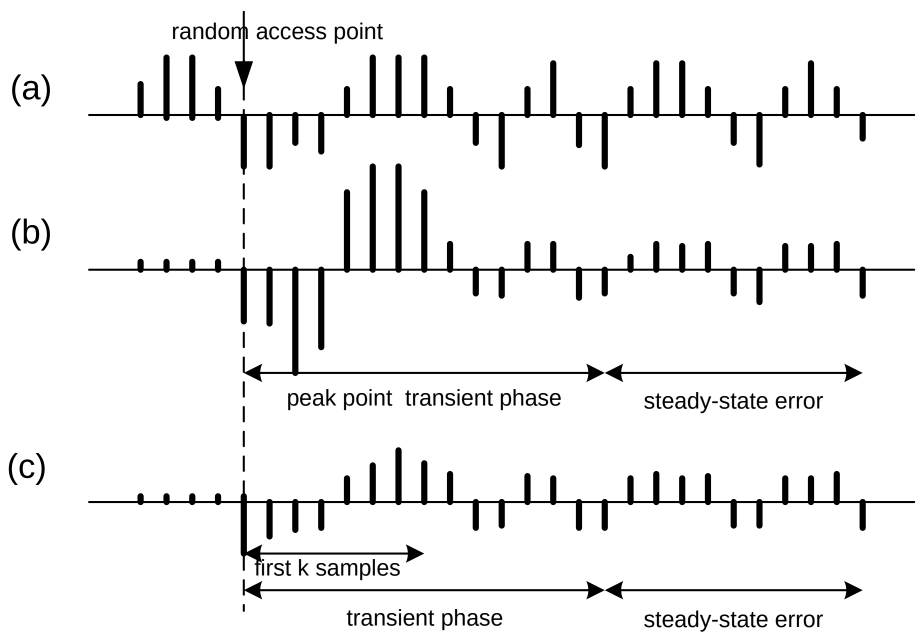


Fig. 5.2: Prediction in random access frames: (a) original signal; (b) residual for an adaptive linear prediction; (c) residual for DPCM and residual for adaptive linear prediction from k+1th sample

Figure 5.2 illustrates the basic concept of separate entropy coding scheme. In (b) it can be seen that the simplified DPCM filter may not process all of the signal samples in transient phase. We suppose the length of the samples which are processed is k. In practical application, $k = K + 1$ for simplicity, where K is the order of the adaptive filter in the first stage.

From the experimentations, the proposed DPCM filter does reduce the residual values at the beginning of random access frames as expected. However, since only a few samples are engaged in this method, it can be excepted that the performance improvement is severely limited. Table 5.1 shows the compressed file size with DPCM and without DPCM in RA mode. We also calculate the relative improvement which is about 0.004% on average. The relative improvement is calculated by

$$c_{rel} = 100\% \cdot \frac{OldCompressedSize - NewCompressedSize}{OldCompressedSize} \tag{5.3}$$

where $OldCompressedSize$ is the compressed file size without DPCM, while $NewCompressedSize$ is the compressed file size with DPCM. For a simple demonstration purpose, we just set $k = K + 1$ and $a = 1$ to get the results. It is believed that there can be further improvement if the optimal values of k and a are used.

Table 5.1 Relative Improvement with DPCM

Test Set	RA without DPCM	RA with DPCM	
	Size	Size	Rel.
48k16b	38,648,069	38,645,724	0.006%
48k24b	81,325,468	81,323,111	0.003%
96k24b	119,519,614	119,514,351	0.004%
192k24b	77,948,128	77,946,745	0.002%
Total	**317,441,279**	**317,429,931**	**0.004%**

5.3.2 Separate Entropy Coding

In the current random access scheme, as seen in Fig. 5.2, the residual error for the first K samples as well as that in transient phase have larger values, compared to the steady-state error, thus separate entropy coding is required.

We propose to use Rice coding with coding index (length of low part bits) $s = res - c$ to code the first $K+b$ samples, while the remaining residual values $d(K+b+1)$ to $d(frameLength)$ are coded with another optimal Rice coding index which is calculated automatically based on Equation (2.7). Here res is the resolution of the PCM audio signals, e.g. 16 or 24bits. A summary of all Rice coding indices is given in Table 5.2, where $d(n), n = 0, \cdots, frameLength - 1$ is the residual error of the adaptive linear predictor.

