

國立交通大學

資訊工程學系

碩士論文

MPEG Layer III 與 MPEG-4 AAC 與 MPEG-4 HE-AAC
上之有效率的位元儲存分配器設計



Efficient Bit Reservoir Design for
MPEG Layer III, MPEG-4 AAC, and MPEG-4 HE-AAC

研究生：陳立偉

指導教授：劉啟民 教授

李文傑 博士

中華民國 九十四 年 六 月

MPEG Layer III 與 MPEG-4 AAC 與 MPEG-4 HE-AAC

上之有效率的位元儲存分配器設計

Efficient Bit Reservoir Design for

MPEG Layer III, MPEG-4 AAC, and MPEG-4 HE-AAC

研 究 生：陳立偉

Student : Li-Wei Chen

指 導 教 授：劉啟民

Advisor : Dr. Chi-Min Liu

李文傑

Dr. Wen-Chieh Lee



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 2005

HsinChu, Taiwan, Republic of China

中華民國 九十四 年 六 月

MPEG Layer III 與 MPEG-4 AAC 與 MPEG-4 HE-AAC

上之有效率的位元儲存分配器設計

學生：陳立偉

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

位元儲存分配器擔負著回收量化後殘餘位元與管控訊框之間位元分配的責任，在目前的音訊壓縮器，如 MP3、AAC 中，扮演了平衡有限位元與壓縮品質之間的核心角色。位元儲存分配器的設計可以從需求導向與儲量導向兩種方式來探討：需求導向的方法根據音訊內容決定所需分配的位元量；儲量導向的方法則是根據位元儲存器中所累積儲存的位元多寡決定所需分配的位元量。現存的位元儲存分配器設計主要是依循儲量導向的方式來實作。本論文中提出一個綜合需求導向與儲量導向的有效率位元儲存分配器設計：經由需求預測器，我們可以適切的估測出每一個訊框的位元需求；同時透過儲量管理器，我們可以根據壓縮器協定與偏好的模組設定來控制分配的位元量。

更進一步來說，爲了在低於 96kbps 的位元率之下達到良好的壓縮品質，因而提出了結合 AAC 壓縮器與 SBR 模組的 HE-AAC 壓縮器。SBR 模組藉由複製訊號的低頻部分來重建高頻部分，使得 AAC 壓縮器可以專注在處理訊號的低頻部分。因此，妥善的分配位元於 AAC 壓縮器與 SBR 模組之間決定了壓縮的品質與效率。在本論文中，我們將位元儲存分配器的概念延伸至 HE-AAC 上並有效的分配位元於 AAC 壓縮器與 SBR 模組之間。同時爲了驗證品質的改進與效率的增進，我們採用了主觀的聆聽評量與客觀的軟體量測，並且獲得良好的結果。

Efficient Bit Reservoir Design for MPEG Layer III, MPEG-4 AAC, and MPEG-4 HE-AAC

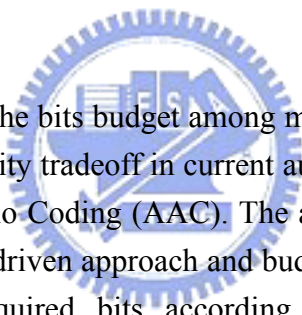
Student: Li-Wei Chen

Advisor: Dr. Chi-Min Liu

Dr. Wen-Chieh Lee

Institute of Computer Science and Information Engineering
National ChiaoTung University

ABSTRACT



Bit reservoir controlling the bits budget among music frames has been the kernel module to have good bits-quality tradeoff in current audio encoders like MPEG Layer III (MP3) and Advanced Audio Coding (AAC). The approaches of bit reservoirs can be investigated from demand-driven approach and budget-driven one. Demand-driven approach determines the required bits according to the audio contents while budget-driven one allocates bits according to the bit budgets accumulated in the bit reservoir. Existing bit reservoirs follow basically the budget-driven approach. This thesis presents an efficient bit reservoir design with concerns from both demand and budget. The bit reservoir includes a demand estimator to adaptively predict the bits required for each frame. Also, there is a budget regulator to control the bits used according to the codec protocol and the preferred scenario.

Furthermore, High efficiency AAC (HE-AAC) has included the Spectral Band Replication (SBR) in combination with AAC to achieve high audio quality at bit rates lower than 96 kbps. SBR reconstructs high frequency signal through replicating the low frequency parts. The bits allocated to AAC encoder module and SBR module decides the quality and compression efficiency. This thesis also extends the concept of bit reservoir to HE-AAC for efficient bits distribution between the AAC encoder and the SBR module. Both subjective and objective tests are conducted to verify the improved quality and efficiency of the new bit reservoir design.

致謝

感謝指導老師劉啟民教授兩年來的指導與栽培，讓我在紮實的訓練中對於口語表達能力及文字撰寫技巧上皆能獲得長足的進步。同時在老師的引領下更讓我有機會可以將研究的成果發表於國外的會議論文，如此的鼓勵加深了我對研究主題的信心和興趣，也令我在這兩年的碩士班求學過程中留下了難忘的紀錄與回憶。同時感謝李文傑博士的指導，讓我在發掘問題與解決問題的能力上學習到很多的技巧，使我獲益良多。感謝博士班的楊宗翰學長與許瀚文學長經常適時地指正我在研究理論與方法上的缺失，使我能及時修正方向並精益求精。感謝已經畢業的蕭又華學長、彭康硯學長、張子文學長和邱挺學長，在龐大的編碼器專案中，有了學長們辛苦而卓越的研究成果為基礎，學弟才能夠專心的在研究主題上盡情發揮。感謝同學蘇明堂不時教導我程式的技巧，拯救我在寫程式時遇到的瓶頸，同時在這兩年之中，一同寫作業、準備考試、討論研究心得、相互吐嘈，在枯燥的研究生生活中增添了不少趣味。另外要感謝學弟楊詠成、張家銘、唐守宏和李侃峻的協助，讓我可以將研究方向延伸到新的編碼器協定之中。對於其他曾經提供協助與鼓勵的同學、朋友們，在此一併表達個人由衷的感謝之意。

最後，感謝我的父母從小至今對我的養育栽培，並在研究所兩年的過程中給予我精神上的關心與鼓勵以及生活上的資助，使我能無後顧之憂、全心全意地在這個專業的領域中研究探索並且完成這本畢業論文。

Contents

Contents	i
Figure List.....	iii
Table List	viii
Chapter 1 Introduction	1
Chapter 2 Backgrounds.....	4
2.1 Psychoacoustic Model	4
2.1.1 Absolute Threshold of Hearing.....	4
2.2.2 Critical Bands.....	6
2.2.3 Masking Effects	8
2.2.3.1 Simultaneous masking	9
2.2.3.1 Nonsimultaneous masking.....	9
2.2 Psychoacoustic Model in MP3 and AAC.....	10
2.3 Perceptual Entropy.....	12
Chapter 3 Bit Reservoir Design in Current Audio Codec.....	15
3.1 Bit Reservoir Schemes in MP3 Codec.....	15
3.1.1 Frame Format.....	15
3.1.2 Bit Rate Scenario	16
3.1.3 Recommended Scheme in Standard.....	17
3.1.4 Scheme in LAME 3.88	18
3.2 Bit Reservoir Schemes in AAC	20
3.2.1 Bitstream Format	20
3.2.2 Recommended Schemes in Standard	22
3.2.3 Schemes in FAAC 1.24.....	22
3.3 Bit Reservoir Schemes in HE-AAC.....	23
3.3.1 Bitstream Overview	23
3.3.2 Schemes in 3GPP	23
Chapter 4 Efficient Bit Reservoir for MP3 and AAC.....	25
4.1 Allocation Entropy	25
4.2 Demand Estimator	29
4.2.1 AE Average	29
4.2.2 Demand Ratio	30
4.2.2.1 Demand Curve for MP3 CBR Mode	31
4.2.2.2 Demand Curve for MP3 ABR Mode	32

