國立交通大學

資訊工程學系

碩士論文

MPEG-4 AAC 中 高效能 TNS 的設計

Temporal Noise Shaping Design for MPEG 4 Advanced Audio Coding

研究生：張子文

指導教授：劉啟民 教授

李文傑 教授

中華民國 九十三 年 六 月

MPEG-4 AAC 中 高效能TNS的設計

Temporal Noise Shaping Design for MPEG 4 Advanced Audio Coding

研究生：張子文　　Student：Tzu-Wen Chang

指導教授：劉啟民　　Advisor：Dr. Chi-Min Liu

李文傑　　Dr. Wen-Chieh Lee

國立交通大學

資訊工程系

碩士論文

A Thesis
Submitted to Institute of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National ChiaoTung University
in partial Fulfillment of the Requirements
for the Degree of Master in
Computer Science and Information Engineering
June 2004
HsinChu, Taiwan, Republic of China

中華民國九十三年六月

MPEG-4 AAC 中 高效能 TNS 的設計

學生：張子文

指導教授：劉啓民 博士
李文傑 博士

國立交通大學資訊工程所碩士班

中文論文摘要

TNS 是用來消除由激發訊號所造成 pre-echo 現象的一個模組，它可以控制量化誤差並且將誤差塑形在遮避能力較大的激發訊號裡，進而改進音樂品質。這篇論文會闡明 TNS 造成的三種人爲雜訊。第一種人爲雜訊會在激發訊號的邊界產生一個很大的噪音，像是 Gibbs phenomenon。第二種則是在時域訊號上的特定位置，會產生一個不尋常的噪音。最後一種則是隨著 order 數愈高，不尋常的噪音則會愈明顯。因此，這篇論文會提出一個有效率且能降低這三種人爲雜訊的方法。

Temporal Noise Shaping Design for MPEG 4 Advanced Audio Coding

Student: Tzu-Wen Chang Advisor: Dr. Chi-Min Liu

Dr. Wen-Chieh Lee

Institute of Computer Science and Information Engineering
National ChiaoTung University

ABSTRACT

Temporal noise shaping has been defined in MPEG-4 AAC to control the pre-echo noise in attack signals. The module, which is especially important for the MPEG-4 Low Delay AAC due to the absence of window switching mechanism, can shape and control quantization noise spread to improve the quality under bit rate constraint. However, this paper illustrates that the TNS will introduce three artifacts. The first artifact is similar to the Gibbs phenomenon which has high noise level occurring at the edge of the attack signal. The second effect is the time-domain aliasing noise which has unusual noise at a distance from the attack time frame. The third is the noise spreading with the TNS filter orders. This thesis will propose the efficient TNS method which shapes noise with good concerns on the above three artifacts. Also, we provide an efficient computing method to activate the TNS. Both subjective and objective tests are conducted to illustrate the improvement over existing TNS methods.

致謝

感謝劉啓民老師兩年來的栽培及李文傑博士給予的指導，實驗室的楊宗瀚學長、同學許瀚文、蕭又華、彭康硯和邱挺,以及學弟陳立偉和蘇明堂的協助,在研究上提供我寶貴的意見，讓我在專業知識及研究方法獲得非常多的啓發。

最後，感謝我的父母與家人及系上同學, 在我研究所兩年的生活中，給予我無論在精神上以及物質上的種種協助，使我能全心全意地在這個專業的領域中研究探索在此一併表達個人的感謝。

Contents

Figure list

Table List

Chapter 1 Introduction

ISO/IEC MPEG-2/4 Advanced Audio Coding (AAC) [1] [2] is the latest MPEG standard on perceptual audio coding designed for many broadcast and electronic music-distribution applications, which has been viewed as the successor of MPEG-1/2 Layer Ⅲ in the new multimedia standard. AAC was developed by the MPEG group that includes Dolby, Fraunhofer (FhG), AT&T, Sony, and Nokia—companies that have also been involved in the development of audio codec such as MP3 and AC3. Compared to MP3, AAC has achieved higher quality and lower bit rate requirement.

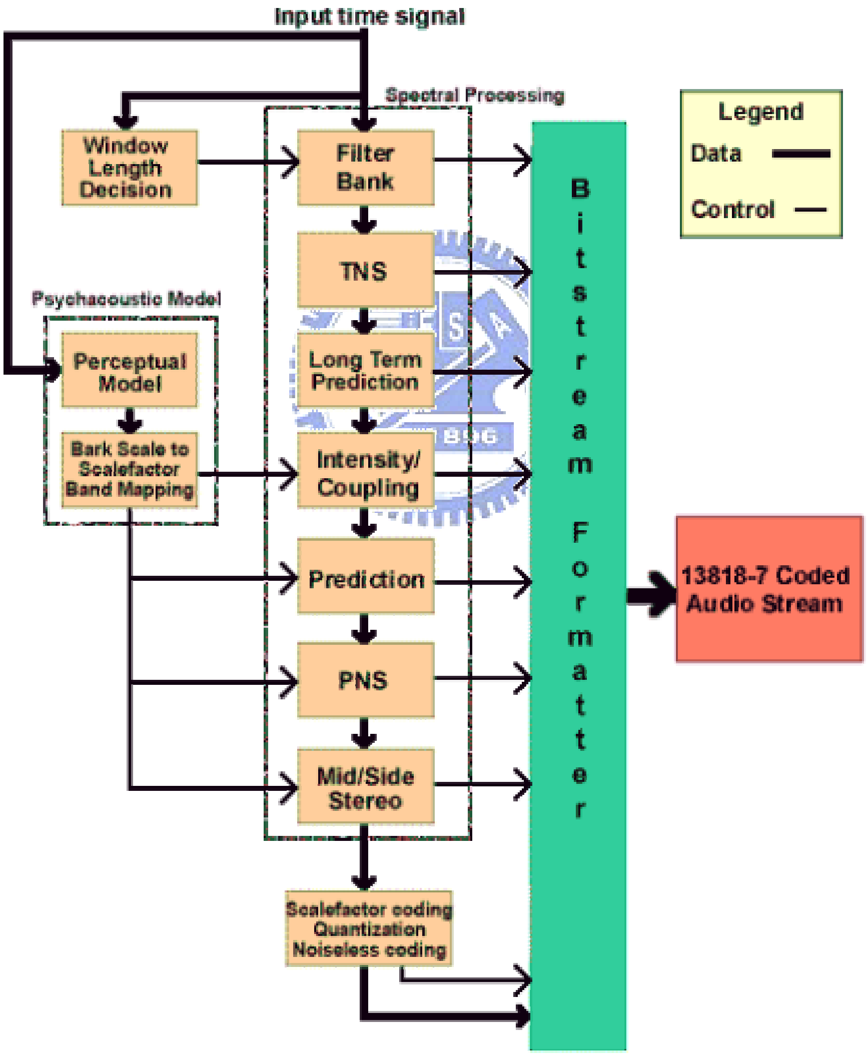


Figure 1: AAC encoder block diagram.

Figure 1 illustrates the flowchart of the AAC encoder. For flexible encoding, AAC supports three kinds of profiles (see as Table 1): “Main profile”, Low-complexity profile and Scalable sampling rate profile”, to satisfy different quality and complexity requirement that users want.

Table 1: Three kinds of AAC profiles.

Profile	Setting
Main	➢ Turn off "Gain control" ➢ The maximum order of TNS is 20 ➢ Turn on "Prediction"
Low-Complexity (LC)	➢ Turn off "Gain control" ➢ The maximum order of TNS is 12 ➢ Turn off "Prediction"
Scalable Sampling Rate	➢ Turn on "Gain control" ➢ The maximum order of TNS is 12 ➢ Turn off "Prediction"

In Figure 1, temporal noise shaping (TNS) [3] [4] [5] [6] [7] is introduced to ease the pre-echo noise caused by attack signals. Although the TNS module can shape and control the quantization noise spread to improve the signals quality, the TNS introduces basically three artifacts. The three artifacts should be carefully controlled when applying the TNS. This thesis investigates the three artifacts. Also, this thesis presents an efficient TNS detection mechanism. The mechanism provides merits in both complexity and quality.

This thesis is organized as follows. Chapter 2 introduces some fundamental knowledge that the reason why the pre-echo phenomenon occurs, the blocking and window switching and the principles of TNS. Chapter 3 introduces the MDCT filterbank and investigates the three artifacts by TNS. Chapter 4 presents the efficient temporal noise shaping method. Chapter 5 considers both the subjective and objective measurement on the new TNS switch mechanism. The objective test is conducted based on the recommendation system by ITU-R Task Group 10/4. Chapter 6 gives a conclusion on this thesis.

Chapter 2 Backgrounds

This chapter explains the “Pre-echo” phenomenon in the perceptual audio coding and introduces the block switching method and the principles of the temporal noise shaping to ease the pre-echo phenomenon.

2.1 Pre-echo Phenomenon

The perceptual audio coding plays an important role for many types of audio codec now. The knowledge of the psychoacoustics is used to eliminate the redundancy and to get the best audio quality in the limited bits. Here introduces the temporal masking effect, briefly.

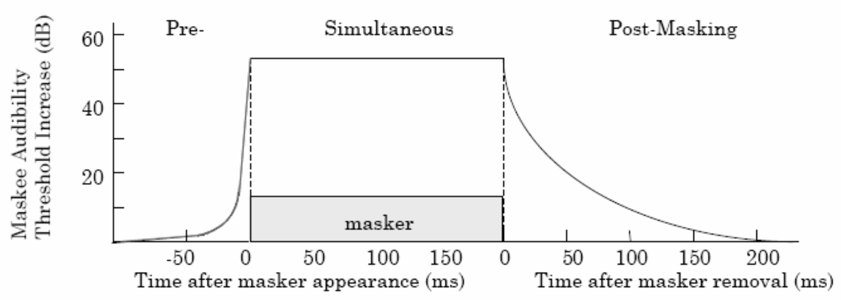


Figure 2: Post-masking, pre-masking and simultaneous masking [8].

The temporal masking effect can be divided into three parts: post-masking, pre-masking and simultaneous masking. From Figure 2, the duration of effective masker of the post-masking and the pre-masking are about 50 ms and 5 ms.

In the perceptual audio coding, the mapping from the time domain signal to the spectral signal is important. To decide the size of the block is a major challenge. When the size of the block is bigger, for the stationary signal, the redundancy is eliminated more easily and the quantization error is smaller. However, if the block of the signal to be coded has a strong component, like transient signals, the block size should be small. It’s because that the signal with the big energy has stronger masking effect than one with the small energy. If a block has the two different energy signals at the same time, the psychoacoustic model will ignore the signal with the small energy and pass the value of SMR (Signal-to-Masking Ratio) dominated by the signal with the big energy to the quantization. Therefore, the quantization will consider both of them as the big energy signal and introduce big error to them. But, the small energy of the signal can’t mask the noise and lead to the bad quality.

In the AAC, the size of the long block is 2048 samples. Under the sampling rate of 48 kHz, the duration of the block is around 43 ms. If the signal with the big energy appears in the front of the block, the noise will be masked by the post-masking effect. Otherwise, if the signal is in the behind of the block, the noise can't be masked and heard by human ears. It's called as "pre-echo" phenomenon. Figure 3 is an example.

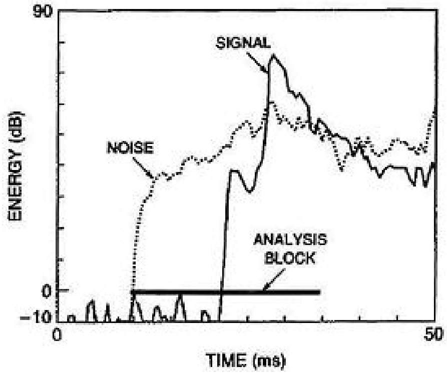
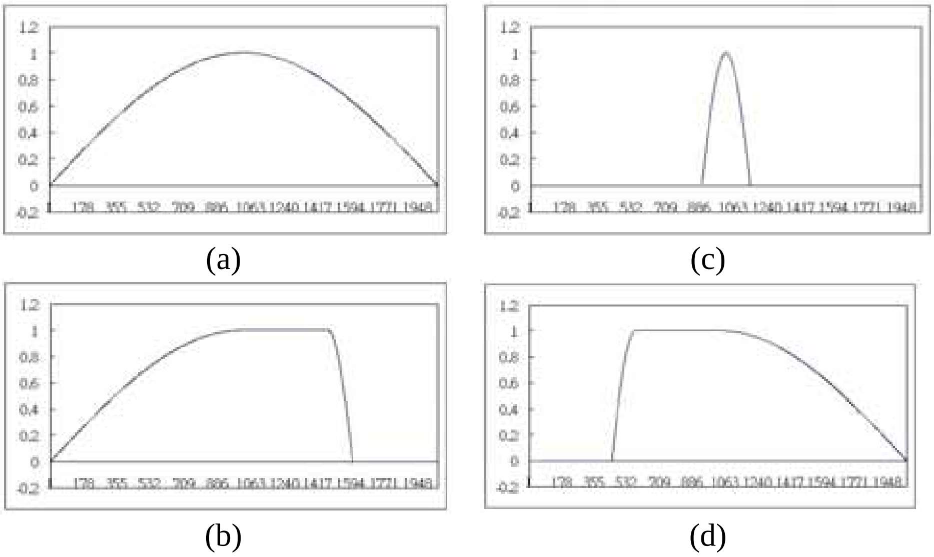


Figure 3: The example of the pre-echo phenomenon.

2.2 Window and Block Switching

An intuitive method to avoid the pre-echo is using a short block to encode the transient signal. In AAC, it provides the two window shapes and four kinds of window sequences. The two window shapes are the Kaiser - Bessel derived (KBD) window and the normal sine window. And, Figure 4 illustrates these four window sequences.



(a) (c)

(b) (d)

Figure 4: Four types of windows used for MDCT in MPEG AAC (a) ONLY_LONW Window (size 2048) (b) LONG_START Window (size 2048) (c) EIGHT_SHORT (size 256) (d) LONG_STOP Window (size 2048).

Figure 5 illustrates the process, which the eight short windows encodes the attack signal and the long start window and the long stop window will appear in the front and in then behind of the eight short windows, to encode the attack signal with the short block such that the pre-echo phenomenon can be controlled at the small block. However, from the temporal masking effect theory, the short duration time of noise can't be heard by the human's ear. But, the change of the eight short windows results in a problem—one frame delay. For the two way communication, the one frame delay will be a serious issue. Like MPEG-4 Low Delay AAC [9], in order to lower the delay time, it doesn't support the window and block switching. TNS plays an important role to ease the pre-echo in MPEG-4 Low Delay AAC.