Table 5.2 Code Parameters for Different Sample Positions

Sample / Residual	Coding Index
$d(0), d(1), \cdots, d(K+b-1)$	$res - c$
$d(K+b), \cdots, d(frameLength-1)$	Optimal coding index

5.3.3 Compression Performance

In the following tests, compression performance is evaluated for the test set, 51 audio clips which are provided by MPEG. All compression results are given in terms of total compressed file size in bytes and total compression rate. The total compression rate is given by,

$$C = 100\% \cdot \frac{total\_compressed\ size}{total\_original\ size} \tag{5.4}$$

In RA mode, the interval between RA frames is set to be 0.5 second, which is required in MPEG-4's CfP as the coarsest granularity [2]. In such case, if the frame length is 4096 the numbers of interval frames between two RA frames N_{RA_RA} can be calculated from Equation (5.1) and shown in Table 5.2. For example, for the PCM audio data sampled with 48 kHz, there is a RA frame every 5 consecutive frames. The original file size of each wave clip and the total original file size are also given in Table 5.2. Thus the total number of RA frames for each wave clip can be calculated by

$$N_{RA} = \left\lceil \frac{N_{frames}}{N_{RA\_RA}} \right\rceil \tag{5.5}$$

where operator $\lceil \quad \rceil$ is the integer operation which is to get the integer lager or equal than $\frac{N_{frames}}{N_{RA\_RA}}$. N_{frames} is the number of frames in each wave clip which is given by

$$N_{frames} = \left\lceil \frac{N_{samples}}{N_f} \right\rceil \tag{5.6}$$

where the number of samples $N_{samples}$ is given by

$$N_{samples} = \frac{filesize - 44}{channel \cdot \frac{res}{8}} \tag{5.7}$$

where res is the resolution and $channel$ is the channels of wave clip. All of the wave clips used in test are dual-channel, i.e. $channel = 2$. Because the header size of wave format file uses 44 bytes, number 44 is subtracted from the file size at first. Clearly, $N_{samples}$ is the number of samples in each channel. From Equations (5.5), (5.6) and (5.7), the number of RA frames N_{RA} can be calculated and shown in Table 5.2 as well.

Table 5.2 Descriptions of the Test Set

Test Set	Files	Size of Each File (bytes)	Total File Size (bytes)	Interval Frames N_{RA_RA}	Frames of Each File	RA Frames N_{RA}
48k16b	15	5760048	86400720	5	352	1065
48k24b	15	8640050	129600750	5	352	1065
96k24b	15	17280044 /17280050 *	259200696	11	704	960
192k24b	6	34560044	207360264	23	1407	372
Total	**51**	**-**	**682562430**	**-**	**-**	**3462**

* All together 15 files, 9 of them each has a file size of 17280044 bytes, another 6 files each has a file size of 17280050 bytes.

Table 5.3 Compression Comparison between No RA and RA without Separate Entropy Coding

Test Set	Original	Continuous Coding (no RA)		RA without Separate Entropy Coding	
		Size	Ratio	Size	Ratio
48k16b	86400720	38,469,974	44.53%	38,742,820	44.84%
48k24b	129600750	81,103,855	62.58%	81,430,228	62.83%
96k24b	259200696	119,084,965	45.94%	120,016,633	46.30%
192k24b	207360264	77,884,516	37.56%	78,287,224	37.75%
Total	**682562430**	**316,543,310**	**46.38%**	**318,476,905**	**46.66%**

Table 5.3 illustrates the compression performance of continuous coding (no RA mode) and RA mode without separate entropy coding scheme. It can be seen that about 0.3% performance drop is suffered for RA mode. The performance of RA mode with separate entropy coding scheme is shown in Table 5.4, from which it can be seen that the compression performance is improved about 0.15% compared to the coding scheme without separate entropy coding. Compared with the original performance drop of 0.3%

the relative improvement 0.15% is rather significant because the performance is improved by two times. Moreover, the number of residual values affected by the method of coding scheme in RA mode is quite small. Therefore the performance improvement should be quite limited.