4.2.2.3 Demand Curve for AAC	33
4.3 Budget Regulator	33
4.3.1 Budget Curve for MP3 CBR Mode	34
4.3.2 Budget Curve for MP3 ABR Mode	35
4.3.3 Budget Curve for AAC	36
4.4 Allocated Bits Calculation	36
4.5 Experiments for MP3 and AAC	38
4.5.1 Objective quality evaluation	39
4.5.1.1 Objective quality evaluation for MP3 CBR mode.....	39
4.5.1.2 Objective quality evaluation for MP3 ABR mode.....	42
4.5.1.3 Objective quality evaluation for AAC	43
4.5.2 Parameter Evaluation	44
4.5.2.1 Parameter evaluation for MP3 CBR mode	45
4.5.2.2 Parameter evaluation for MP3 ABR mode	46
4.5.2.3 Parameter evaluation for AAC.....	47
4.5.3 Objective quality measurement based on music database	49
4.5.3.1 Objective quality measurement based on music database for MP3 CBR mode.....	50
4.5.3.2 Objective quality measurement based on music database for MP3 ABR mode.....	52
4.5.3.3 Objective quality measurement based on music database for AAC	54
4.5.4 Objective quality measurement with existing codecs.....	55
Chapter 5 Bit Reservoir Design for HE-AAC	59
5.1 Spectral Band Replication (SBR)	59
5.2 Demand Estimator for SBR	61
5.3 HE-AAC Bit Allocation.....	65
5.4 Experiments for HE-AAC.....	67
5.4.1 Objective quality evaluation for HE-AAC	67
5.4.2 Parameter evaluation for HE-AAC.....	71
5.4.3 Objective quality measurement based on music database for HE-AAC	72
5.4.4 Objective quality measurement with existing codecs.....	75
5.4.5 Subjective quality evaluation for HE-AAC	76
5.4.5.1 MUSHRA	77
5.4.5.2 Results of listening test.....	79
Chapter 6 Conclusion.....	81
Reference	82

Figure List

Figure 1: General Perceptual encoder.....	1
Figure 2: Block diagram of HE-AAC encoder.....	2
Figure 3: Experiment result of absolute threshold of hearing [9].	5
Figure 4: The absolute threshold of hearing in quiet.	5
Figure 5: The structure of human ear [8].	6
Figure 6: Critical bandwidth measurement: (a) and (c) detection threshold decreases as masking tones transition from auditory filter passband into stopband; (b) and (d) the same interpretation with roles reversed [11].	7
Figure 7: Critical bandwidth as a function of center frequency [11].	7
Figure 8: Example of simultaneous masking.	9
Figure 9: Example of nonsimultaneous masking [8].	10
Figure 10: Block diagram of MPEG Psychoacoustic Model II [12].	11
Figure 11: Flow chart of MPEG Psychoacoustic Model II.	12
Figure 12: Bitstream format of MPEG-1 Layer III [1].	16
Figure 13: Example of MP3 frame structure [1].	16
Figure 14: The flow chart of recommended bit reservoir control in [1].	17
Figure 15: The flow chart of bit reservoir design in LAME 3.88.	19
Figure 16: ADIF bistream.	20
Figure 17: ADTS bitstream.	21
Figure 18: Flow chart of bit reservoir control in [2].	22
Figure 19: Bitstream organization of HE-AAC [3].	23
Figure 20: Flow chart of 3GPP HE-AAC.	24
Figure 21: Effective bandwidth for MP3 (Long Window, Sample Rate: 44.1 KHz) [19].	26
Figure 22: Effective bandwidth for AAC (Long Window, Sample Rate: 44.1 KHz) [19].	27
Figure 23: The spectrogram (top) and the values of AE (bottom) of natural vocal (es03).	28
Figure 24: The spectrogram (top) and the values of AE (bottom) of complex sound (sc02).	28
Figure 25: The spectrogram (top) and the values of AE (bottom) of transient (si02).	28

Figure 26: The spectrogram (top) and the values of AE (bottom) of harmonic (si03).	29
Figure 27: Flow chart of $AE_{average}$ calculation.....	30
Figure 28: Demand curve for MP3 CBR mode.	31
Figure 29: Demand curve for MP3 ABR mode.	32
Figure 30: Demand curve for AAC ABR mode.....	33
Figure 31: Budget curve for MP3 CBR mode.	34
Figure 32: Budget curve for MP3 ABR mode.	35
Figure 33: Budget curve for AAC ABR mode.....	36
Figure 34: Flow chart of the efficient bit reservoir design.....	37
Figure 35: The ODG range comparison of Table 3. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.	40
Figure 36: The ODG range comparison of Table 4. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.	41
Figure 37: The ODG range comparison of Table 5. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.	43
Figure 38: The ODG range comparison of Table 6. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.	44
Figure 39: The average objective quality of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	51
Figure 40: The enhancement tracks distribution of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	51
Figure 41: The degradation tracks distribution of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	51

Figure 42: The average objective quality of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	52
Figure 43: The enhancement tracks distribution of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	53
Figure 44: The degradation tracks distribution of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).....	53
Figure 45: The average objective quality of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).....	54
Figure 46: The enhancement tracks distribution of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).....	55
Figure 47: The degradation tracks distribution of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).....	55
Figure 48: The ODG range comparison of Table 11.....	57
Figure 49: The ODG range comparison of Table 12.....	57
Figure 50: The ODG range comparison of Table 13.....	58
Figure 51: Block diagram of SBR encoder.....	60
Figure 52: The spectrogram of an interval in input signal with the superimposed envelope time-frequency grid [5].	61
Figure 53: Block diagram of bit reservoir for HE-AAC.....	61
Figure 54: Natural vocal (es03).	62
Figure 55: Complex sound (sc02).....	62
Figure 56: Transient (si02).....	63
Figure 57: Harmonic (si03).....	63
Figure 58: The flow chart of bit reservoir design for HE-AAC.....	66
Figure 59: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 48kbps.	68

Figure 60: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 64kbps.	68
Figure 61: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 80kbps.	68
Figure 62: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 96kbps.	69
Figure 63: The ODG range comparison of Table 15. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.	70
Figure 64: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 48 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	72
Figure 65: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 64 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	73
Figure 66: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 80 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	73
Figure 67: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 96 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	73
Figure 68: The enhancement tracks distribution of NCTU-HEAAC without bit reservoir and with new bit reservoir at different bit rates for the 16 bitstream sets in PSPLAB audio database. Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	74
Figure 69: The degradation tracks distribution of NCTU-HEAAC without bit reservoir and with new bit reservoir at different bit rates for the 16 bitstream sets in PSPLAB audio database. Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).	74
Figure 70: The ODG range comparison of Table 17.	76
Figure 71: ITU-R five-grade impairment scale.	77
Figure 72: Main dialog box of ABC/Hidden Reference Audio Comparison Tool.	78

Figure 73: ABX dialog box of ABC/Hidden Reference Audio Comparison Tool. 79

Figure 74: Subjective quality evaluation of NCTU-HEAAC with and without new bit reservoir design at bit rate 80kbps. 80