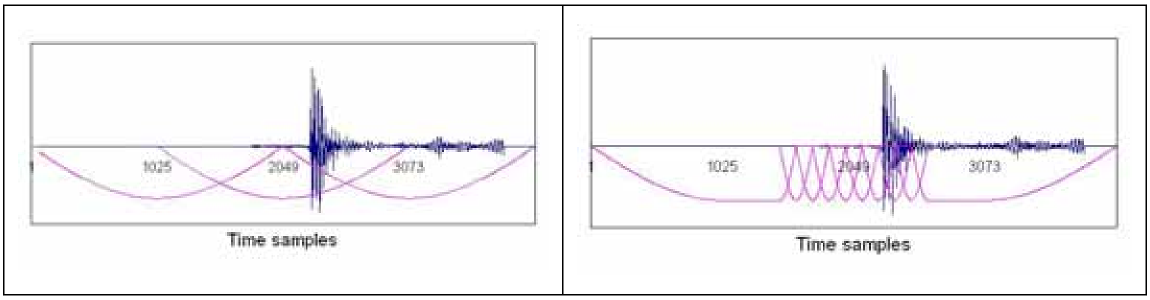


Figure 5: The design of the window and block switching.

2.3 Temporal Noise Shaping

Another disadvantage of the window switch method needs additional complexity into the codec and complicates the structure. To get rid of the disadvantages, the temporal noise shaping module was introduced in [4]. It can be considered as an extension of the basic codec schema and an insertion between the filterbank and quantization. Figure 6 illustrates the simplified flowchart of AAC codec, where the filterbank is composed as the sin window and MDCT.

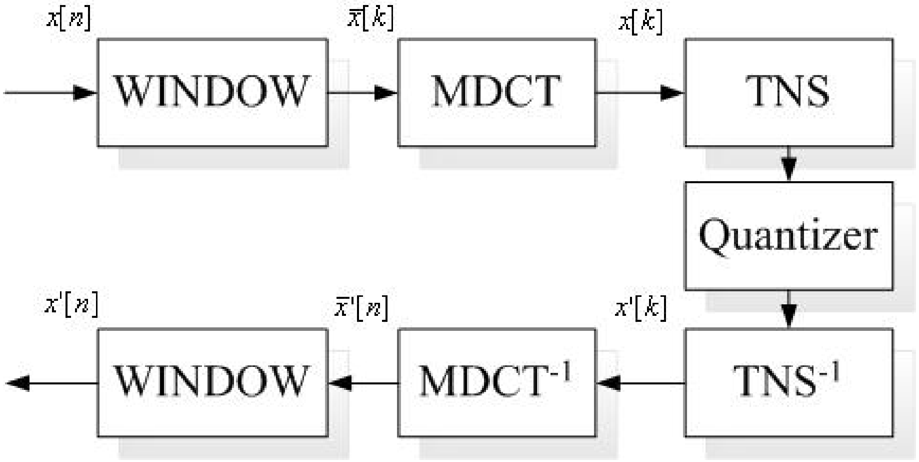
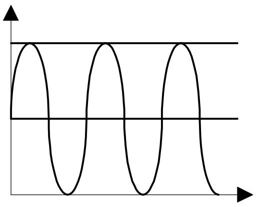
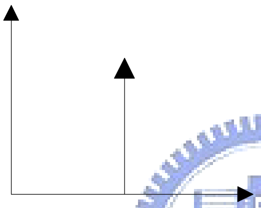
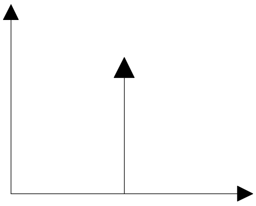
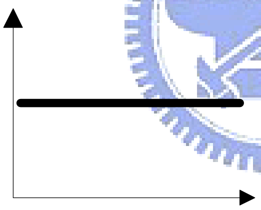


Figure 6: The simplified flowchart of AAC codec

2.3.1 **Principles of TNS**

The TNS module is based on the two principles. First, TNS is based on the consideration of the time/frequency duality between spectral envelope and Hilbert envelope [4] [7]. From Table 2, if the signal is flatter in the frequency domain, the prediction gain will be higher. The corresponding signal in the time domain will be a transient signal more possibly. In the AAC standard, it uses open-loop predictive coding to get the prediction gain information determining whether the signal is an attack signal or not.

Table 2: Optimum coding methods for extreme input signal characteristics [7].

Input Signal		**Optimum Coding**	
Time Domain	**Freq. Domain**	**Direct Coding**	**Predictive Coding**
		Coding of spectral data	Prediction in time domain
		Coding of time domain data	Prediction in frequency domain

The second principle is that the TNS filter shapes quantization noise with open-loop predictive coding [4] [7]. Figure 7 indicates the flowchart of the open-loop predictive coding, where $x[k]$ is the input of the frequency domain signal in the analysis, $y[k]$ is the output of the frequency domain signal in the synthesis filter and the open-loop predictive coding adopts the previous signal to predict the present signal. The reconstruction error is defined as $r[k]$.

$$r[k] = x[k] - y[k] \tag{1}$$

From (1) and the Figure 7, *r[k]* can be derived as

$$\begin{aligned} r[k] &= x[k] - (\sum_{j=1}^{n} h_j \cdot y[k-j] + u[k]) \\ &= x[k] - \sum_{j=1}^{n} h_j \cdot y[k-j] - (d[k] + q[k]) \end{aligned} \tag{2}$$

where d[k] is the residual spectral lines of the predictive coding.

$$
\begin{aligned}
r[k] &= x[k]-\sum_{j=1}^{n} h_j \cdot y[k-j]-(x[k]-\sum_{j=1}^{n} h_j \cdot x[k-j]+q[k]) \\
&= \sum_{j=1}^{n} h_j \cdot r[k-j]+q[k] \qquad (3)
\end{aligned}
$$

Applying the z-transform to the spectral lines yields

$$R[z]=\frac{Q[z]}{1-\sum_{j=1}^{n} h_j z^{-j}}=\frac{Q[z]}{H[z]} \qquad (4)$$

From (4), the quantization error will be shaped in z-domain depending on the envelope of the inverse filter *H[z]*, which will be similar to the envelope of the time-domain input signal.

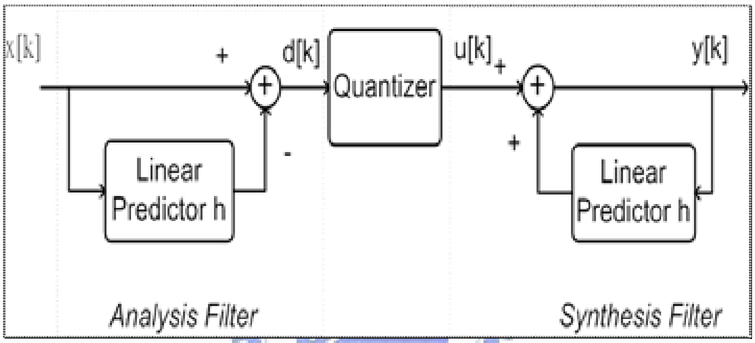


Figure 7: Open-loop predictive coding scheme.

2.3.2 **The implementation of TNS in AAC**

The TNS flowchart of AAC is illustrated in Figure 8. The input of the TNS module is “spectral coefficients for some frequency range” which is defined in Table 4. Then, based on the two principles of TNS, the open-loop predictive coding is applied. The current sample of x[k] can be predicted from the previous samples x[k - j]. The predicted value is given by

$$\mathrm{y[k]}=\sum_{j=1}^{\mathrm{N}} \mathrm{h}_{\mathrm{j}} \cdot x[k-j] \qquad (5)$$

where N is the number of the predictive order. The reconstruction error is *d[k]*.

$$d[k]=y[k]-x[k] \qquad (6)$$

$$=\sum_{j=1}^{\mathrm{N}} \mathrm{h}_{\mathrm{j}} \cdot x[k-j]-x[k] \qquad (7)$$

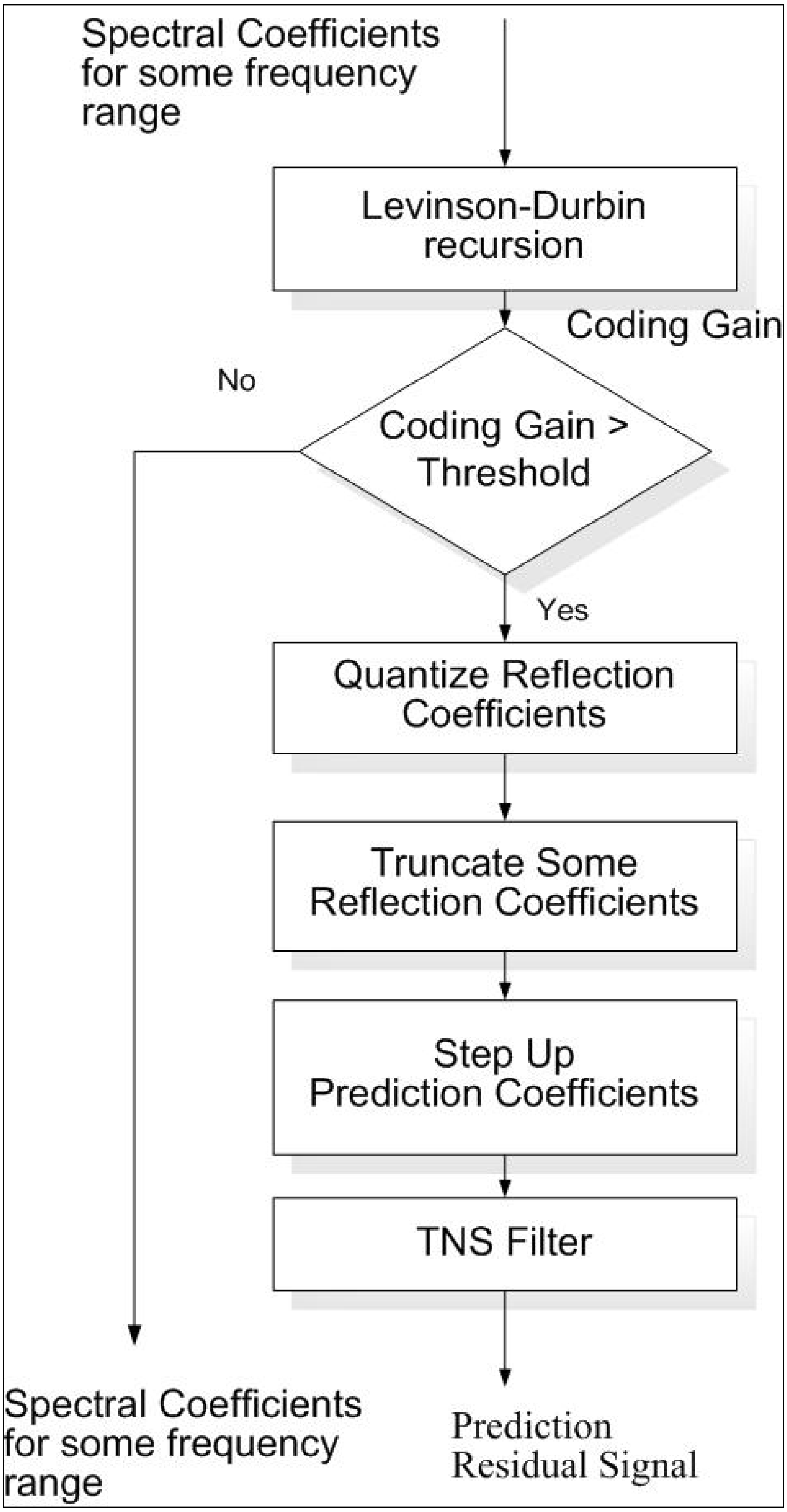

Spectral Coefficients for some frequency range
Levinson-Durbin recursion
Coding Gain
No
Coding Gain > Threshold
Yes
Quantize Reflection Coefficients
Truncate Some Reflection Coefficients
Step Up Prediction Coefficients
TNS Filter
Spectral Coefficients for some frequency range
Prediction Residual Signal

Figure 8: The TNS flowchart of AAC.

$$(d[k]^2) = (\sum_{j=1}^{N} h_j \cdot x[k-j] - x[k])^2 \tag{8}$$

In order to get the optimal predictor coefficients $\{h_j\}$, taking the derivative of the expected value of d[k] with respect to the coefficient $\{h_j\}$. The N equations can be derived.

$$\frac{\delta}{\delta h_j} E[(\sum_{i=1}^{N} h_i x[k-i] - x[k])^2] = 0$$

$$\Rightarrow 2E[(\sum_{i=1}^{N} h_i x[k-i] - x[k])] \cdot x[k-j] = 0$$

$$\Rightarrow \sum_{i=1}^{N} h_i \cdot E[x[k-i] \cdot x[k-j]] = E[x[k] \cdot x[k-j]] \tag{9}$$

The value of the $E[x[k] \cdot x[k-j]]$ is usually estimated by the autocorrelation method.

In the autocorrelation approach, we assume the {x[k]} sequence is stationary and $E[x[k-i] \cdot x[k-j]] = R_{yy}(|i-j|)$. If the i and j are bigger than k, we set x[k-i] and x[k-j] be zero. So

$$R_{yy}(m) = \sum_{q=1}^{N-m} x[q] \cdot x[q+m] \tag{10}$$

We get the matrix $RA = P$

$$R = \begin{bmatrix} R_{yy}(0) & R_{yy}(1) & \cdots & R_{yy}(N-2) & R_{yy}(N-1) \\ R_{yy}(1) & R_{yy}(0) & \cdots & R_{yy}(N-3) & R_{yy}(N-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{yy}(N-2) & R_{yy}(N-3) & \cdots & R_{yy}(0) & R_{yy}(1) \\ R_{yy}(N-1) & R_{yy}(N-2) & \cdots & R_{yy}(1) & R_{yy}(0) \end{bmatrix} \tag{11}$$

$$A = [h_1, h_2, \cdots, h_N]^T \qquad P = [R_{yy}(1), R_{yy}(2), \cdots, R_{yy}(N)]^T \tag{12}$$

$$\Rightarrow A = R^{-1}P \tag{13}$$

However, the R is a special form matrix called a Toeplitz matrix that obviates the need for computing R^{-1}. There are a lot of efficient algorithms to compute the inversion of Toeplitz matrices [13]. One of them is the Levinson-Durbin algorithm.

```
Step 1. Let $E_0 = R_{yy}(0)$ , $i = 0$
Step 2. $i = i + 1$
Step 3. $k_i = (\sum_{j=1}^{i-1} h_j \cdot R_{yy}(i - j + 1) - R_{yy}(i)) / E_{i-1}$
Step 4. $E_i = (1 - k_i^2) \cdot E_{i-1}$
Step 5. If $i < N$, go to Step 2
Step 6 Else return coding gain = $E_0 / E_i$
```

Algo 1: Levinson-Durbin algorithm.