Table 5.4 Compression Comparison among No RA and RA without/with Separate Entropy Coding

Test Set	Original	Continuous Coding (no RA)	RA without Separate Entropy Coding	RA with Separate Entropy Coding	
		Ratio	Ratio	Size	Ratio
48k16b	86400720	44.53%	44.84%	38,648,069	44.73%
48k24b	129600750	62.58%	62.83%	81,325,468	62.75%
96k24b	259200696	45.94%	46.30%	119,519,614	46.11%
192k24b	207360264	37.56%	37.75%	77,948,128	37.59%
Total	**682562430**	**46.38%**	**46.66%**	**317,441,279**	**46.51%**

Table 5.5 Compression Comparison between TUB Encoder and the Proposed Encoder

Test Set	Original	TUB optimal encoder (RA mode)		RA with Separate Entropy Coding		
		Size	Ratio	Size	Ratio	Rel. vs. TUB
48k16b	86400720	39,079,276	45.23%	38,648,069	44.73%	1.10%
48k24b	129600750	81,774,080	63.1%	81,325,468	62.75%	0.55%
96k24b	259200696	120,264,328	46.4%	119,519,614	46.11%	0.62%
192k24b	207360264	78,211,226	37.72%	77,948,128	37.59%	0.34%
Total	**682562430**	**319,328,910**	**46.78%**	**317,441,279**	**46.51%**	**0.59%**

Table 5.5 compares the compression performance of TUB optimal encoder and our proposed encoder for 0.5 second random access. The performance of TUB optimal

encoder, which is reported in [37], is regarded as the benchmark. Our proposed encoder achieves 0.27% better compression ratio performance than TUB optimal encoder which uses LPC technique. In table 5.5, we also give out the relative improvement according to Equation (5.3). Here *OldCompressedSize* is the compressed file size from TUB encoder, while *NewCompressedSize* is the compressed file size from our encoder. The relative improvement is about 0.59% on average.

5.3.4 Discussion

In this section, we propose a separate entropy coding scheme for adaptive linear prediction in order to counteract the performance drop in RA mode. With the proposed scheme, the performance improvement is significant. Most importantly, the proposed encoder outperforms TUB optimal encoder which is regarded as the benchmark not only in continuous coding but also in RA mode.

Basically, the proposed separate entropy coding scheme does not lead to an increase in computational complexity, since the length of the samples processed by Rice coding or BGMC is same. Moreover, the parameters b and c are preset in encoder and decoder respectively. Therefore, the bits are saved without transferring these parameters.

However, since the parameters b and c are preset, it means that they are same in every RA frame. It is obvious that the preset values of the parameters are not accuracy or optimal for separate entropy coding scheme because the convergence behavior is always different in every RA frame. Therefore, we try to choose the optimal b and c in each RA frame, which will be discussed in following section.

5.4 An Improvement of Separate Entropy Coding Scheme

In this section, we propose an improvement method based on the separate entropy coding scheme which is discussed in last section. Considering the different behavior of convergence in every RA frame, we try to choose the optimal b and c in order to get better compression performance. However, in such case, the parameters b and c have to be transferred in bit stream for each RA frame since they could be in different values. The more bits are needed if the value range of b or c is wider. Therefore, the trade-off should be made between the range of the parameters and the corresponding compression ratio so that the optimal performance is reached with this method.

In the following test, we use 6 bits for b and 4 bits for c. It means that b is an integer from 0 to 63 and c is an integer from 0 to 15. The compression ratio and the relative improvement (about 0.011% on average) are shown in Table 5.6. However, in this test the values of parameters b and c are suboptimal because they are not chosen by full search scheme, i.e. the adaptive process will be terminated when worse performance is met in search path. Obviously, the purpose of partial search scheme is to save the encoding time in encoder. In fact, in such case the computational complexity increased in encoder can almost be neglected, while no computational complexity is increased in decoder.

When the optimal values of parameters b and c are chosen with full search method, Table 5.7 shows the corresponding compression ratio and relative improvement (about 0.021% on average) are improved as expected. Of course, the computational complexity is increased inevitably in encoder. The wider the ranges of parameters b and c are, the longer the encoding time will be. However, still no computational complexity is increased in decoder.