Table List

Table 1: Idealized critical band filter bank [8].	8
Table 2: The twelve test tracks recommended by MPEG.	38
Table 3: Objective measurements through the ODGs for different bit reservoir designs in MP3 CBR mode (Long/Short window, M/S coding).	40
Table 4: Objective measurements through the ODGs for different bit reservoir designs in MP3 CBR mode (Long window, without M/S coding).	41
Table 5: Objective measurements through the ODGs for different bit reservoir designs in MP3 ABR mode.	42
Table 6: Objective measurements through the ODGs for different bit reservoir designs in AAC.	43
Table 7: MP3 CBR mode parameters evaluation.	45
Table 8: MP3 ABR mode parameters evaluation.	46
Table 9: AAC parameters evaluation.	48
Table 10: The PSPLAB audio database.	49
Table 11: Objective quality comparison for MP3 CBR mode at different bit rates.	56
Table 12: Objective quality comparison for MP3 ABR mode at different bit rates.	57
Table 13: Objective quality comparison for AAC at different bit rates.	58
Table 14: The minimum, maximum, average, and standard deviation of bits usage at 80 kbps. The “Minimum” and “Maximum”, “Average”, and “Standard Deviation” columns denote respectively the minimum, the maximum bits, the average bits, and the standard deviation used in the SBR encoder among all the frames in the correspondent track. The percentage in each the above category column is the bit percentage for the budget in a frame at bit rate 80 kbps.	63
Table 15: Objective measurements through the ODGs for different bit reservoir designs at different bit rates in HE-AAC (Long/Short window, M/S coding, and TNS).	69
Table 16: HE-AAC parameters evaluation.	71
Table 17: Objective quality comparison for HE-AAC at different bit rates.	



Chapter 1

Introduction

Current perceptual audio encoders like MPEG-1 Layer 3 (MP3) [1] and MPEG-4 Advanced Audio Coding (AAC) [2] has included a mechanism referred to as the bit reservoir to control the bits variation among frames. The mechanism provides the space to loan or deposit bits to control the audio quality under a bit rate constraint.

The general perceptual encoders can be considered in Figure 1. The audio signal is segmented into frames for encoding. The bit allocation module assigns the available bits provided by the bit reservoir to quantize bands according to the information from the psychoacoustic models. The bit reservoir deciding the dynamic of the available bits among frames is the quality buffer avoiding the severe quality degradation from critical frames.

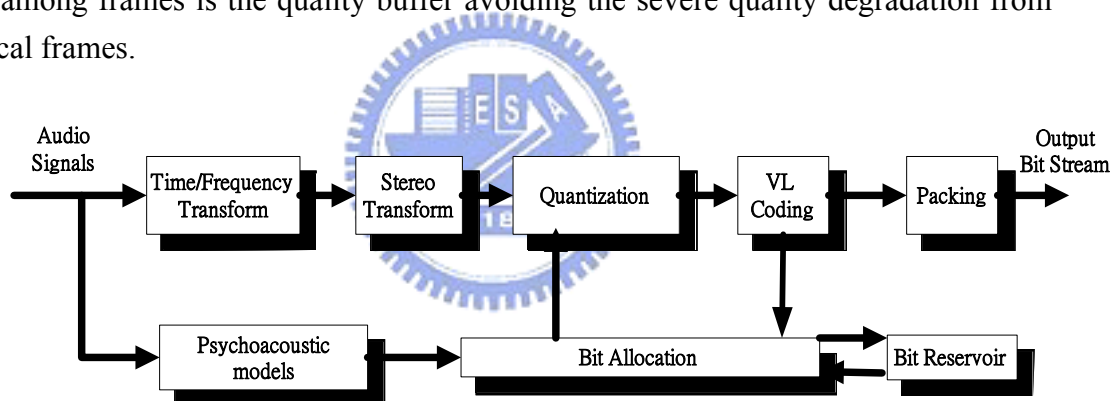


Figure 1: General Perceptual encoder.

The explicit bit reservoir in MP3 or the implicit bit reservoir in AAC have been used to efficiently maintain the frame bits and quality during varying audio contents such as attack, critical tracks, and silence. The bit reservoir design can be considered from the bit demand to compress the basic time frame in a track and the bit budget to regulate the consumed bits. The methods of bit reservoir can be classified into demand-driven method and budget-driven method. Demand-driven approach determines the required bits according to the audio contents while budget-driven one allocates bits according to the bit budgets accumulated in the bit reservoir.

This thesis considers the design through two novel modules: the demand estimator and the budget regulator. The demand estimator adaptively predicts the bits required for a frame to achieve a specific quality. The budget regulator controls the bit

budget according to the codec protocol and preferred scenario. The codec protocols affect the buffering size and resolution of the bit deposit and loan. On the preferred scenario, there are in general three cases: the constant bit rate (CBR), the variable bit rate (VBR), and the average bit rate (ABR). The CBR allows very limited bit variation in consecutive frames. The VBR in general have no regulation on the bit rates. The ABR allow the constant bit rates over a time period longer than several time-frames. The bit reservoir presented in this thesis can adjust the bit dynamic to efficiently maintain the audio quality for the CBR and the ABR.

In order to achieve high audio quality at bit rates lower than 96 kbps, High Efficiency AAC (HE-AAC) is proposed. HE-AAC is the extension of the conventional AAC codec by supporting the Spectral Band Replication (SBR) module [3][4][5][6]. The block diagram of the HE-AAC is illustrated in Figure 2. The audio signal is fed into the filterbank and split into high frequency signal $s_h(n)$ and low frequency signal $s_l(n)$ through a filterbank. The low frequency signal $s_l(n)$ is half the sampling rate of the original signal. The high frequency signal $s_h(n)$ is reconstructed through the band replication technique from the low frequency signal $s_l(n)$. The replication parameters are used to keep the reconstructed high frequency bands perceptually similar to the original high frequency bands. The bit reservoir finds the suitable bit distribution between the AAC encoder and SBR encoder according to the signal contents and the available bit budget.

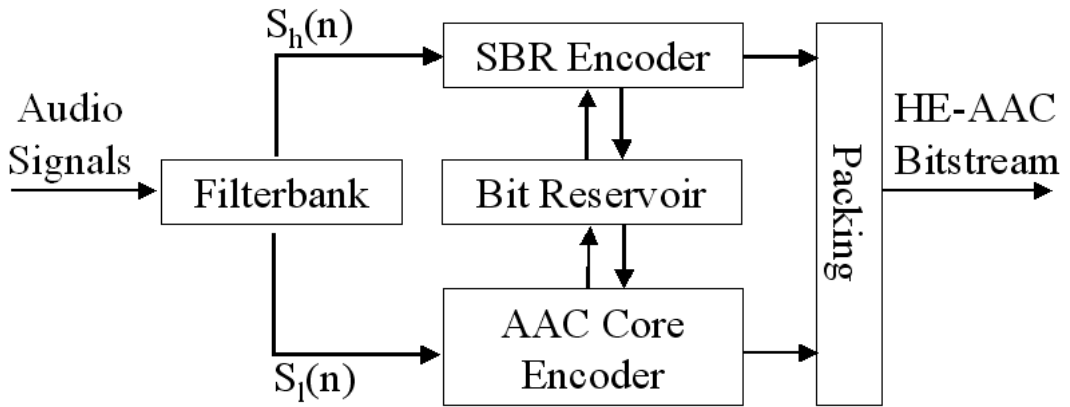


Figure 2: Block diagram of HE-AAC encoder.