The Levinson-Durbin algorithm will generate k_i coefficients known as the reflection coefficients and also calculate $E[r[k]^2]$. It denotes the average error using an *mth*-order filter by E_m.

The autocorrelation method using the Levinson-Durbin algorithm not only has the lower computation complexity but also provides a simple way to recursively adapt the prediction order. The more the prediction order is, the lower the variance of the residual error is and the higher the value of the coding gain defined in *Step 6*. When the coding gain is bigger than a threshold 1.4 in the AAC standard, the TNS module will be active. It means that the input signal is predicted much more easily and can be viewed as an attack signal.

Once the various reflection coefficients have been obtained, they need to be coded and transmitted in the encoder. To efficiently transmit the reflection coefficients, the module of the Quantize Reflection Coefficients is needed in Figure 8. Although reflection coefficients are less sensitive to quantization, they are very sensitive when the magnitude of them is close to unity. Therefore, an arcsine function is suitable for the reflection coefficients. The curve of the arcsine function is Figure 9.

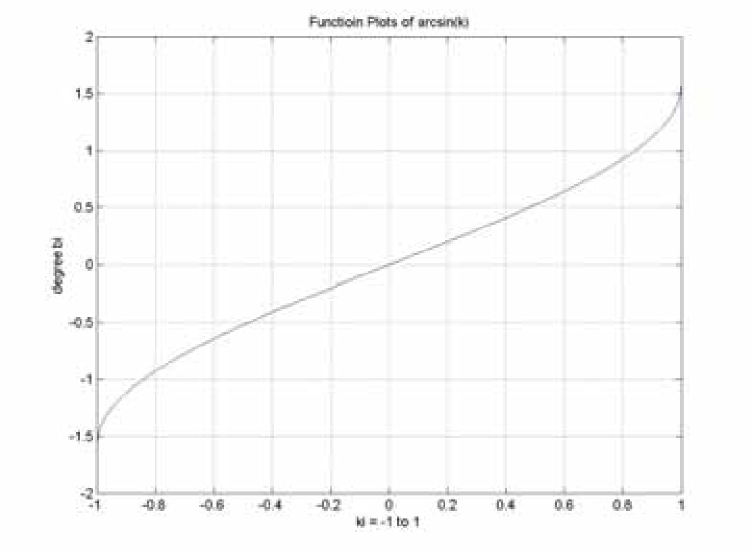


Figure 9: The curve of the arcsin.

To efficiently transmit the reflection coefficients, these reflection coefficients are represented as

$$b_j = \arcsin(k_j) \tag{14}$$

The value of each b_j restricted to $[-\pi/2, \pi/2]$ is quantized by (15)

$$index_j = NINT(b_j \cdot Q) \tag{15}$$

The Q is

$$Q = \begin{cases} ((1 << (\mathrm{r_{bit}} - 1)) - 0.5)/(\pi/2), \mathrm{k_j} >= 0 \\ ((1 << (\mathrm{r_{bit}} - 1)) + 0.5)/(\pi/2), \quad \mathrm{k_j} < 0 \end{cases} \quad \forall j \tag{16}$$

The r_{bit} means the number of the bits to represent the index$_j$, determined by the parameter “coefcompress$_{w,f}$.

$$\mathrm{r_{bit}} = \begin{cases} 4, \text{when coef\_compress}_{\mathrm{w,f}} = 0 \\ 3, \text{when coef\_compress}_{\mathrm{w,f}} = 1 \end{cases} \tag{17}$$

where the parameter “coef_compress” is defined in Table 3. The curve of (15) is illustrated in Figure 10. When the magnitude of the k_i is close to unity, the quantization error is smaller.

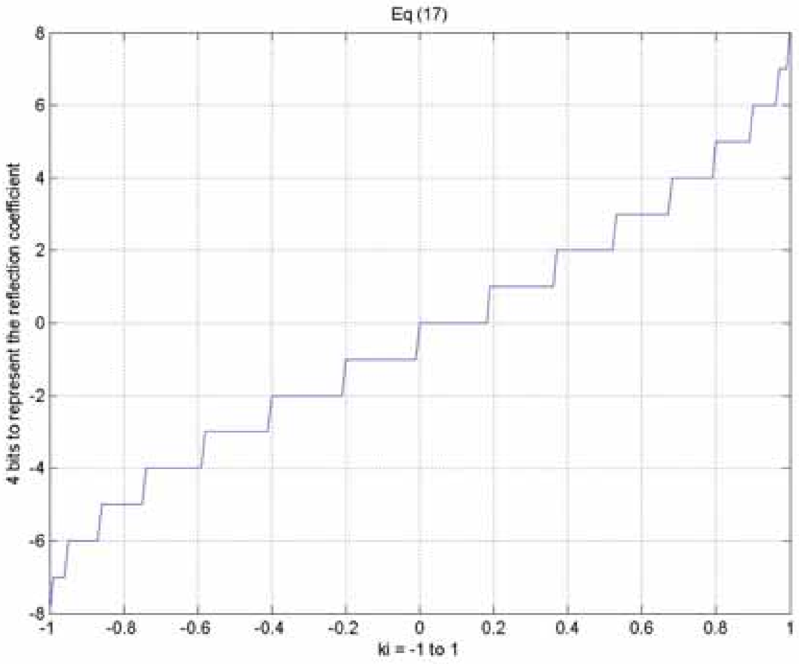


Figure 10: The curve of (15).

In decoder, the value of $k_j^{/}$ can be obtained through (18).

$$k_j^{/} = NINT(\sin(index_j / \mathrm{Q})) \tag{18}$$

The next module is “Truncate Some Reflection Coefficients”. In view of reducing bits, the order can be decreased by subsequently removing all coefficients with an absolute value smaller than a threshold from the “tail” of the reflection coefficients array. Finally, to get the optimal predictor coefficients $\{h_j^i\}$ that mean the coefficients of the *i-th* order filter, the module of “Set Up Prediction Coefficients” is an important step. The coefficients $\{h_j^i\}$ depends on the coefficients of the *(i-1)*-th order filter and $k_{i.}$

```
Step 1. $i = 0$
Step 2. $i = i + 1$
Step 3. Set $h_i^{(i)} = k_i$
Step 4. $h_j^i = h_j^{(i-1)} + k_i \cdot h_{i-j}^{i-1}$   for $j = 1, 2, \ldots, i-1$
Step 5. if i < order, go to Step 2
```

Algo 2: Set Up prediction coefficients algorithm.

Then, the module of “TNS Filter” can use the prediction coefficients to produce the “Prediction Residual Signal” by (19).

$$\mathrm{y[k]} = x[k] - \sum_{j=1}^{\mathrm{N}} \mathrm{h}_{\mathrm{j}} \cdot x[k-j] \tag{19}$$

2.3.3 **Side information of TNS**

The need of information for TNS is described in Table 3. It supports several filters performing on the distinct frequency range, of which the maximum value is 3. For each filter, a band is used to be a unit of the frequency bandwidth for applying the filter. According to the different profiles and windows, the value of the maximum order is not the same. For long windows, in the main profile, the maximum order is 20, which it needs at least 5 bits to represent. For the low complexity profile and the scaleable sampling rate profile, it is 12. However, for short windows, the value for the constant TNS_MAX_ORDER is 7 for all profiles, which 3 bits can represent. In order to gain more bits, it can sacrifice the resolution of the transmitted filter coefficients from 4 bits to 3 bits. Besides, when b is the number of bits to represent the coefficients and the values of all coefficients are between -2^{b-2} and 2^{b-2}-1, encoding the coefficients can just use b-1 bits. It provides not only the forward prediction but also the backward prediction.

Table 3: The side information of TNS.

Bitstream element	bits	Description
n_filt	2	Number of filter
length$_{f}$	6	The number of bands processed by the filter f
order$_{f}$	5/3	The maximum bits of the orders can be used for the filter f. 5 is used for the long window and 3 is for the short window
coef_compress$_{f}$	1	Indicating whether the most significant bit of the coefficients for the filter can be omitted or not.
coef_res	1	token indicating the resolution of the transmitted filter coefficients, switching between a resolution of 3 bits (0) and 4 bits (1)
direction$_{f}$	1	Backward or forward prediction
coef$_{f,i}$	2~4	Depend on the coef_compress$_{f}$ and coef_res

According to the sampling rate and profile in use, the value of the maximum order is set as Table 4. However, the minimum band is not limited.

Table 4: The TNS_MAX_BANDS for different profiles and sampling rates.

Sampling Rate [Hz]	Low Complexity / Main Profile (long windows)	Low Complexity / Main Profile (short windows)	Scaleable Sampling Rate Profile (long windows)	Scaleable Sampling Rate Profile (short windows)
96000	31	9	28	7
88200	31	9	28	7
64000	34	10	27	7
48000	40	14	26	6
44100	42	14	26	6
32000	51	14	26	6
24000	46	14	29	7
22050	46	14	29	7
16000	42	14	23	8
12000	42	14	23	8
11025	42	14	23	8
8000	39	14	19	7

Chapter 3 TNS Artifacts

As mentioned in Chapter 2, the quantization errors will be shaped in time domain if the spectral lines obtained are through a kind of discrete four transform. However, the time-frequency transform in AAC is the filterbank instead of the Fourier transform. This difference of the filterbank from the Hilbert transform leads to some perceptual artifacts when the temporal noise shaping is applied. In [7], the time-domain aliasing is mentioned. However, according to the properties of the filterbank, the chapter classifies the time-domain aliasing into three types. In next chapter, we will introduce the schemes to handle the artifacts.

3.1 Modified Discrete Cosine Transform (MDCT)

Modified Discrete Cosine Transform is a kind of tool used in AAC mapping the time-domain signal into the frequency-domain and also a TDAC (Time Domain Aliasing Concealing) transform with PR property.

The direct MDCT and inverse modified discrete cosine transform (IMDCT) are defined as

$$x[k] = \sum_{n=0}^{2N-1} h_n x[n] \cdot \cos\left(\frac{(n+(N+1)/2)(k+0.5)}{N}\pi\right) \quad for\ k = 0 \cdots N-1 \tag{20}$$

and

$$\hat{x}[n] = h_n \cdot \frac{2}{N} \sum_{k=0}^{N-1} x[k] \cdot \cos\left(\frac{(n+(N+1)/2)(k+0.5)}{N}\pi\right) \quad for\ n = 0 \cdots 2N-1 \tag{21}$$

where $x[n]$ is the time domain input signal of $2N$ samples and h_n is a window function satisfying the constraints of perfect reconstruction:

$$h_n = h_{2N-1-n} \tag{22}$$

and

$$h_n^2 + h_{n+N}^2 = 1 \tag{23}$$

For example, the sine window is widely used in most audio coding with coefficients

$$h_n = \sin\left(\pi \frac{k+1/2}{2N}\right) \quad for\ k = 0, \ldots, 2N-1 \tag{24}$$

For convenience, (20) and (21) can be denoted as the matrix representation.

$$\hat{x}_{2N} = \left[h_n\right]_{2N\times 2N} \left[imdct\right]_{2N\times N} \left[mdct\right]_{N\times 2N} \left[h_n\right]_{2N\times 2N} \vec{x}_{2N} \tag{25}$$

$$\hat{x}_{2N} = H \cdot M^T \cdot M \cdot H \cdot \vec{x}_{2N} \tag{26}$$

where $\vec{x}_{2N}$ is the input sample, $\hat{x}_{2N}$ is the output sample , H is $[h_n]_{2N\times 2N}$, M^T is $\left[imdct\right]_{2N\times N}$ and M is $[mdct]_{N\times 2N}$. The property of $M^T \cdot M$ is

$$M^T M = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}_{2N*2N} \tag{27}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}_{N*N} \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{N*N} \tag{28}$$

Therefore, based on (28), the property of MDCT is that the input signal can not be reconstructed from one single block of MDCT coefficients. In the overlap-add process, the aliasing is cancelled in two consecutive blocks to achieve the perfect reconstruction through the property:

$$A + B = 2I \tag{29}$$

However, the relationship between MDCT and DFT can be established via Shifted Discrete Fourier Transforms (SDFT) in [14]. The direct SDFT is defined as

$$SDFT_{u,v}(x[n]) = x[k]^{u,v} = \sum_{k=0}^{2N-1} x[n] \cdot \exp[i2\pi \frac{(k+u)(r+v)}{2N}] \quad . \tag{30}$$

and inverse SDFT is

$$ISDFT_{u,v}(x[k]^{u,v}) = x[n]^{u,v} = \frac{1}{2N} \sum_{r=0}^{2N-1} x[k]^{u,v} \cdot \exp[-i2\pi \frac{(k+u)(r+v)}{2N}] \quad . \tag{31}$$

where u and v represent arbitrarily the shifts in the time and frequency domain, respectively. DFT is the most widely known special case of which (for zero shifts u and v).

$$x[k]^{u,v} = \sum_{k=0}^{2N-1} x[n] \cdot \exp[i2\pi \frac{kr}{2N}] \tag{32}$$

It has been proven that MDCT is equivalent to the SDFT of a modified input signal [15] [16].

$$x[k] = \frac{1}{2} \sum_{k=0}^{2N-1} \hat{x}[n] \cdot \exp[i\pi \frac{(k+(N+1)/2)(r+1/2)}{N}] \tag{33}$$

For real-valued signals, the MDCT coefficients are proven to be equal to the real part of SDFT$_{(N+1)/2,1/2}$ of the input signal.

$$x[k] = real\{SDFT_{(N+1)/2,1/2}(\hat{x}[n])\} \tag{34}$$

And $\hat{x}[n]$ is defined as

$$\hat{x}[n] = \begin{cases} h_k \cdot x[k] - h_{N-1-k} \cdot x[N-1-k] & ,k = 0,..., \quad N\text{ -}1 \\ h_k \cdot x[k] + h_{3N-1-k} \cdot x[3N-1-k] & ,k = N,...,2N-1 \end{cases} \tag{35}$$

The right side of (33) is SDFT$_{(N+1)/2,1/2}$ that can be expressed by means of the conventional DFT as

$$\sum_{k=0}^{2N-1} \hat{x}[n] \cdot \exp[i\pi \frac{(k+(N+1)/2)(r+1/2)}{N}]$$

$$= \left\{ \sum_{k=0}^{2N-1} \left[\hat{x}[n] \cdot \exp(i2\pi \frac{k}{4N}) \right] \exp(i2\pi \frac{kr}{2N}) \right\} \exp(i2\pi \frac{(N+1)r}{4N}) \exp(i\pi \frac{N+1}{4N}) \tag{36}$$

The conclusion is that SDFT$_{(N+1)/2,1/2}$ is the conventional DFT of this signal shifted in the time domain by (N+1)/2 of the sampling interval and evaluated with the shift of 1/2 the frequency-sampling interval. Figure 6 illustrate the simplified flowchart of AAC encoder without the psychoacoustic module. From (34), the MDCT operation can be replaced by the “Shaper” and “SDFT”. The operation of the “Shaper” is the same as the Eq (35). In the decoder, the ISDFT is substituted for the IMDCT. Figure 11 shows the flowchart equivalent to the simplified encoder in Figure 6.