Table 5.6 Compression Comparison between Encoders with and without Improvement (partial search)

Test Set	Original	RA with Separate Entropy Coding		RA with improved Separate Entropy Coding		
		Size	Ratio	Size	Ratio	Rel.
48k16b	86400720	38,648,069	44.73%	38,644,590	44.73%	0.01%
48k24b	129600750	81,325,468	62.75%	81,320,958	62.75%	0.0056%
96k24b	259200696	119,519,614	46.11%	119,498,642	46.10%	0.018%
192k24b	207360264	77,948,128	37.59%	77,940,684	37.59%	0.01%
Total	**682562430**	**317,441,279**	**46.51%**	**317,404,874**	**46.50%**	**0.011%**

Table 5.7 Compression Comparison between Encoders with and without Improvement (full search)

Test Set	Original	RA with Separate Entropy Coding		RA with improved Separate Entropy Coding (full search)		
		Size	Ratio	Size	Ratio	Rel.
48k16b	86400720	38,648,069	44.73%	38,639,924	44.72%	0.02%
48k24b	129600750	81,325,468	62.75%	81,313,300	62.74%	0.015%
96k24b	259200696	119,519,614	46.11%	119,489,652	46.10%	0.025%
192k24b	207360264	77,948,128	37.59%	77,932,183	37.58%	0.02%
Total	**682562430**	**317,441,279**	**46.51%**	**317,375,059**	**46.50%**	**0.021%**

In this section, we did a fundamental survey for the improvement of separate entropy coding scheme. The method proposed in this section, which is to search the suboptimal or optimal values of b and c, is proved to be efficient to improve the compression ratio in RA mode. Although the improvement in our experiments is not significant, the method is valuable considering that a little and no computational complexity is increased in encoder and decoder respectively. Moreover, the test results above are not optimal. It is expected that better results can be gained with this improved separate entropy coding scheme.

5.5 Summary

In this chapter, we implement successfully the random access function in the proposed CODEC for lossless audio coding, which uses the adaptive linear prediction instead of LCP technique.

Since the proposed adaptive linear predictor gets the higher prediction gain than the predictor used in LPC technique, it is no doubt that the former outperforms the latter in compression performance. In continuous coding (no RA mode), the advantage of the proposed prediction technique is outstanding compared with LPC technique. However, in RA mode this advantage in compression performance is much weakened. It is inevitable that the prediction gain is actually decreased mainly because of the transient phase of the adaptive linear predictor in each RA frame.

In order to improve the performance of the proposed CODEC in RA mode, we discuss and propose the separate entropy coding scheme, which is proved as a promised method. The basic idea is to code residual errors in transient phase and steady state with different codes. Several optional methods and their performances are also discussed. In addition to the separate coding with different codes, the compression performance can get more improvement by introducing a simplified DPCM prediction filter.

The random access function is implemented successfully in the proposed CODEC. No matter in continuous coding or RA mode, its compression performance surpasses that of the state-of-the-art TUB optimal encoder which uses the LPC technique.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter makes the summary and conclusion of this thesis and recommends for future work. The subject of this project is to propose the lossless coding techniques for digital audio data.

6.1 Conclusion

In the beginning we have previewed the background of data compression and lossless audio coding. Because of the high correlation among the audio signals, some kind of predictor must be applied before entropy coding. We discussed the LPC technique and Rice coding which are used widely and efficiently in this application. As a benchmark of performance, we have introduced the state-of-the-art lossless audio CODECs, Monkey's Audio Coding and ALS CODEC from TUB.

The overview structure of the CODEC system we proposed has been discussed in Chapter 3. Among the structure the predictor is the main part discussed in this thesis. Instead of the LPC technique, we proposed adaptive linear prediction technique in audio coding. In Chapter 4 we successfully designed the linear predictor with adaptive linear filters which work together in a cascade structure. The proposed cascade structure is found by lots of experiments and analysis in audio signal modeling. With such a cascade structure, the experimental data shows that the proposed adaptive linear predictor can obtain higher prediction gain than LPC technique for audio signals. Meanwhile, we gave out the MSE performance bound of the applied adaptive filter without the independence assumption, which shows that the MSE of adaptive filter may be lower than that of LPC. Furthermore, the detailed theoretical analysis is given out to prove that the cascaded adaptive predictor can perform a linear prediction with a successive refinement strategy, which means that if each stage converges to its steady-state value, lower MSE and faster convergence speed is possible with the increasing of stages.

In Chapter 5, we have implemented the RA feature in the proposed audio coding system. In each RA frame, the transient phase can be reduced by increasing the convergence speed, but is inevitable. Since the transient phase of adaptive predictor degrades the compression ratio, to guarantee the compression performance is more difficult in the proposed system. We have proposed separate entropy coding scheme while implementing the RA function. The basic idea is to coding the residuals of transient phase and steady-state phase with different code words. Moreover, the DPCM filter is proved to be effective to be applied in transient phase.