The bit reservoir design in AAC should be extended to HE-AAC. On the bit allocation for the AAC encoder and SBR encoder, the problem leads to a closely dependent issue. The SBR, which reconstructs the high frequency signal from the low frequency signal encoded, needs to check the AAC encoding results to predict the bit required. However, the AAC also need to know the bit consumed by SBR to efficiently encode the signals based on the available bits. Furthermore, the bit reservoir should control the quality among frames with regulation on the bit variation.

Although this kind of deadlock or interdependent issue can be approached through an iterative manner, the complexity would increase tremendously due to the inherent complexity in AAC encoder and SBR encoder. This thesis proposes a single iteration approach on the bit reservoir. Based on the demand estimator and budget regulator, we design a SBR demand estimator through a recurrent mechanism. Also, on the budget regulator, we modify the budget regulator that was used in AAC to be the one for both AAC and SBR.

This thesis is organized as follows: Chapter 2 introduces the fundamental knowledge of psychoacoustic model. Chapter 3 introduces the related bit reservoir design in current audio codec. Chapter 4 presents an efficient bit reservoir design for MP3 and AAC through demand estimator and budget regulator. Chapter 5 extends the bit reservoir design to HE-AAC through demand estimator of SBR and global budget regulator. Both subjective and objective measurements are conducted to verify the audio quality and efficiency of our bit reservoir design in Chapter 4 and Chapter 5. The objective test is 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 introduces some fundamental background knowledge of the perceptual audio coding. The psychoacoustic model for optimizing coding efficiency and quality is described first. Some psychoacoustic phenomena and measurements are shown in this chapter.

2.1 Psychoacoustic Model

The objective of perceptual audio coding is to achieve transparent audio quality under bit rate constraint. The psychoacoustic model derives the masking thresholds that quantify the maximum amount of coding distortion without introducing audible artifacts from human auditory system. Therefore, the psychoacoustic model allows the coding algorithms and quantization to exploit perceptual irrelevancies. Irrelevant information is identified during signal analysis by incorporating into several psychoacoustic principles including absolute hearing thresholds, critical band analysis, simultaneous masking, the spread of masking along the basilar membrane, and temporal masking. Furthermore, the theory of perceptual entropy [7], which combines these psychoacoustic notions above with basic properties of signal quantization, is proposed. It is a quantitative estimate for transparent audio signal compression.

2.1.1 Absolute Threshold of Hearing

The absolute threshold of hearing, or threshold in quiet, represents the minimum amount of sound level at given frequency to be detected by a listener in a noiseless environment. The absolute threshold is typically expressed in terms of dB SPL, which is a standard metric for quantifying the intensity of an acoustical stimulus [8]. Through the information of absolute threshold, the quantization noise lower than this threshold level would not be perceived by human hearing so some minor details can be ignored during coding process without introducing audio distortion. In general, the threshold stimulus is measured by tuning the sound pressure level of a test tone whose frequency is slowly sweeping from low to high values to the testing listeners. Fletcher [9] reported the frequency dependence of this threshold. The result is close to the

zigzag curve as shown in Figure 3. By connecting the top and bottom points in the zigzag graph, the average of these two curves could be used to evaluate the absolute threshold of hearing.

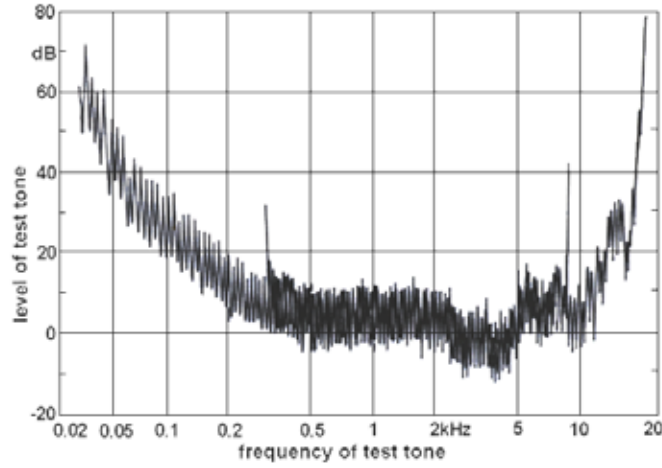


Figure 3: Experiment result of absolute threshold of hearing [9].

Terhardt [10] proposed a well approximated nonlinear function:

$$T_q(f) = 3.64 * \left(\frac{f}{1000}\right)^{-0.8} - 6.5 * e^{-0.6 * \left(\frac{f}{1000} - 3.3\right)^2} + 10^{-3} * \left(\frac{f}{1000}\right)^4 \quad (\text{dB SPL}), \quad (1)$$

where $T_q(f)$ could be deemed the maximum allowable energy level for coding distortion applying to audio coding. The graph of the frequency dependent function above can be depicted as Figure 4.

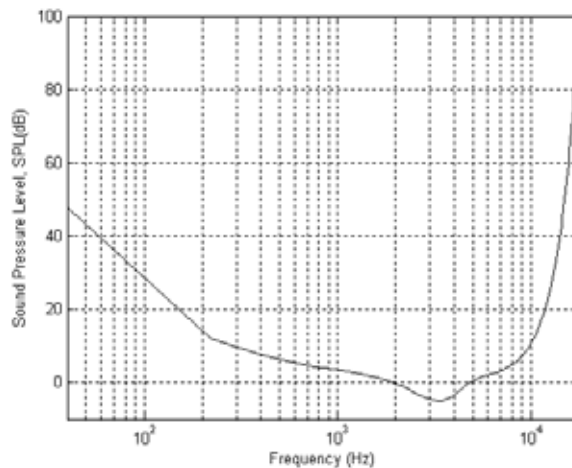


Figure 4: The absolute threshold of hearing in quiet.

Nevertheless, there are two caveats need to be notified. First, the quantization noise is associated with complicated spectrum containing not only pure tone stimuli but also other stimulus such that not all noises can be masked by absolute threshold of hearing. Second, the algorithm designers have no prior knowledge regarding actual

playback levels; hence the curve is often referenced to the coding system by equating the lowest point (i.e., near 4 kHz) to the energy in ± 1 bit of signal amplitude [11].

2.2.2 Critical Bands

The absolute threshold of hearing shapes the basic coding distortion spectrum. However, it is not clear and definite understanding in the coding context. It is necessary to exploit the human hearing model to get maximum coding gain. The structure of human ear is shown in Figure 5. The function of outer ear is to collect sound energy and to transmit this energy through the outer ear canal to the eardrum. The eardrum that is firmly attached to the malleus operates over a wide frequency range as a pressure receiver. The motions of the eardrum are transmitted to the stapes by the middle ear ossicles named malleus, incus, and stapes. The stapes, together with a ring-shaped membrane call the oval window, forms the entrance to the inner ear. The inner ear, cochlea, is shaped like a snail and is embedded in the extremely hard temporal bone. When the oval window receives the excitation from mechanical vibrations, the cochlea structure induces traveling waves along the length of the basilar membrane. The traveling waves generate peak responses at frequency-specific membrane positions, and different neural receptors are effectively tuned to different frequency bands according to their locations. Therefore, the cochlea where the frequency-to-plane transformation takes place can be assumed as a bank of highly overlapping band pass filters [9]. These band pass filters is called the critical bands and its frequency dependent bandwidth is so called “critical bandwidth.”