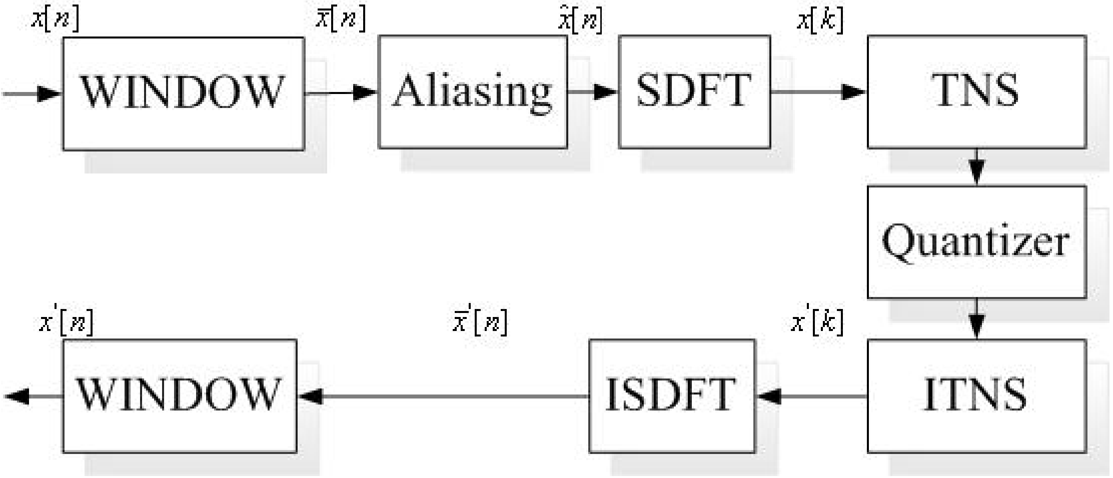


Figure 11: The codec equivalent to the Figure 6.

3.2 Noise Amplification around Attack

Although the filterbank MDCT instead of the Discrete-Time Fourier Transform is adopted in AAC, the mapping between the z-domain and the time domain is similar. By (4), the quantization error will be filtered with the inverse filter *H[z]* which is an all-pole filter. However, when the pole is far away from the unit circle, the width of the resonance point will be wider. Like Figure 12, it's an single pole filter $H[z]=\dfrac{1}{(z-re^{j\vartheta})}$, where ϑ is setting as π and the r is 1, 0.9 and 0.7. Obviously, the width of the resonance point is affected by the length of "r". Therefore, the quantization error is spread out before the attack signal by the inverse filter *H[z]*.

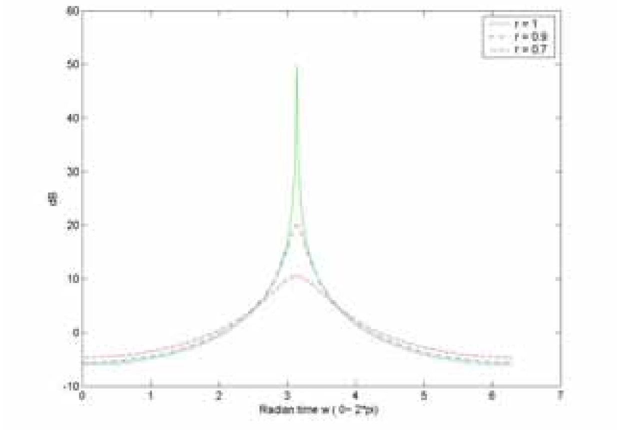
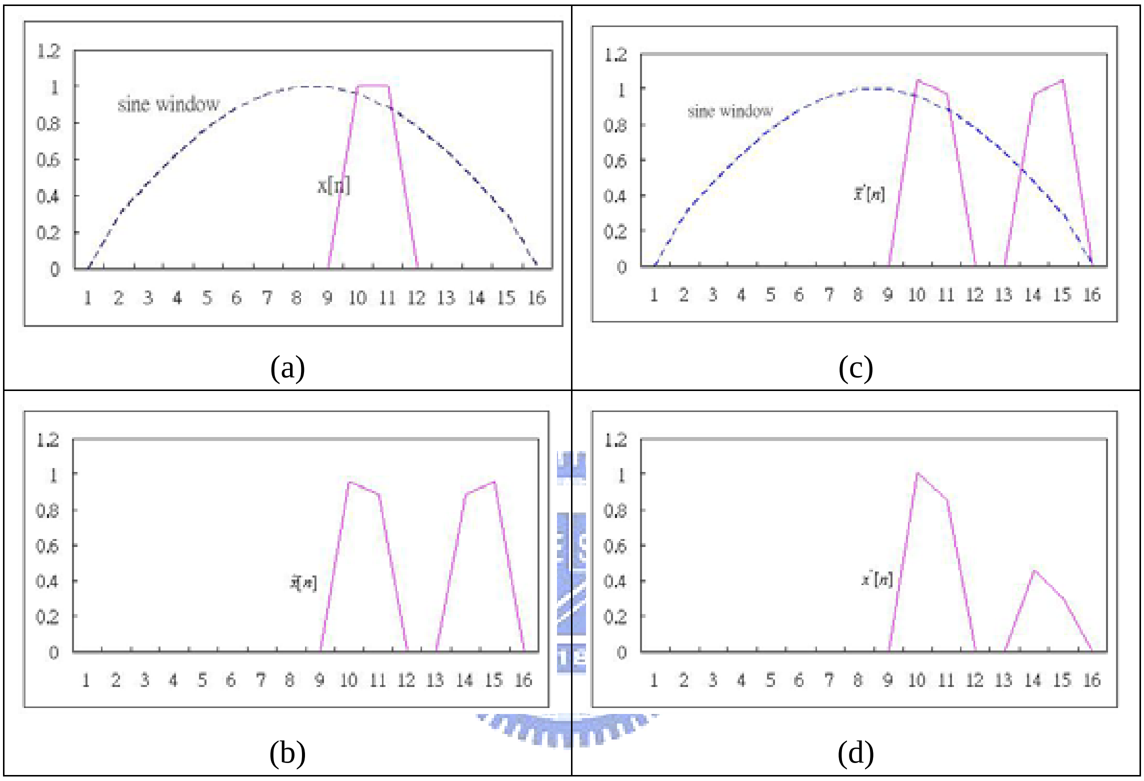


Figure 12: Time response for a single pole, with $\theta=\pi$, r=1, 0.9, 0.7.

Figure 14 illustrates original signal, coded signal without the TNS and with TNS. Figure 15 illustrates the noise signal which is the difference between the original signals and the decoded signals without TNS and with TNS. We can find that the noise around the attacking time interval is amplified after the TNS is applied although the pre-echo is reduced in general. The noise may not be very sensitive to the human auditory system if the noise is controlled to be localized around the attacking time due to the pre-masking effect.

3.3 Time-Domain Aliasing

In Section 2.3.1 , the ideal envelope of inverse filter H[z] should be similar to the envelope of the time-domain input signal $x[n]$. However, in Figure 11, the MDCT operation can be replaced by the "Shaper" and "SDFT". The aliasing is added to the original signal by the operation of the "Shaper". For the 'SDFT", the input signal is $\hat{x}[n]$ rather than $\bar{x}[n]$. Therefore, the envelope of inverse filter H[z] will be similar to the envelope of $\hat{x}[n]$ not $\bar{x}[n]$. Therefore, the quantization noise will be amplified by the non-ideal inverse filter H[z] and the aliasing occurs at the annoying position. In order to illustrate the whole process, in an intuitive way, we have employed an artificial time domain signal $x[n]$ (N=8) as shown in Figure 13 (a) and the wild-used window in AAC is the sine window. The result of the shaper module is $\hat{x}[n]$ in Figure 13 (b). So, the envelope of inverse filter H[z] will be similar to Figure 13 (b). It causes the quantization error will be amplified at the 10th, 11th, 14th and 15th points in Figure 13 (c). After the multiplication of the sine window, post-aliasing artifact will occur in Figure 13 (d). And the pre-aliasing artifact can be derived in the same way. The real examples are in Figure 16 and Figure 17.



(a)

(c)

(b)

(d)

Figure 13: (a) the input signal x[n] and the sine window (b) the output of the shaper module $\hat{x}[n]$ (c) assume the output of SDFT^{-1} is $\bar{x}^{'}[n]$ (d) the finally output behind the sine window in Figure 11.

3.4 Noise from High-Order Prediction Filter

In general, the coding gain increases with the order of the prediction filter. Hence, the quantization noise may be considered to shape better with the increase of filter order. It means that the noise will be more concentrated on the signal with high energy. Besides, the pre-aliasing and post-aliasing artifact will be more apparent Figure 18 illustrates from the TNS with order 3 and order 12.

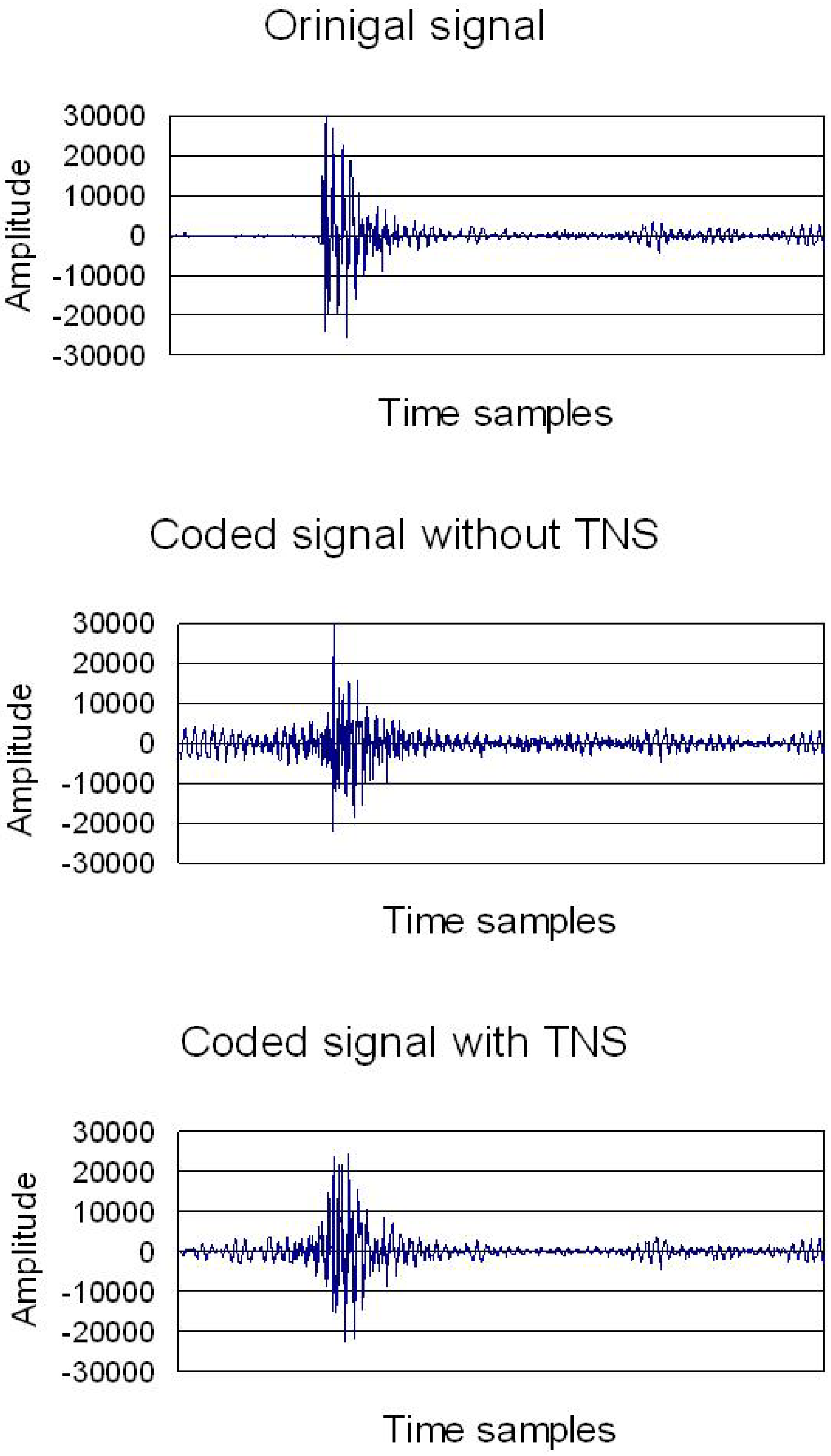


Figure 14: The top figure is the original signal. The middle figure is the signal coded without TNS. The bottom is the signal with TNS. The top is original signal. The bottom is coded with TNS.