6.2 Future Work

Despite achieving significant results with the proposed cascaded adaptive linear prediction technique, there is still much room for improvement. Moreover, based on this project many interesting ideas can be investigated.

1. Adaptive Prediction Algorithm

In the proposed framework, any adaptive prediction filter can be applied in each stage. It is possible that some other adaptive algorithm can obtain better performance than what we get. For high compression performance, fast convergence and low MSE are required. Moreover, the algorithm should be stable and simple so that it is suitable for practical application. Therefore, other prediction algorithms can be investigated in lossless audio coding.

2. The Cascade Structure

According to the audio signal modeling, we proposed a cascade structure which outperforms the LPC. However, much work has not been done in this area. For example, how many stages can be the optimal for audio signals? How about the precise order selection in each stage? Is there any precise guidance to design the cascaded predictor?

3. Random Access

We have proposed some ideas in this thesis to improve the compression performance for RA implementation. However, we have some other interesting ideas to get the further improvement, e.g. transferring the coefficients of predictor in RA frame.

4. Inter-channel De-correlation

It is well known that most audio applications deal with the multi-channel audio data streams [38]. Therefore, inter-channel de-correlation is an important topic in lossless audio coding. With a good de-correlation method implemented into the proposed system, the compression performance can be further improved.

5. The Complexity and Speed

The complexity of the proposed CODEC system mainly depends on the complexity of the adaptive algorithms applied and the length of the predictor. In this thesis, the analysis of complexity has not been discussed. However, it is well known that most of the adaptive algorithms require a lot of calculations and an efficient audio prediction requires a high order predictor. Therefore, to design a fast lossless audio coding CODEC with adaptive linear prediction technique is a challenging task in the future.

REFERENCES

[1] ISO/IEC 14496-3:2001, "Information technology – Coding of audio-visual objects – Part 3: Audio," *International Standard*, 2001.

[2] ISO/IEC JTC1/SC29/WG11 N5040, "Call for Proposals on MPEG-4 Lossless Audio Coding," the 61st MPEG Meeting, Klagenfurt, Austria, July 2002.

[3] "Monkey's Audio", http://www.monkeysaudio.com/.

[4] ISO/IEC JTC1/SC29/WG11 M9781, T. Liebchen, "Detailed technical description of the TUB 'lossless only' proposal for MPEG-4 Audio Lossless Coding," the 64th MPEG Meeting, Awaji Island, Japan, March 2003

[5] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol.27, pp. 379-423 and 623-656, 1948.

[6] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Norwell, MA: Kluwer Academic, 1992.

[7] M. Hans and R. W. Schafer, "Lossless Compression of Digital Audio," IEEE Signal Processing Magazine, pp. 21-32, July 2001.

[8] X. Lin, *et al*, "A Novel Prediction Scheme for Lossless Compression of Audio Waveform," in Proc. ICME 2001, Tokyo, Japan, August 2001, pp. 197-200.

[9] R. Yu, C.C. Ko, S. Rahardja, and X. Lin, "An RLS-LMS Algorithm for Lossless Audio Coding," the 37th Asilomar Conference on Signals, Systems and Computers, USA, November 2003.

[10] URL: http://www.lossless-audio.com/index.htm.

[11] G. Schuller, B. Yu and D. Huang, “Lossless Coding of Audio Signals Using Cascade Prediction,” in Proc. ICASSP 2001, Utah, USA, May 2001, pp.3273-3277.

[12] M. Hans and R.W. Schafer, “AudioPaK— An Integer Arithmetic Lossless Audio Codec,” in Proc. Data Compression Conf., Snowbird, UT, 1998, pp. 550.

[13] A. Wegener, “MUSICompress: Lossless, Low-MIPS Audio Compression in Software and Hardware,” in Proc. Int. Conf. Signal Processing Applications and Technology, 1997. URL: http://members.aol.com/sndspace.

[14] M. Purat, T. Liebchen, and P. Noll, “Lossless Transform Coding of Audio Signals,” in Proc. the 102nd AES, Munich, Germany, 1997, preprint 4414.