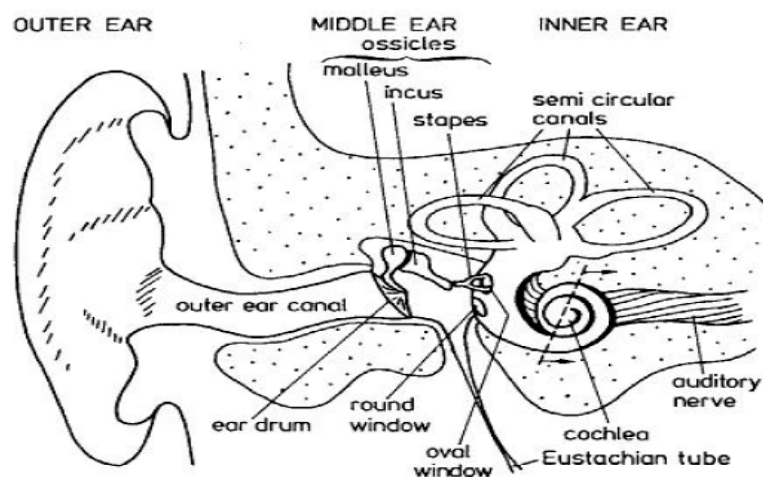


Figure 5: The structure of human ear [8].

The critical bandwidth measurement experiments are shown in Figure 6. Figure 6 (a) and (c) show that the detection threshold for a narrow-band noise source presented

between two masking tones. The threshold remains constant as long as the frequency separation between the two masking tones remains within a critical bandwidth. Beyond this bandwidth, it drops off rapidly. An analogous experiment with reverse masker and maskee is shown in Figure 6 (b) and (d).

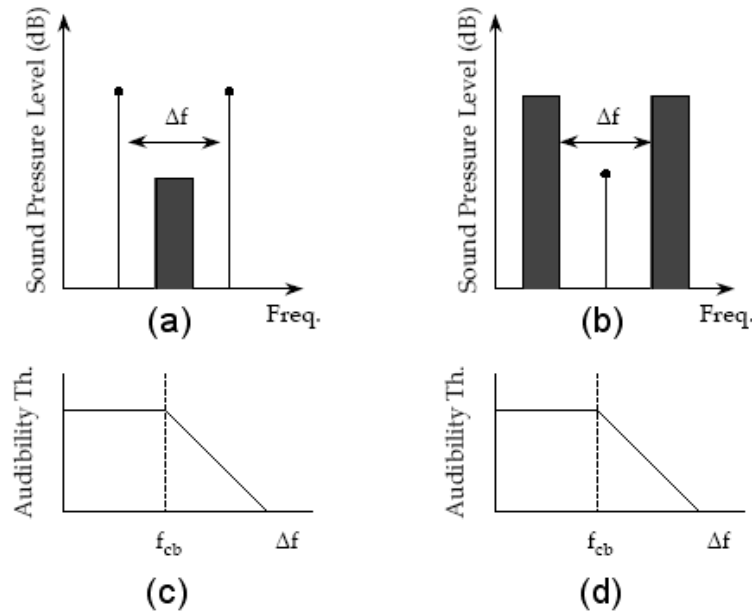


Figure 6: Critical bandwidth measurement: (a) and (c) detection threshold decreases as masking tones transition from auditory filter passband into stopband; (b) and (d) the same interpretation with roles reversed [11].

The critical bandwidth can be conveniently approximated [8] by

$$\Delta f = 25 + 75 \left[1 + 1.4 \left(\frac{f}{1000} \right)^2 \right]^{0.69} \quad (\text{Hz}). \quad (2)$$

The critical bandwidth is narrower in low frequency region and wider in high frequency region. The curve of critical bandwidth and the relation between frequency and critical band can be illustrated in Figure 7.

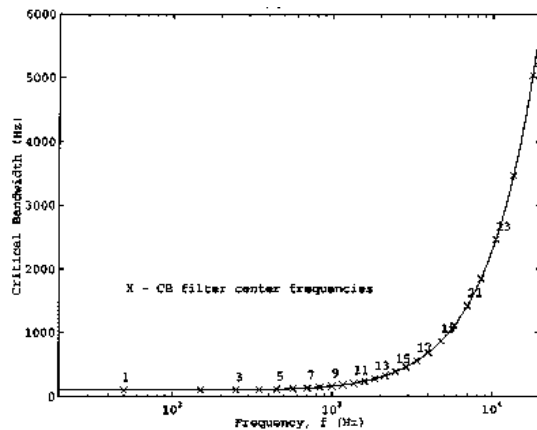


Figure 7: Critical bandwidth as a function of center frequency [11].

In addition, a distance of one critical band is commonly referred to as one ‘‘Bark’’. The formula [8]

$$z(f) = 13 \arctan\left(\frac{0.76f}{1000}\right) + 3.5 \arctan\left[\left(\frac{f}{7500}\right)^2\right] \quad (\text{Bark}). \quad (3)$$

is often used to transfer from frequency in Hertz to the Bark scale. Table 1 gives the transformation that the nonuniform Hertz spacing of the filter bank is actually uniform on Bark scale. The masking curve shapes are much easier to describe with the Bark scale notion.

Table 1: Idealized critical band filter bank [8].

Band No.	Central Frequency (Hz)	Bandwidth (Hz)	Band No.	Central Frequency (Hz)	Bandwidth (Hz)
1	50	0 – 100	14	2150	2000 – 2320
2	150	100 – 200	15	2500	2320 – 2700
3	250	200 – 300	16	2900	2700 – 3150
4	350	300 – 400	17	3400	3150 – 3700
5	450	400 – 510	18	4000	3700 – 4400
6	570	510 – 630	19	4800	4400 – 5300
7	700	630 – 770	20	5800	5300 – 6400
8	840	770 – 920	21	7000	6400 – 7700
9	1000	920 – 1080	22	8500	7700 – 9500
10	1170	1080 – 1270	23	10500	9500 – 12000
11	1370	1270 – 1480	24	13500	12000 – 15500
12	1600	1480 – 1720	25	19500	15500 -
13	1850	1720 – 2000			

2.2.3 Masking Effects

Masking effect is the phenomenon that one sound is inaudible because of the existence of another sound at the same time. It is an important reference for perceptual audio encoder designers to optimize the bit allocation strategy for input signals. If one sound tends to be masked by other sounds, the audio encoder could allocate most bits to the most audible sound and allocate little bits to the insensitive one. However, the relation between masker and maskee is complicated, and it is difficult to exactly analyze the masking effect within them. In general, the masking effects could be discussed from two categories: simultaneous masking (spectral

masking) and nonsimultaneous masking (temporal masking).

2.2.3.1 Simultaneous masking

The simultaneous masking is that the existence of a strong tone or noise masker creates excitation of sufficient strength on the basilar membrane at the critical band location to block effective detection of a weaker signal. This phenomenon in spectral domain is shown in Figure 8. This figure illustrates the masker with strong SPL masks the other weak signals at nearby frequencies. The masking threshold indicates the lowest sound level that can be heard. Therefore, we can focus on the significant components and ignore those unperceived ones.