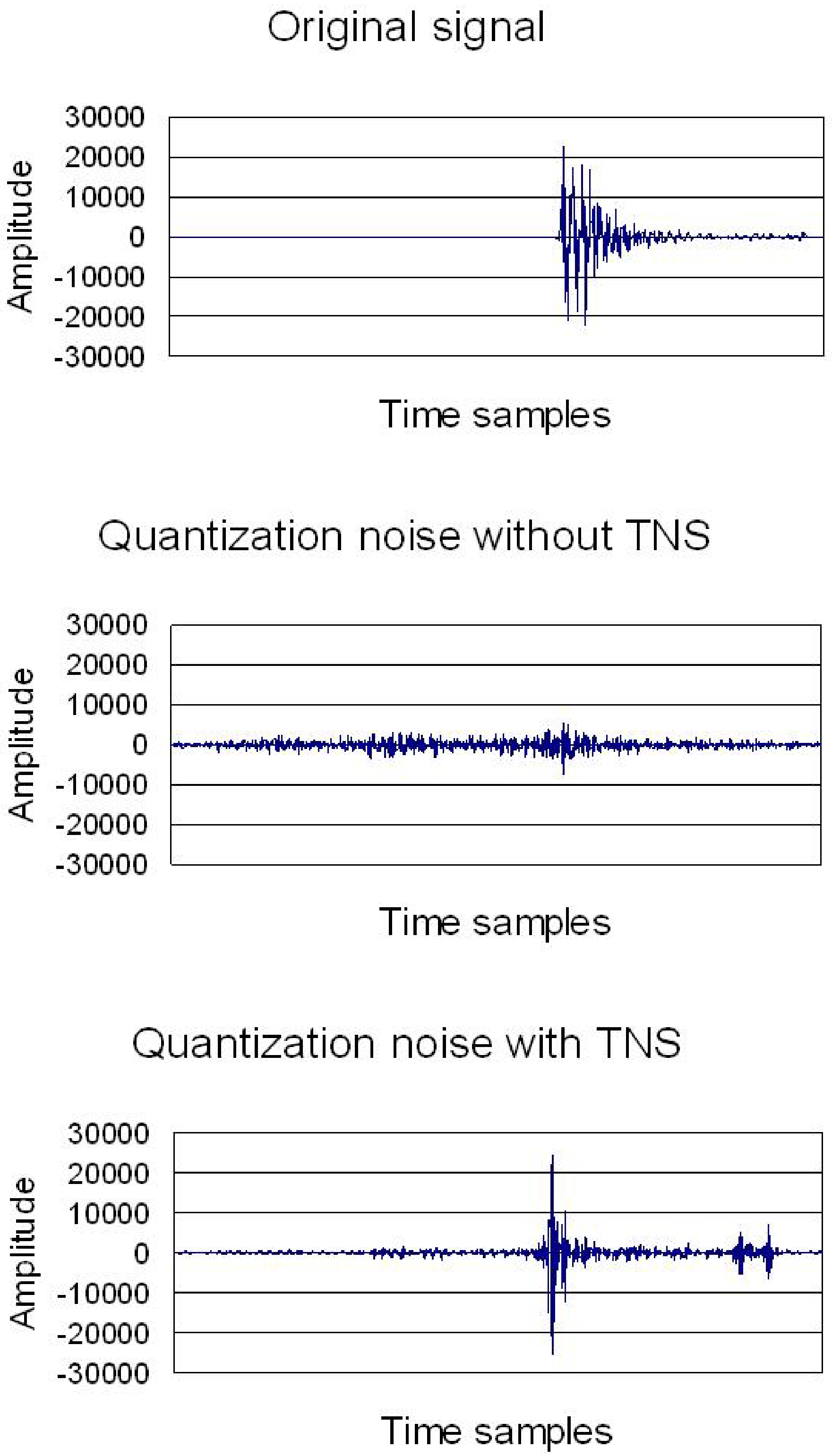


Figure 15: The noise around the attacking time interval is amplified after the TNS is applied although the pre-echo is reduced in general.

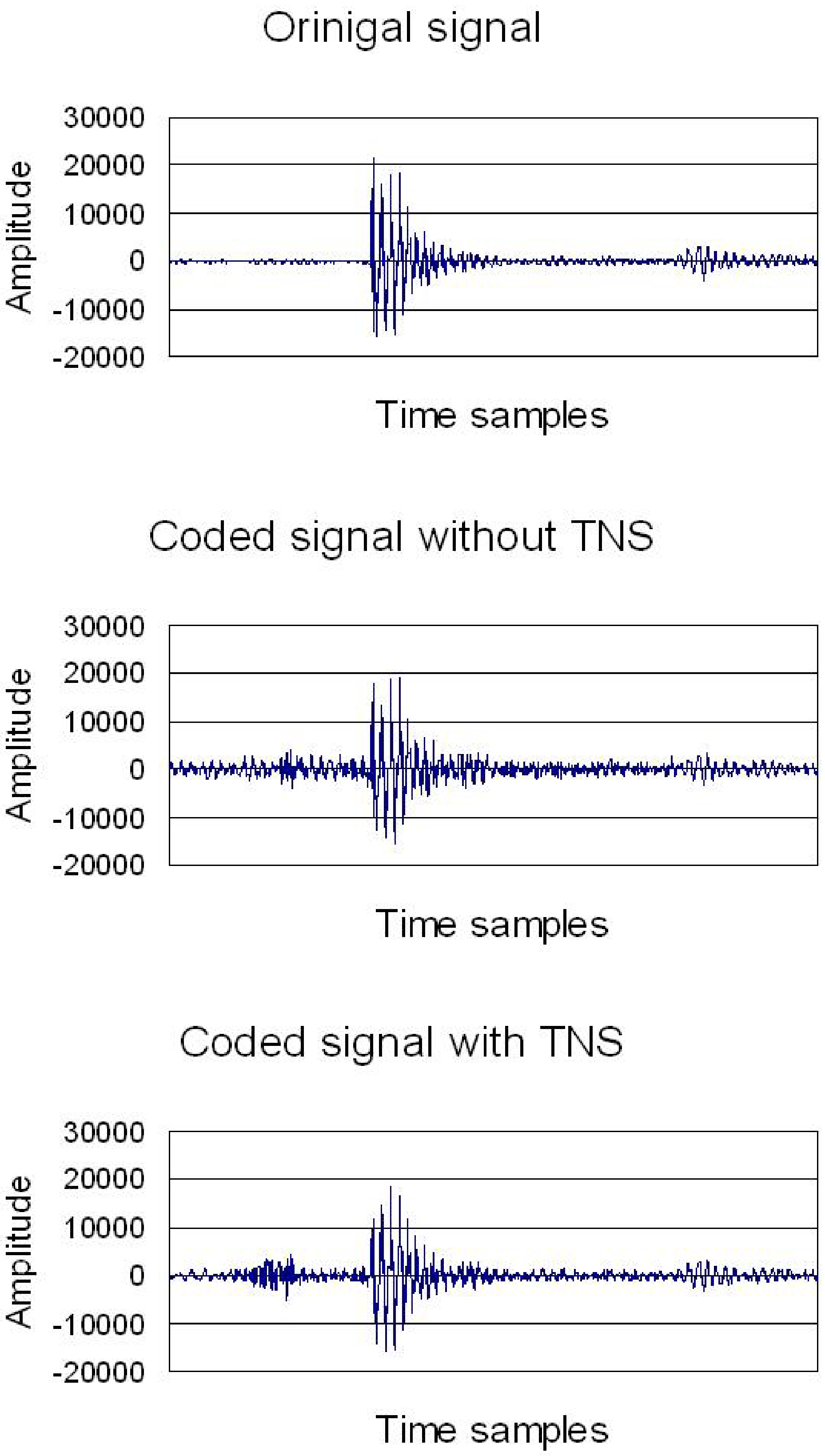


Figure 16: The pre-aliasing artifact emerges before the attack signal (there are 1024 points and the size of the window is 2048).

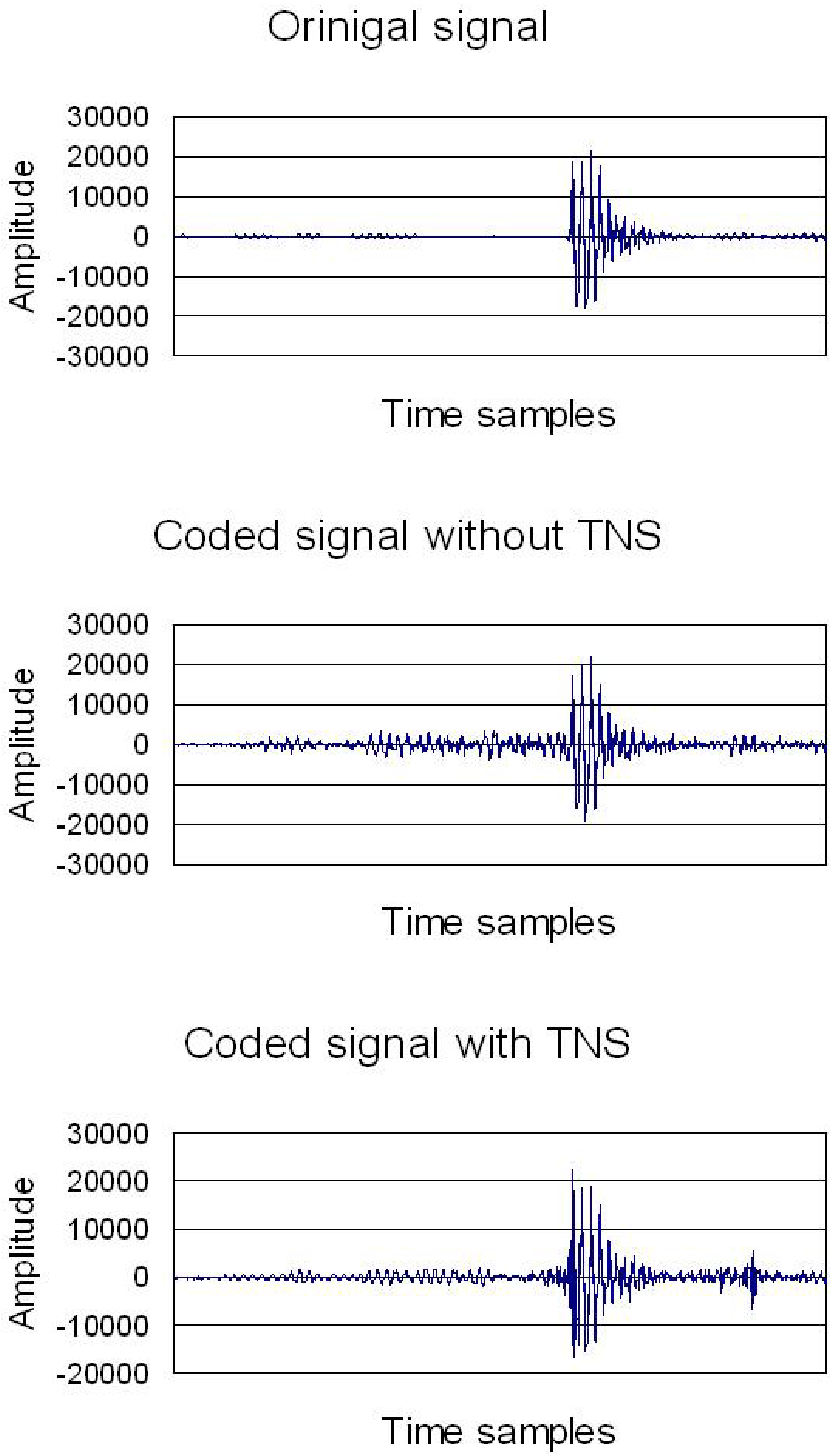


Figure 17: The post-aliasing artifact emerges behind the attack signal (there are 1024 points and the size of the window is 2048).

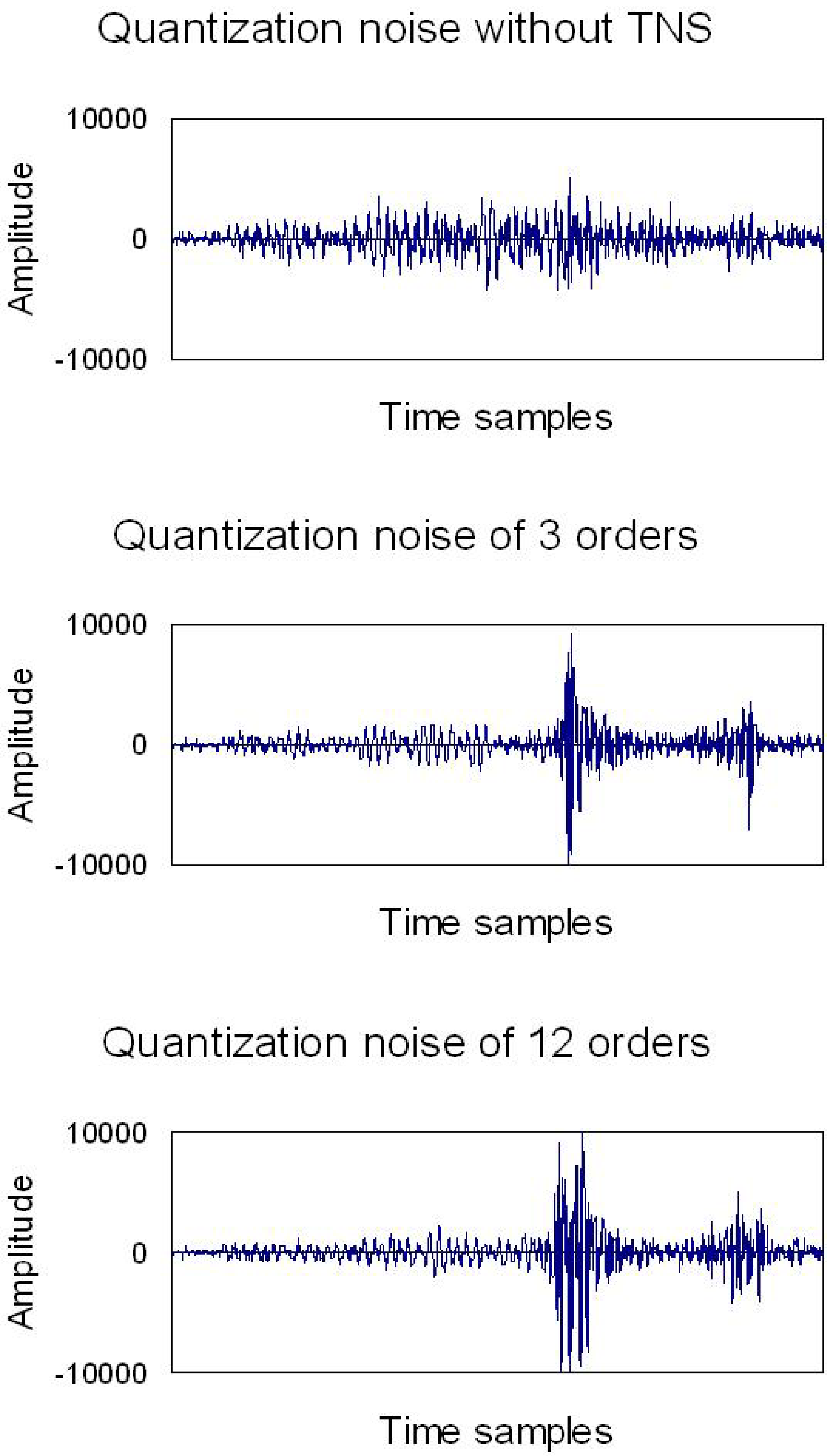


Figure 18: The effect from the different filter order.