[15] ISO/IEC JTC1/SC29/WG11 M9134, R. Yu, X. Lin and S. Rahardja, “Advanced Audio Zip – Scalable Perceptual and Lossless Audio Codec,” the 63rd MPEG Meeting, Awaji Island, Japan, December 2002.

[16] P. Craven and M. Gerzon, “Lossless Coding for Audio Discs,” Journal AES, vol. 44, no. 9, pp. 706-720, September 1996.

[17] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice-Hall, 1978.

[18] N. Jayant and P. Noll, *Digital Coding of Waveforms, Principles and Applications to Speech and Video,* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[19] G. Sullivan, “Efficient Scalar Quantization of Exponential and Laplacian Random Variables,” IEEE Trans. on Information Theory, vol. 42, pp. 1365-1374, September 1996.

[20] M.D. Paez and T. H. Glisson, “Minimum Mean-squared-error Quantization in Speech PCM and DPCM Systems,” IEEE Trans. on Communications, vol. 20, pp. 225-230, 1972.

[21] R. F. Rice, "Some Practical Universal Noiseless Coding Techniques," Technical Report, JPL-79-22, Jet Propulsion Laboratory, Pasadena, CA, March 1979.

[22] S.W. Golomb, "Run-Length Encodings," IEEE Trans. on Information Theory, vol.12, pp. 399-401, 1966.

[23] T. Robinson, "SHORTEN: Simple Lossless and Near-lossless Waveform Compression," Cambridge Univ. Eng. Dept., Cambridge, UK, Technical Report 156, 1994.

[24] A. Bruekers, A. Oomen, and R. van der Vleuten, "Lossless Coding for DVD Audio," in Proc. the 101st AES, Los Angeles, CA, November 1996, preprint 4358.

[25] ISO/IEC JTC1/SC29/WG11 M11061, Tilman Liebchen and Yuriy A. Reznik, "Proposed Text of ISO/IEC 14496-3:2001/PDAM 4, Audio Lossless Coding (ALS)," the 69th MPEG Meeting, Redmond, USA, July 2004.

[26] E.N. Gilbert and E.F. Moore, "Variable-Length Binary Encodings," Technical Report, Bell System Technical Journal, 1959, pp. 932-967.

[27] S. Haykin, *Adaptive Filter Theory*, 3rd edition, Prentice-Hall, 1996.

[28] P. Prandoni and M. Vetterli, "An FIR Cascade Structure for Adaptive Linear Prediction," IEEE Trans. on Signal Processing, vol. 46, no.9, pp. 2566-2571, September 1998.

[29] P. Craven, M. Law and J. Stuart, "Lossless Compression Using IIR Prediction Filters," in Proc. the 102nd AES, Munich, Germany, 1997, preprint 4415.

[30] C. Montgomery, "Personal Communications on OggSquish," August 1997. URL: http://www.xiph.com.

[31] Huang Dong Yan, Su Xin Rong and A. Nallanathan, "Characterization of a Cascade LMS Predictor," ICASSP 2005, Philadelphia, Pennsylvania, USA, March 2005.

[32] K.J. Quirk, L. B. Milstein and J. R. Zeidler, "A Performance Bound for the LMS Estimator," IEEE Trans. on Information Theory, vol. 46, no. 3, pp. 1150-1158, May 2000.

[33] Huang Dong Yan and Su Xin Rong, "A Performance Bound for a Cascade LMS Predictor," ISCAS 2005, Kobe, Japan, May 2005.

[34] Huang Dong Yan, Su Xin Rong and A. Nallanathan, "A Cascade LMS Structure for Audio Signals Modeling", submitted to ICME 2005, Amsterdam, Netherlands, July 2005.

[35] Su Xin Rong, Huang Dong Yan and A. Nallanathan, "High Sampling Rate Audio Signal Modeling," the 38th Asilomar Conference on Signals, Systems and Computers, USA, November 2004.

[36] Takehiro Moriya, Dai Tracy Yang and Tilman Liebchen, "Extended Linear Prediction Tools for Lossless Audio Coding," in Proc. ICASSP 2004, Quebec, Canada, May 2004.

[37] ISO/IEC JTC1/SC29/WG11 N6676, "Status of Performance and Complexity of Lossless Audio Coding Architectures," the 70th MPEG-4 Meeting, Spain, October 2004.

[38] T. Liebchen, "Multichannel Lossless Audio Coding," the 113th AES Convention, USA, October 2002.