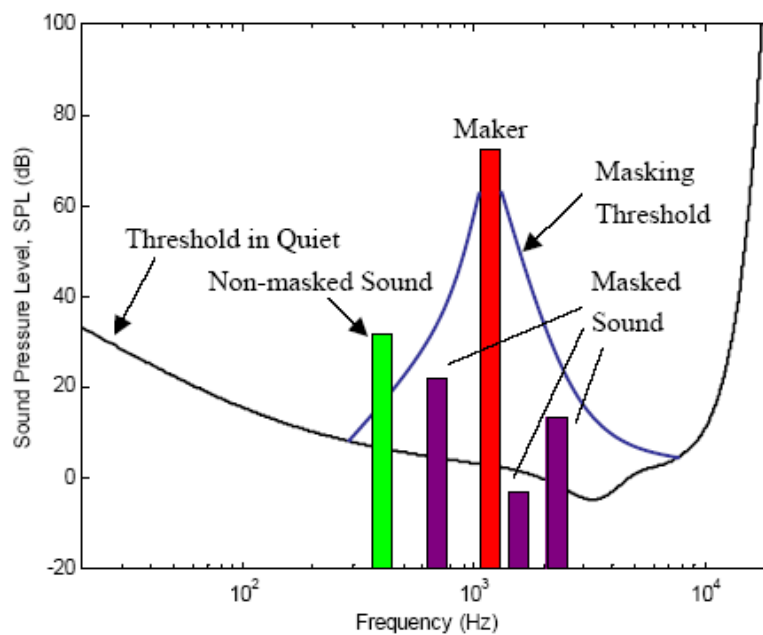


Figure 8: Example of simultaneous masking.

2.2.3.1 Nonsimultaneous masking

Nonsimultaneous masking, or temporal masking, as shown in Figure 9 is different to simultaneous masking in occurrence of maskee. There are two types of temporal masking: pre-masking and post-masking. Pre-masking appears before the onset of the masker; post-masking appears after the masker is vanished.

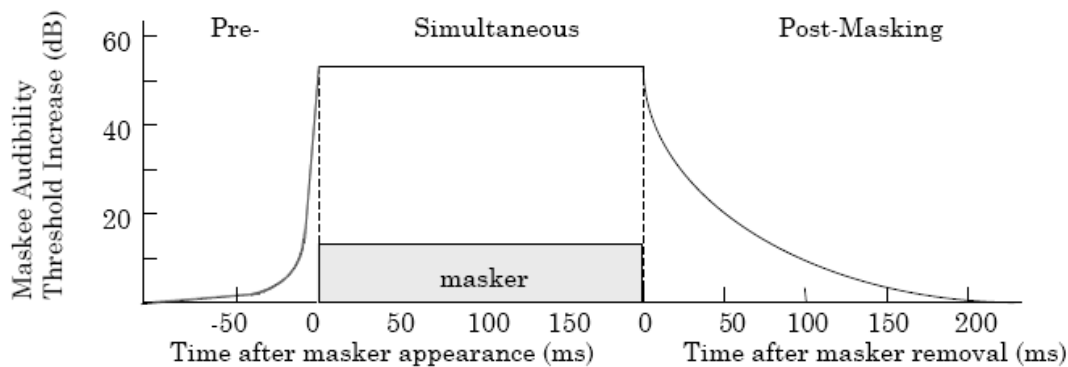


Figure 9: Example of nonsimultaneous masking [8].

The statement of pre-masking is not well comprehensible since it occurs prior to the masker is trigger. The duration of pre-masking lasts 50~60 ms but only a few milliseconds preceding the masker are effective. Pre-masking is an important design issue for pre-echo problem. It has been utilized in conjunction with adaptive block switching to compensate for pre-echo distortions in several audio coding. With referring to post-masking, it presents a momentary masking after the masker and stronger masking effect than pre-masking. Post-masking can sustain more than 100 ms after the maker removal. The duration of post-masking depends on the strength, duration, and relative frequency of masker [8].

2.2 Psychoacoustic Model in MP3 and AAC

There are two psychoacoustic models presented in [1]. The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I or Psychoacoustic Model II. Typically, Model I is applied to MPEG-1 Layer I and II, and Model II to MPEG-1 Layer III, MPEG-2 AAC, and MPEG-4 AAC. The process of psychoacoustic model is mainly to receive the time representation of the signal content over a certain time interval and the corresponding outputs are the signal-to-mask ratio (SMR) for every frequency partition in coders. Based on SMR, the noise shaping allowance and bit allocation are determined for each band in input signals. In this section we only pay attention to Psychoacoustic Model II because Model I is not the main policy adopted in MP3 and AAC encoder design.

A block diagram of Model II is shown in Figure 10. At the first stage, analysis stage, the input data is applied to a Hanning-windowed FFT. The outputs of FFT are grouped into “threshold calculation partitions” which are roughly 1/3 of a critical band or one FFT line wide. The following procedures are divided into two branches. One branch uses the predicted magnitude and phase for unpredictability measure

calculation. The unpredictability is convolved with a spreading function to estimate the tonality index. The property of tonality is that high unpredictability approaching to 0 while low unpredictability approaching to 1. In the other one branch, the partitioned energies are also applying the same spreading function. With the tonality, we can use the noise masking tone (NMT) and tone masking noise (TMN) effects to evaluate the signal-to-noise ratio (SNR). Finally, the actual SMR is derived from SNR and the renormalized signal energies with spreading function.

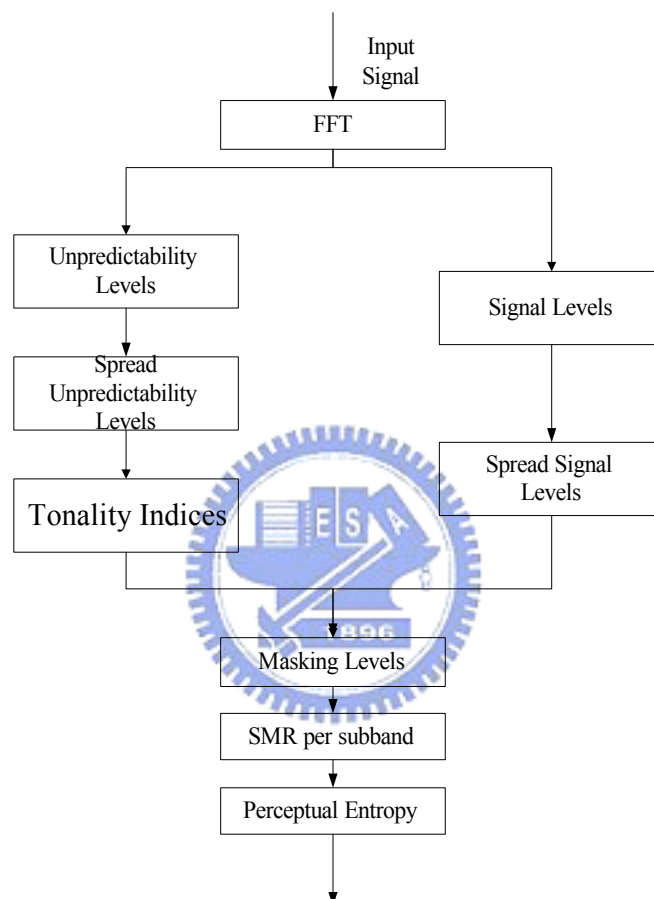


Figure 10: Block diagram of MPEG Psychoacoustic Model II [12].

The precise steps for masking calculation in Psychoacoustic Model II are depicted in Figure 11. We only point out the main function in each step. The details are described in [1] and [2]. $r(w)$ and $f(w)$ represent the magnitude and phase components. $c(w)$ represents the unpredictability measure.