Chapter 4 The Efficient Temporal Noise Method

There are two problems associated with the detection mechanism. First, as illustrated in last section, the coding gain can not reflect the injection of the above three artifacts. Also, the switch mechanism based on the coding gain directly leads to computing overhead from the TNS filtering. This chapter presents a detection mechanism based on the perceptual entropies. Also, we propose the methods to handle the three artifacts. The method can leads to merits in both quality and complexity.

4.1 The Perceptual Entropy Switch Method

In order to resolve these disadvantages mentioned above, the efficient switch criterion through PE (Perceptual Entropy) is proposed in [17].
The PE is defined as:

$$PE = \sum_{b} PE_b = \sum_{b} BW_b * \log\left(\frac{E_b + 1}{Masking_b}\right) \qquad (37)$$

where *b* is the index of the threshold calculation partition, BW_b is the number of the frequency lines in partition b, E_b is the sum of the energy in partition *b* and $Masking_b$ is the masking threshold in partition *b*. The masking threshold $Masking_b$ is defined as

$$Masking_b = \max(qthr_b, \min(nb_b, nb\_l_b * repelev)) \qquad (38)$$

where $qthr_b$ is the threshold in quiet, nb_b is the masking threshold of partition *b*, nb_l_b is the threshold of partition *b* for the last block and *rpelev* is set to '1' for short blocks and '2' for long blocks. From (37) and (38), when the $(N\text{-}1)^{th}$ signal is like quiet sound and the N^{th} signal is an attack signal, the $Masking_b$ of the N^{th} signal is the small value $nb\_l_b * repelev$, not nb_b. Take an example, in Figure 20, the 1st frame is a quiet sound and the 2nd frame is an attack signal. For the calculation of the 2nd PE, the nb_l_b is much smaller than the nb_b and the corresponding PE is high. It means that the N^{th} input signal is an attack signal. However, the PE just detects the signal leading to the pre-echo phenomenon. In order to ease the post-echo phenomenon, the PE_b will be useful. When two consecutive frames have different contents , like an attack signal and a quiet sound, the masking value for each band should be different. Therefore, once the previous PE_b is much bigger than the current PE_b, it means that the post-echo phenomenon will be happened. So, except the low frequency band, if one of the values that the previous PE_b divides the current PE_b is over a threshold, the signal should be applied with the TNS module. Besides, the PE value of each frame has been computed in the psychoacoustic model. To avoid computing the Levinson-Durbin method for each frame, an attack flag decided through the information of the PE and PE_b in the psychoacoustic model is sent to the TNS module. Figure 19 illustrates the new flowchart of the TNS. Compared to Figure 8, the decision block “whether is coding gain bigger than the threshold” is replaced with the block “whether is the attack flag true”. If the flag is true, the Levinson-Durbin recursion will be computed. Obviously, the computation complexity is reduced a lot. However, by this way, if two attack signals appear in the two continues frame, the PE value of the second attack signal is not high enough and the signal is viewed as non-attack signal by the efficient switch method.

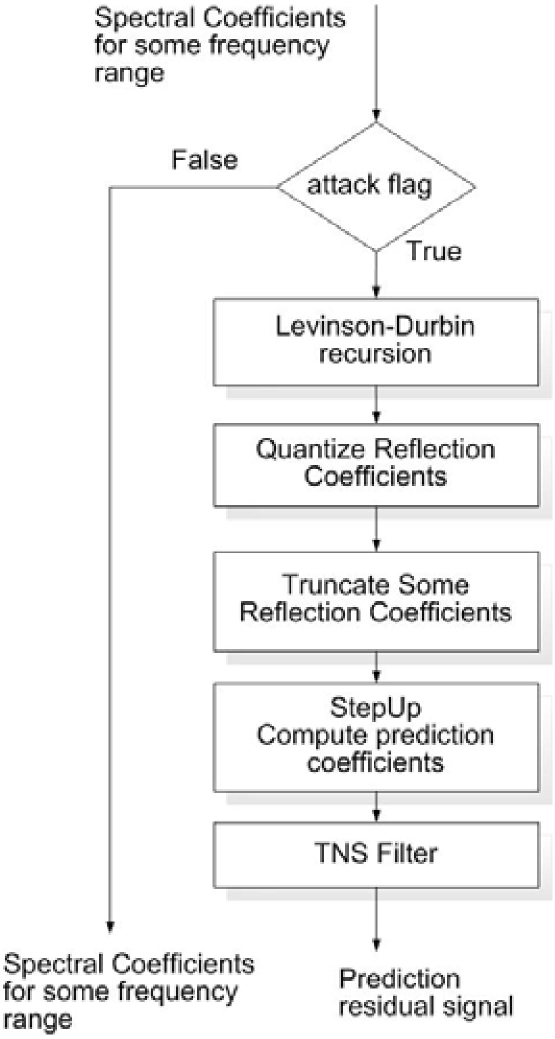

Spectral Coefficients for some frequency range
False
attack flag
True
Levinson-Durbin recursion
Quantize Reflection Coefficients
Truncate Some Reflection Coefficients
StepUp Compute prediction coefficients
TNS Filter
Spectral Coefficients for some frequency range
Prediction residual signal

Figure 19: The TNS flowchart with The PE method.

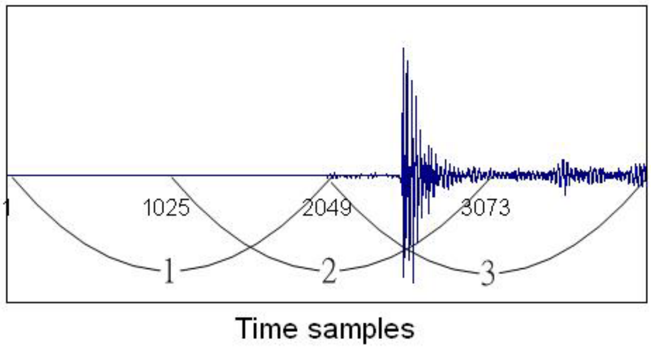


Figure 20: An attack signal appears in two frames.

Since the overlapping property of MDCT windows, the attack signal will appear in two consecutive blocks as Figure 20. Both the 2nd and 3rd frames should be applied with the TNS module. Thus, to ensure the two consecutive blocks active with the TNS module, if the previous frame is detected as an attack signal, the current frame is applied with the TNS module.

4.2 Ease pre-aliasing and post-aliasing artifact

In Chapter 3, the reason of pre-aliasing and post-aliasing artifact has been discussed in detail. The more order is, the more apparent the aliasing artifact is. It will lead to the bad performance of the TNS module. Obviously, to solve the problem, from Figure 13 (c), if the values at the tail of the window are zero, after the multiplication, the post-aliasing at the 14th and 15th point will be disappeared in Figure 13 (d). Similarly, if the values in the front of the window are zero, the pre-aliasing artifact can be eased. However, the LONG_START and LONG_STOP window defined in AAC are suitable for the above requirement. Then, based on the above PE switch method, an improved method to ease the artifact is proposed. First, in order to ease the artifact, the most important thing is to identify the position of the attack signal detected by the above PE method. According to the position, TNS can choose a suitable window for a better coding. Therefore, the Algo 3 is designed to detect the position, which classifies a long window block into eight zones and the energy of each zone is calculated. Starting from the zone 2, if one energy ratio over a threshold which is the energy of the current zone divide the energy of the previous zone is found, the zone is call as the position of the attack signal. In Figure 21, for the 2nd frame, the energy ratio for the zone 7 is over the threshold. So, the attack position is viewed as the zone 7. After detecting the position of the attack signal, the next step is to determine the suitable window to ease the aliasing artifact. If the position is between zone 5 and zone 8, the window of the current frame is set to the

LONG_START window and the next frame becomes the LONG_STOP window. In Figure 21, because of the attack position regarded as zone 7, the window of the 2nd frame is set to the LONG_START window and the next frame is the LONG_STOP window. Otherwise, if the attack signal locates at the frame between the zone 1 and 4, it should be the LONG_START window and the previous frame is the LONG_STOP window, the disadvantage of which is that the additional frame delay is needed. Therefore, for the efficiency, it retains the ONLY_LONG window for the attack position between zone 1 and 4. Finally, whether TNS is active or not, it depends on the attack flag, the window type and the attack position. It can be analysed as three conditions. One condition is that, if the window type is the LONG_START window and the attack position is at zone 5 and 6, TNS is active. But, if the attack position is at zone 7 and 8, it means that the current window doesn't contain the attack signal. For the next window, the attack position will be at zone 3 and 4. Therefore, the other condition is that if the window type is belong to the "LONG_STOP" window and the attack position is at zone 3 and 4, TNS is also active. Besides, the third condition is applying the TNS to the signal which the attack position is between zone 1 and 4. To reduce the pre-aliasing and post-aliasing, this condition should use less prediction order to shape the time domain noise.

```
Step 1. If attack flag is false, leave the algorithm
Step 2. Divide a frame into 8 zones
Step 3. Calculate the energy for each zone
Step 4. return the first position i such that energy[i]/energy[i-1]
        >TNS_SWITCH_RATIO, if exist
```

Algo 3: Detect Position algorithm.

```
Step 1. If the attack position is belong to the right half of the frame (i = 5,6,7,8) and
        the block type of the previous frame is ONLY_LONG, the block type is set
        as LONG_START
Step 2. Else if the block type of the previous frame is LONG_START, the block type
        is LONG_STOP.
Step 3. Else the block type is ONLY_LONG
```

Algo 4: Window switch algorithm.

Condition 1. the block type is LONG_START and the attack position is 5 or 6
Condition 2. the block type is LONG_STOP and the attack position is 3 or 4
Condition 3. (i)the block type is ONLY_LONG
(ii)the attack flag is true
(iii)the attack position is 1~4
Step 1. if one of the above 3 conditions is satisfied, the TNS module is active.
Step 2. if condition3 is satisfied, the prediction order should be less

Algo 5: TNS applied algorithm.

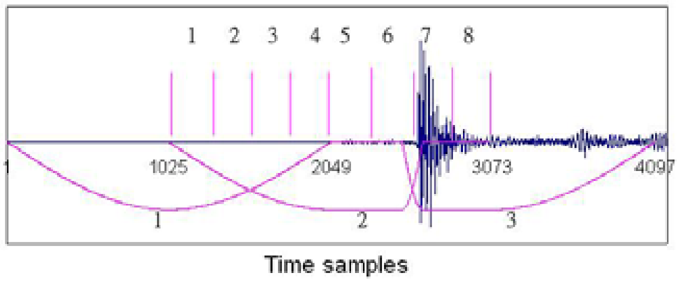


Figure 21: The position of an attack signal is at zone 7 for the 2rd frame.

Chapter 5 Experiments

5.1 Experiment Environment

Here, we have adopted for objective quality measure the PEAQ (perceptual evaluation of audio quality) which is the recommendation system by ITU-R Task Group 10/4. From the objective measurement method, the objective difference grade (ODG) is the output variable. The ODG values should range from 0 to - 4, where 0 corresponds to an imperceptible impairment and -4 to impairment judged as very annoying. The PEAQ has been widely used to measure the compression technique due to the capability to detect perceptual difference sensible by human hearing systems. The following experiments are based on the system [18] and the NCTU-AAC 1.0 codec which is belong to the Low-complexity profile, an implementation of AAC codec [19]. The 12 tracks used are listed in Table 5.

5.2 The Experiment of the Perceptual Entropy Switch Method

With a lot of experiments, the optimal threshold for the PE is found. In order to improve the PE method, we also consider the condition that the attack signal will appear in two consecutive blocks. NCTU-AAC 1.0 without TNS, NCTU-AAC 1.0 with TNS based on the PE method and NCTU-AAC 1.0 with TNS based on the PE and two consecutive block method are adopted for comparison in Figure 22. The two different TNS switch methods have a great improvement on the attack audio es01, es03, si02 and sm03 for both objective and subjective tests. However, the TNS based on the PE and two consecutive block method has a quality better than the TNS based on the PE method.

Figure 23 adopts NCTU-AAC 1.0 without TNS, NCTU-AAC 1.0 with TNS based on the coding gain method and NCTU-AAC 1.0 with TNS based on the PE and two consecutive block method for comparison. The max order for each experiment is 12. The TNS based on the coding gain method still has a better quality than the TNS based on the PE and two consecutive block method, especially in the es03. But, in the sm02, the TNS based on the coding gain method has a poorer quality than NCTU AAC without TNS.

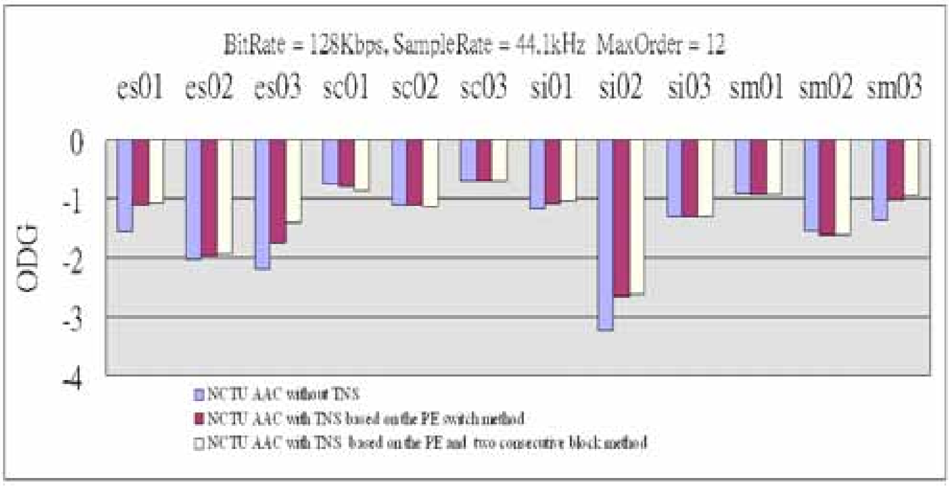


Figure 22: Objective test on the three methods: “NCTU-AAC 1.0 without TNS”, “NCTU-AAC 1.0 with TNS based on the PE method” and “NCTU-AAC 1.0 with TNS based on the PE and two consecutive block method”.