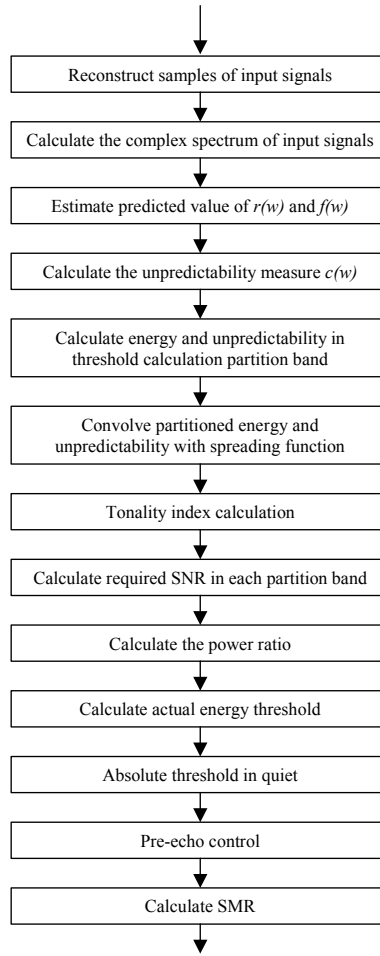


Figure 11: Flow chart of MPEG Psychoacoustic Model II.

2.3 Perceptual Entropy

Johnston [7][13] defined the perceptual entropy (PE) as a measure of perceptually relevant information contained in audio signals. PE combines the psychoacoustic masking with signal quantization principles to represent theoretical bits requirement for transparent audio coding. The signal is first windowed and transformed to the frequency domain. The masking threshold is then obtained by perceptual rules. Finally, the number of bits required to quantize the spectrum without injecting any perceptual difference with respect to the original signal is determined. The estimation process of PE is accomplished as follows:

First, it is assumed that the quantization noise σ_n associated with a uniform quantizer with step size Δ is given by

$$\sigma_n^2 = \frac{\Delta^2}{12}. \quad (4)$$

Then the masking power per spectral line is calculated as

$$\frac{T_i}{k_i}, \quad (5)$$

where T_i is the psychoacoustic masking threshold and k_i is the number of spectral lines in critical band i . Since the real and imaginary parts of the spectrum are quantized independently, the energy at each frequency must be divided by 2. Hence the masking power per real and imaginary components is

$$\frac{T_i}{2k_i}. \quad (6)$$

Next we want that the quantization noise per component can be lower than the masking capacity. It means

$$\frac{\Delta_i^2}{12} \leq \frac{T_i}{2 * k_i}, \quad (7)$$

and the step size Δ_i is derived from

$$\Delta_i \leq \left(\frac{6 * T_i}{k_i} \right)^{\frac{1}{2}}. \quad (8)$$

Now, the quantizer levels N_i to represent quantized spectral lines are determined by

$$N_i^{\text{Re}}(\omega) = \left| \text{nint} \left(\frac{\text{Re}(\omega)}{\Delta_i} \right) \right|, \quad (9)$$

and

$$N_i^{\text{Im}}(\omega) = \left| \text{nint} \left(\frac{\text{Im}(\omega)}{\Delta_i} \right) \right|, \quad (10)$$

where ω represents each spectral line in critical band i , nint is a function that returns the nearest integer to its argument, and $| |$ denotes the absolute value function. With the level information, the number of bits required for per band i are

$$b_i^{\text{Re}}(\omega) = \log_2(2 * N_i^{\text{Re}}(\omega) + 1), \quad (11)$$

and

$$b_i^{\text{Im}}(\omega) = \log_2(2 * N_i^{\text{Im}}(\omega) + 1). \quad (12)$$

Finally, the PE per band can be estimated as

$$PE_i = \sum_{\omega=bl_i}^{bh_i} (b_i^{\text{Re}} + b_i^{\text{Im}}) \quad (13)$$

$$= \sum_{\omega=bl_i}^{bh_i} \log_2 \left(2 * \left\lfloor n \operatorname{int} \left(\frac{\operatorname{Re}(\omega)}{\sqrt{6 * T_i / k_i}} \right) \right\rfloor + 1 \right) + \log_2 \left(2 * \left\lfloor n \operatorname{int} \left(\frac{\operatorname{Im}(\omega)}{\sqrt{6 * T_i / k_i}} \right) \right\rfloor + 1 \right), \quad (14)$$

where bl_i and bh_i are lower and upper bounds of band i .



Chapter 3

Bit Reservoir Design in Current Audio Codec

This chapter reviews the bit reservoir schemes in current codec standards and audio encoders to illustrate the fundamentals of bit reservoir design and to provide reference materials for comparing with our proposed design.

3.1 Bit Reservoir Schemes in MP3 Codec

MPEG-1 Layer III (MP3) uses a so-called bit reservoir to make up for temporary shortage of bits required for encoding. If a frame is easy to code then the some of the bits can be saved in the reservoir. If a frame is difficult to compress, some additional bits can be allocated to the frame. Through this mechanism, the reservoir can control the audio frame quality without direct limitation to the bit regulation.

3.1.1 Frame Format

The bitstream format of MP3 [1] is shown in Figure 12. The length of Header is always 4 bytes long; the length of side information is 17 bytes in mono mode and 32 bytes in stereo mode. The distance between adjacent frame headers is determined by *bit_rate_index* in Header. Once the value of *bit_rate_index* in current frame is extracted, the position of next frame header can be derived immediately.

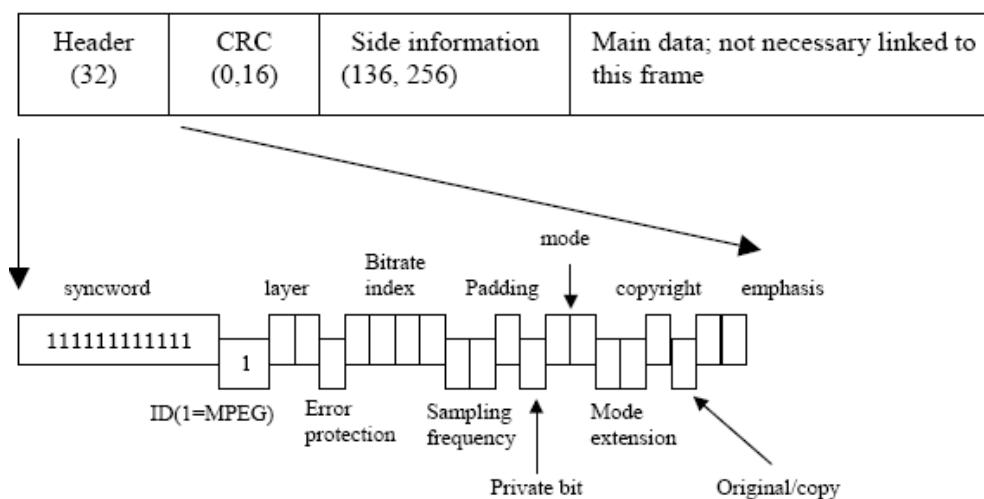


Figure 12: Bitstream format of MPEG-1 Layer III [1].

The bit reservoir technique is realized by the dynamic data allocation of MP3. It is implemented by a 9-bit pointer (*main_data_begin*), which indicates the location of the starting byte of the audio data (*main_data*) for that frame, in side information. The frame structure is shown in Figure 13. With the assistance of *main_data_begin* pointer, the Frame 3 with much more demand can utilize the space of Frame 2 to achieve better quality without breaking bit rate restriction.