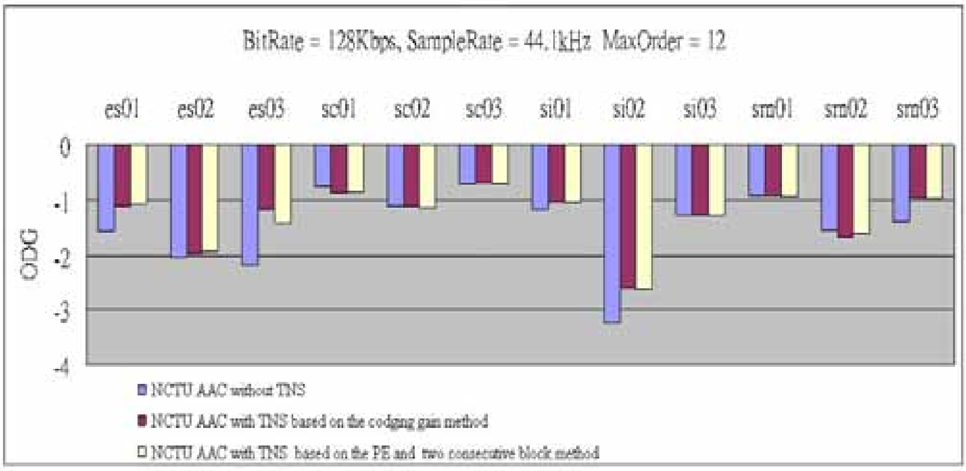


Figure 23: Objective test on the three methods: “NCTU-AAC 1.0 without TNS”, “NCTU-AAC 1.0 with TNS based on the coding gain method” and “NCTU-AAC 1.0 with TNS based on the PE and two consecutive block method”.

However, the more the filter order, the more apparent the pre-aliasing and post-aliasing artifact are. Figure 24 and Figure 25 illustrate the phenomenon. In order to show how the aliasing affects on the audio quality, the next two experiments limit the number of the MaxOrder to decrease the influence of the artifact. Because of the Low-complexity profile, the maximum MaxOrder is 12. Although the switch methods are different, with the order reduced, the quality improves. For convenience, “the efficient switch method” represents “the PE and two consecutive block method”.

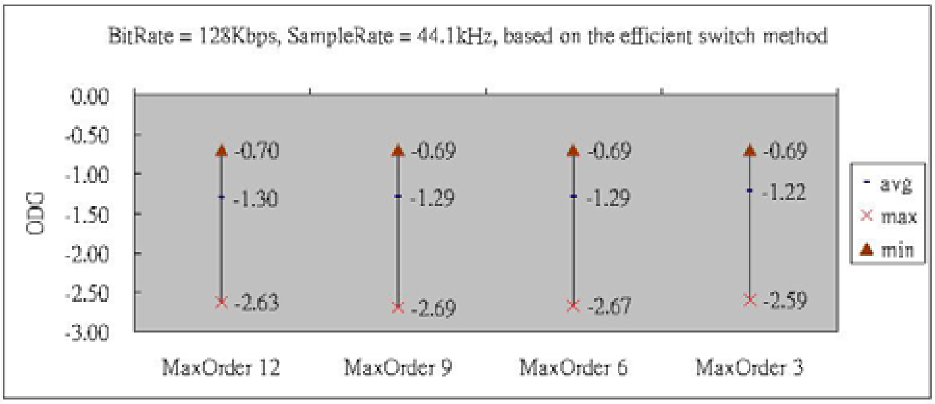


Figure 24: ODG for different MaxOrder based on the efficient switch method. The horizontal line is the average ODG among all the tested tracks in Table 5. The best ODG and the worst ODG in the tested tracks are marked with the triangle and “X” around the horizontal line.

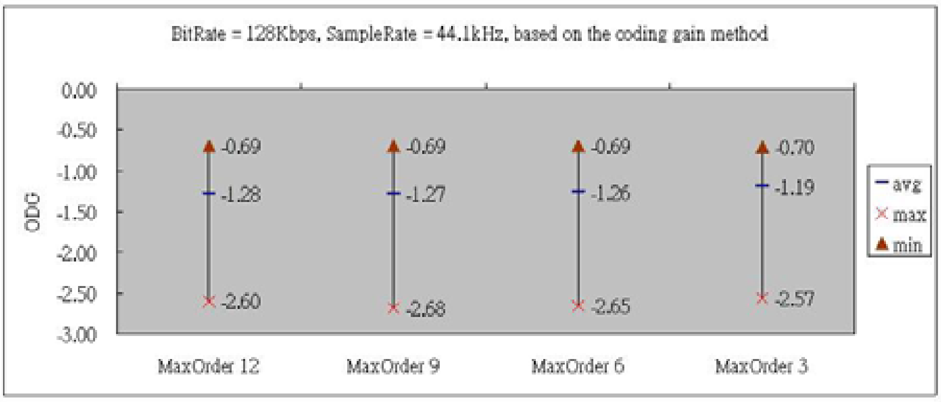


Figure 25: ODG for different MaxOrder based on the coding gain method.

From the experiments of Figure 24 and Figure 25, the optimal MaxOrder is 3. For each MaxOrder, the coding gain method has a quality better than the efficient switch method. In detail, comparing the two methods is in Figure 26. NCTU-AAC 1.0 without TNS, NCTU-AAC 1.0 with TNS based on the coding gain method and NCTU-AAC 1.0 with TNS based on the efficient switch method are adopted for comparison. The major difference is in the es02 and es03. It points out the disadvantage of the efficient switch method which can’t detect the attack signal, the distance between the previous attack-like signal and which is too close.

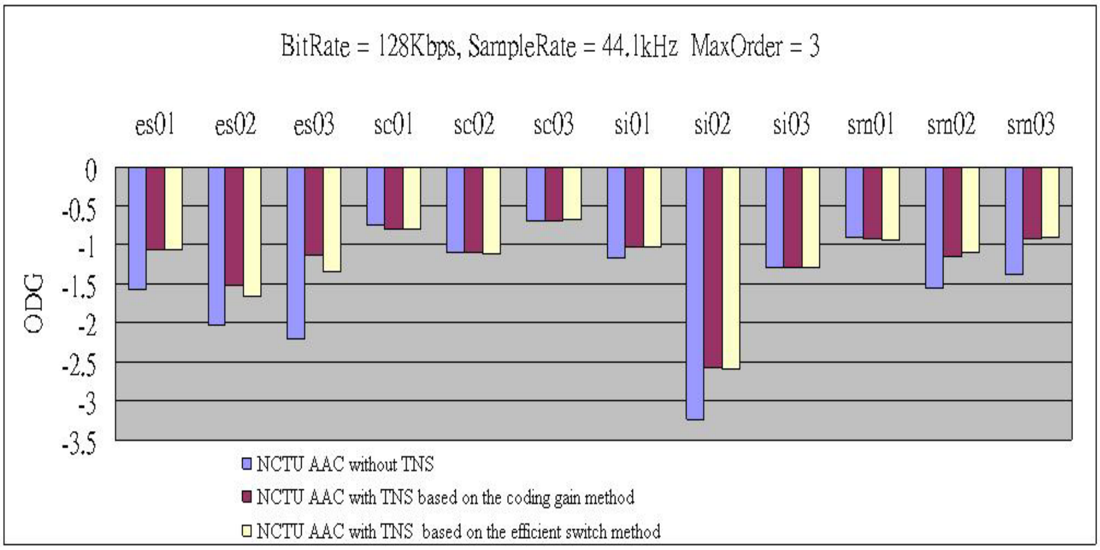


Figure 26: Objective test on the three methods: “NCTU-AAC 1.0 without TNS”, “NCTU-AAC 1.0 with TNS based on the coding gain method” and “NCTU-AAC 1.0 with TNS based on the efficient switch method”.

5.3 The Experiment for easing pre-aliasing and post-aliasing artifact

From the theory and the experiments of Figure 25 and Figure 24, the pre-aliasing and post-aliasing artifact has been shown that they have a great influence on the quality. In order to decrease the effect of the aliasing, the lower order is selected. However, the easing aliasing method is more effective.

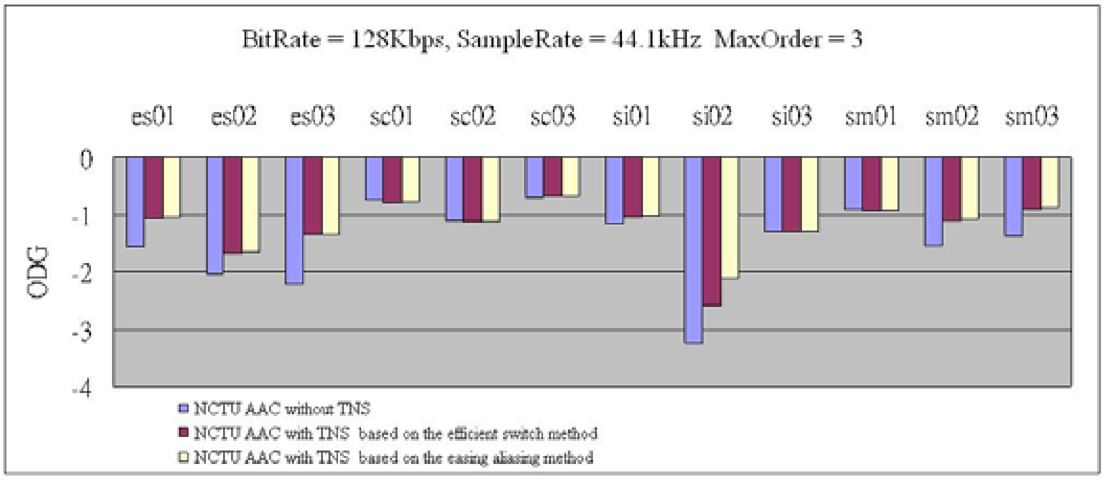


Figure 27: Objective test on the three methods: “NCTU-AAC 1.0 without TNS”, “NCTU-AAC 1.0 with TNS based on the efficient switch method” and “NCTU-AAC 1.0 with TNS based on the easing aliasing method”.

Figure 27 explains the comparison of NCTU-AAC 1.0 without TNS, NCTU-AAC 1.0 with TNS based on the efficient switch method and NCTU-AAC 1.0 with TNS based on the easing aliasing method. The comparison between the efficient switch method and the easing aliasing method is that, except the es03, sc02, sc03, si03 and sm01, the easing aliasing method also has an improvement, especially in the si02. In Section 3.4 , it's mentioned that, with the increase of filter order, the noise will be more concentrated on the signal with high energy. But, there is an issue that the signal with high energy can mask such big noise. After easing the aliasing artifact, Figure 28 explains that neither the biggest value 12 nor the small value is the best MaxOrder. The optimum value is 6. Therefore, the experiment has shown that we prefer to spread a little noise to the signal with low energy not to concentrate the total noise on the signal with high energy.

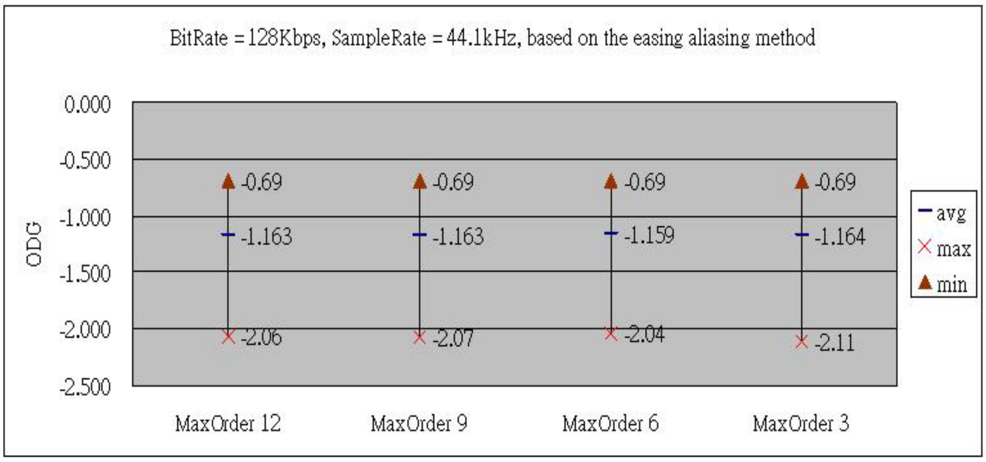


Figure 28: ODG for different MaxOrder based on the easing aliasing method.

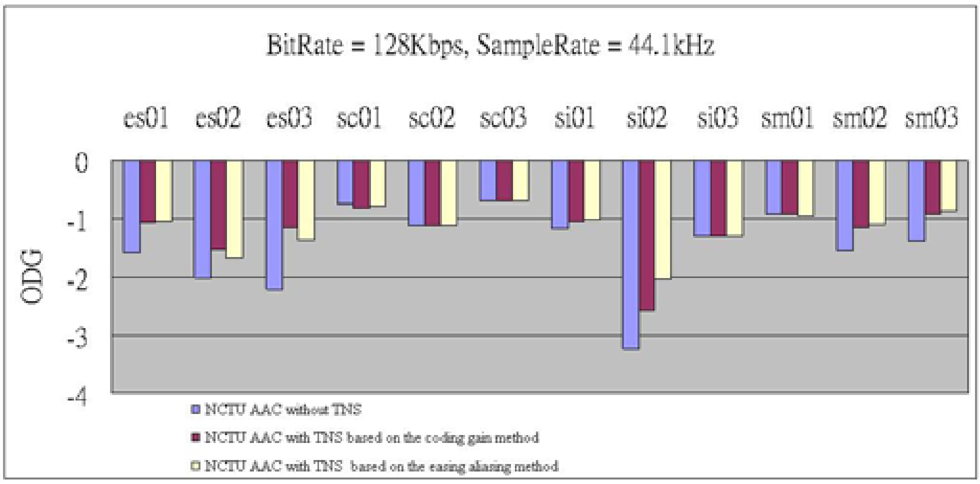


Figure 29: Objective test on the three methods: "NCTU-AAC 1.0 without TNS", "NCTU-AAC 1.0 with TNS based on the coding gain method" and "NCTU-AAC 1.0 with TNS based on the easing aliasing method".

Table 5: The twelve test tracks for quality evaluation.

Track		Signal Description			
		Signal	Mode	Time(sec)	Remark
1	es01	vocal (Suzan Vega)	Stereo	10	(c)
2	es02	German speech	Stereo	8	(c)
3	es03	English speech	Stereo	7	(c)
4	sc01	Trumpet solo and orchestra	Stereo	10	(d)
5	sc02	Orchestral piece	Stereo	12	(d)
6	sc03	Contemporary pop music	Stereo	11	(d)
7	si01	Harpsichord	Stereo	7	
8	si02	Castanets	Stereo	7	(a)
9	si03	pitch pipe	Stereo	27	(b)
10	sm01	Bagpipes	Stereo	11	(b)
11	sm02	Glockenspiel	Stereo	10	(a)
12	sm03	Plucked strings	Stereo	13	

Remark:

(a) Transients: pre-echo sensitive, smearing of noise in temporal domain.

(b) Tonal/Harmonic structure: noise sensitive, roughness.

(c) Natural vocal (critical combination of tonal parts and attacks): distortion sensitive, smearing of attacks.

(d) Complex sound: stresses the Device Under Test.

(e) High bandwidth: stresses the Device Under Test, loss of high frequencies, program-modulated high frequency noise.

(f) Low volume testing.

In order to measure the TNS quality, a large number of test bitstreams are needed. In PSPLab audio database [20], there are 16 sets and 327 tracks. For each bitstream set, there are briefly described in Table 6

Figure 30 illustrates three experiments for the 16 bitstream sets. For each bar chart, it means the average ODG of each bitstream set. Generally, the easing aliasing method has a better quality than the coding gain method. In detail, for 327 tracks, Figure 31 illustrates the improvement tracks distribution for the two different methods. The x-axis represents four different improvement ranges and the y-axis means the number of tracks improved. It's deserved to be mentioned that by the easing aliasing method, the number of tracks that the ODG improvement is beyond

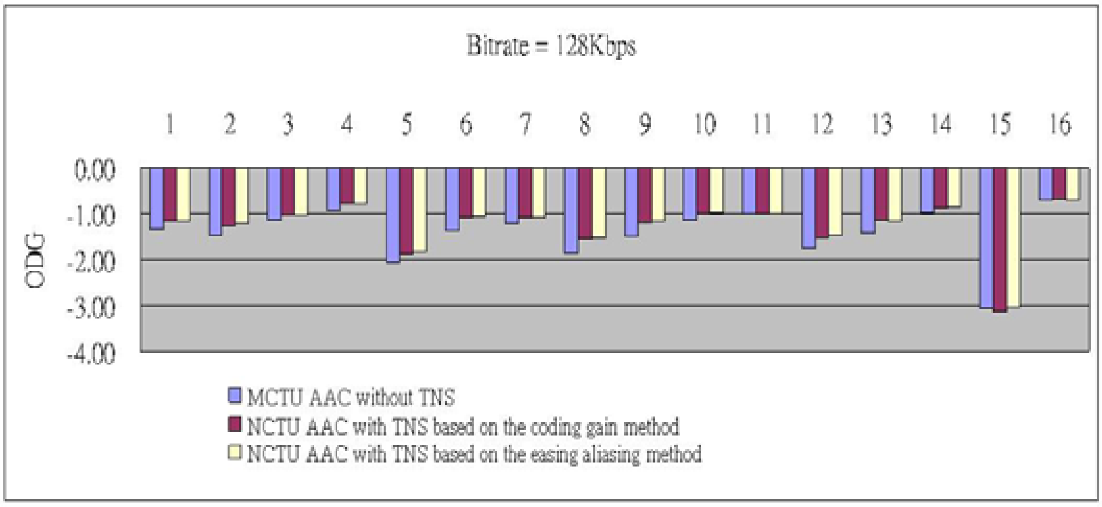


Figure 30 : For 16 bitstream sets, objective test on the three methods: "NCTU-AAC 1.0 without TNS", "NCTU-AAC 1.0 with TNS based on the coding gain method" and "NCTU-AAC 1.0 with TNS based on the easing aliasing method".

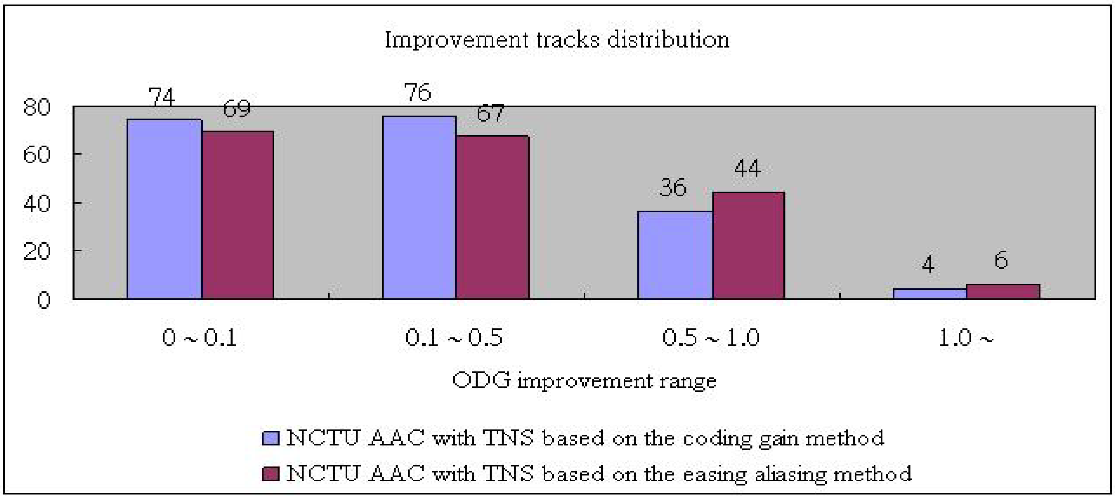


Figure 31: The improvement tracks distribution.

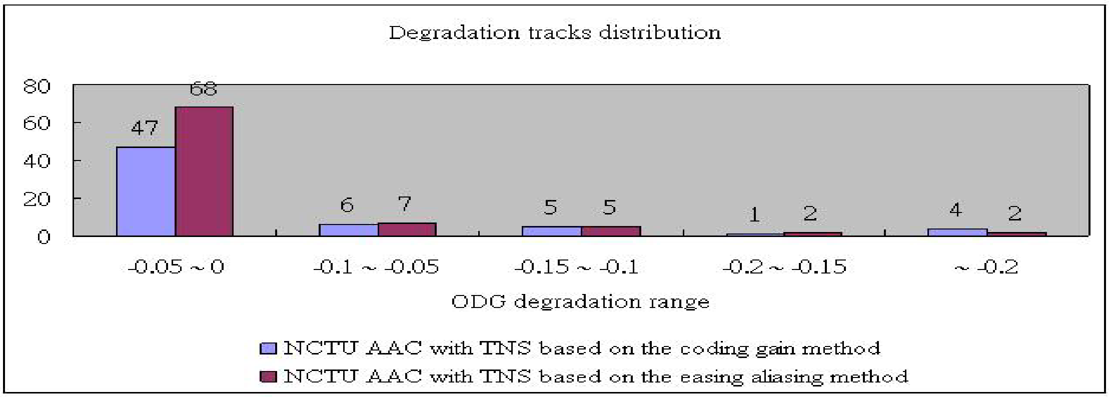


Figure 32: The degradation tracks distribution.

0.5 are more than by the coding gain method. Besides, like Figure 31, Figure 32 illustrates the degradation tracks distribution for the two different methods. Although the easing aliasing method has worse quality at the range “-0.05 ~ 0” than the coding gain method, the ODG degradation of -0.05 is acceptable. But, the number of tracks the ODG degradation is beyond -0.1 is nine by the easing aliasing method. The reason causing the bad quality is that, when the condition 3 in Algo 5 is satisfied, the aliasing produced by the TNS is not cancelled.

Table 6: The description for each Bitstream set

Item	Bitstreams categories	Number of Tracks	Remark
1	ff123	103	Killer bitstream collection from ff123.
2	gpsycho	24	LAME quality test bitstream.
3	HA64KTest	39	64 Kbps test bitstream for multi-format in HA forum.
4	HA128KTestV2	12	128 Kbps test bitstream for multi-format in HA forum.
5	horrible_song	16	Collections of killer songs among all bitstream in PSPLab.
6	ingets1	5	Bitstream collection from the test of OGG Vorbis pre 1.0 listening test.
7	Mono	3	Mono test bitstream.
8	MPEG	12	MPEG test bitstream set for 48KHz.
9	MPEG44100	12	MPEG test bitstream set for 44100 Hz.
10	Phong	8	Test bistream collection from Phong.
11	PSPLab	37	Collections of bitstream from early age of PSPLab. Some are good as killer.
12	sjeng	3	Small bitstream collection by sjeng.
13	SQAM	16	Sound quality assessment material recordings for subjective tests.
14	TestingSong14	14	Test bitstream collection from rshong.
15	TonalSignals	15	Artificial bitstream that contains sin wave etc.
16	VORBIS_TESTS_Samples	8	
	Total	327	

5.4 Complexity

For the coding gain method, each of the input frames must conduct the TNS filtering module, the complexity focus on the Levinson-Durbin algorithm. First, the autocorrelation, the complexity of which is $O(kL)$ where k is the number of the reflections coefficients and L is the range of the spectral coefficients, is computed. Then, the complexity of Algo 1 is $O(k^2)$. The whole complexity of the algorithm is $O(kL + k^2)$. Therefore, the complexity of the TNS method is $O(M(kL+k^2))$, where M is the number of input frames. However, with the easing aliasing method, TNS filtering is applied only when attack flag is active. The complexity of the method needs the additional $O(N)$ load to calculate each energy of the eight zones and detect the attack position, where N is the size of the input frame. So, the whole complexity is reduced to $O(\ m(kL + k^2 + N)\ )$, where m is the number of the attack frames in the entire frames. For most tracks, the number of frames that attack flag is active may be only a small portion less than 20% in Figure 33. Hence, the complexity is highly reduced.

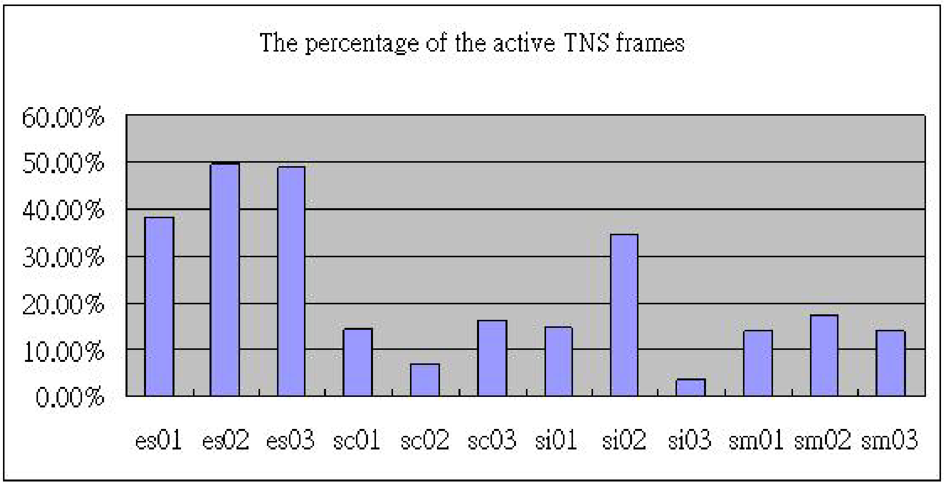


Figure 33: The percentage of the active TNS frames.

Chapter 6 Conclusion

This thesis discusses three artifact resulted from the combination of the MDCT filterbank and the TNS module and the existing coding gain transient detector taking too much complexity to detect the attack signal. An efficient PE method is proposed to counter this problem. The proposed detector not only reduces the computation complexity but also takes the pre-aliasing and post-aliasing artifact into account. The algorithm has been implemented into the NCTU-AAC 1.0 encoder and the objective test is conducted based on the recommendation system by ITU-R Task Group 10/4. The proposed efficient method is proven to improve the objective quality measure over the traditional coding gain detection method.

Reference

[1] ISO/IEC, “Coding of Moving Pictures and Audio –IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC)”, Doc. ISO/IEC JTCI/SC29/WG11 n1650, Apr. 1997.

[2] ISO/IEC, “Information Technology- Coding of audiovisual objects”—ISO/IEC.D 4496 (Part 3, Audio), 1999

[3] ISO/IEC, “Information Technology- Coding of audiovisual objects”—ISO/IEC.D 14496 (Part 3, Audio), 1999.

[4] J. Herre and J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)", Proc. 101st AES Conv., Los Angeles, Nov. 1996.

[5] J. Herre and J. D. Johnston, "Continuously signal-adaptive filterbank for high-quality perceptual audio coding,” 1997 IEEE ASSP Workshop, 19-22 Oct. 1997.

[6] J. Herre and J. D. Johnston, "Exploiting Both Time and Frequency Structure in a System that Uses an Analysis/Synthesis Filterbank with High Frequency Resolution", Proc. 103rd AES Conv., New York, Sept. 1997.

[7] J. Herre, “Temporal Noise Shaping, Quantization And Coding Methods in Perceptual Audio Coding: A Tutorial Introduction,” The AES 17th International Conference: High-Quality Audio Coding, pp 17-31, Sept. 1999, pp 17-31.

[8] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer-Verlag, Berlin Heidelberg, 1990.

[9] E. Allamanche, R. Geiger, J. Herre, T. Sporer, “MPEG-4 Low Delay Audio Coding Based on the AAC Codec”, 106th AES Conv., 1999.

[10] J. Herre, “Perceptual noise shaping in the time domain via LPC prediction in the frequency domain”, US Patient 5,781,888

[11] J. Herre, U. Gbur, A. Ehret, M. Dietz, B. Teichmann, O. Kunz, K. Brandenburg and H. Gerhauser, “Method for coding an audio signal”, US Patient 6,424,939

[12] J. D. Markel and A. H. Gray “Linear Prediction of Speech”, Berlin: Springer Verlag, 1976.

[13] D.C. Farden, “Solution of a Toeplitz Set of Linear Equations”, IEEE Transaction on Antennas and Propagation, AP-24:906-907, Nov. 1976.

[14] Y. Wang and M. Vilermo, “Modified Discrete Cosine Transform— Its Implications for Audio Coding and Error Concealment”, J.Audio Eng. Soc. , vol. 51, No. ½, 2003 Jan./Feb..

[15] Y. Wang, L. Yaroslavsky, M. Vilermo, M. Väänänen, “Restructured Audio Encoder for Improved Computational Efficiency,” AES 108th International Convention, February 19-22, 2000, Paris, France.

[16] Y. Wang, L. Yaroslavsky, M. Vilermo, “On the Relationship between MDCT, SDFT and DFT,” 16th IFIP World Computer Congress (WCC2000)/5th International Conference on Signal Processing (ICSP2000), August 21-25, 2000, Beijing, China.

[17] C. M. Liu, W. C. Lee and T. W. Chang, “The Efficient Temporal Noise Shaping Method”, 116th AES Conv., 2004

[18] ITU Radiocommunication Study Group 6, “DRAFT REVISION TO RECOMMENDDATION ITU-R BS.1387- Method for objective measurements of perceived audio quality”.

[19] NCTU-AAC 1.0 website http://psplab.csie.nctu.edu.tw/projects/nctu-aac.html

[20] PSPLab audio database

http://psplab.csie.nctu.edu.tw/projects/index.pl/testbitstreams.html