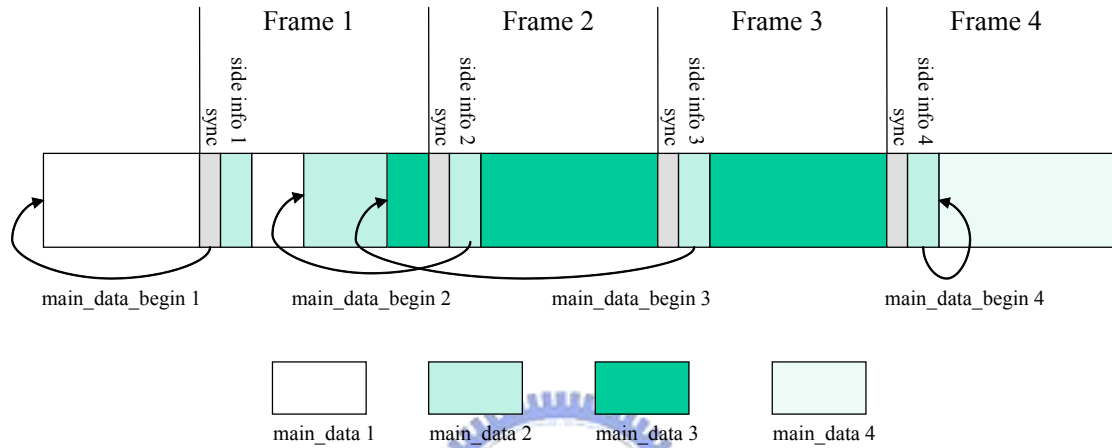


Figure 13: Example of MP3 frame structure [1].

The *main_data_begin* pointer also determines the maximum bit reservoir size in MP3. The maximum amount of accumulated bits is calculated by

$$\text{Maximum bit reservoir size} = 8 * 2^9 = 4096 \text{ (bits)} \quad (15)$$

The exceed bits will be padded to maintain bit rate constraint while the bit reservoir overflows.

3.1.2 Bit Rate Scenario

On the preferred scenario, there are three cases: the constant bit rate (CBR), the variable bit rate (VBR), and the average bit rate (ABR). Coding at a constant perceptual quality generally leads to a variable rate coding like VBR. The VBR in general has no regulation on bit rates. Conversely, coding at a constant bit rate like CBR will usually result in a time-dependent coding quality depending on segments of input signal. The CBR allows very limited bit variation in consecutive frames. Both concepts can be combined advantageously by using ABR. ABR allows the constant bit rate over a time period longer than several time-frames and attains constant output quality over times. Therefore, our efficient bit reservoir design for MP3 presented in this thesis mainly focuses on CBR mode and ABR mode.

The implementation of CBR mode in MP3 is achieved by fixed *bit_rate_index* setting. Hence the maximum size of bit reservoir is 4096 bits as calculate in (15). In ABR mode, the *bit_rate_index* can vary between frames as long as the bit rate on average over a long time satisfies the desired bit rate request. For this reason, the limitation of physical bit reservoir size can be freed. An abstract bit reservoir with larger size, which is implemented by adjusting *bit_rate_index* dynamically, is used to regulate bit rate and obtain better audio quality. The details are shown in later chapters in this thesis.

3.1.3 Recommended Scheme in Standard

The MPEG draft [1] recommends a scheme for bit reservoir design. The numbers of bits, which are made available for the *main_data* are derived from the actual estimated threshold (the PE as calculated by the psychoacoustic model), the average number of bits (*mean_bits*) and the actual content of the bit reservoir. The number of bytes in the bit reservoir is given by *main_data_begin*. The actual rules for controlling bit reservoir in [1] are given in Figure 14.

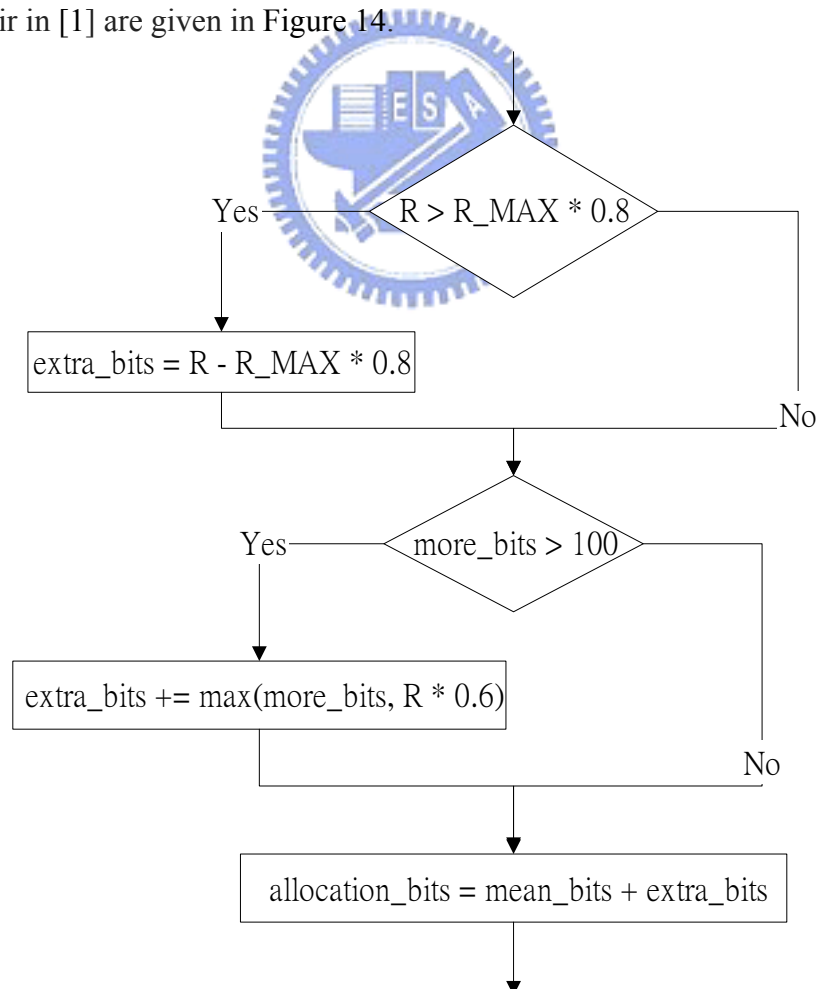


Figure 14: The flow chart of recommended bit reservoir control in [1].

The variables in Figure 14 are defined as follow:

R	current reservoir size;
R_MAX	maximum allowable reservoir size;
$more_bits$	derived from
	$more_bits = 3.1 * PE - mean_bits ;$ (16)
PE	perceptual entropy calculated by the psychoacoustic model;
$max()$	function that return the maximum value between arguments;
$mean_bits$	average bits of one frame derived from desired bit rate;
$allocation_bits$	allocated bits for quantization.

After the actual loops computations have been completed, the number of bytes not used for *main_data* is added to the bit reservoir. If the number of bytes accumulated in the bit reservoir exceeds the maximum allowable content, stuffing bits are written to the bitstream and the content of the bit reservoir is adjusted accordingly. This scheme only considers the reservoir size variation (budget size) and roughly estimates the demand. It may encounter quality risks while applying to different bit rate request.

3.1.4 Scheme in LAME 3.88

LAME 3.88 [14] is currently a popular MP3 encoder. The bit reservoir control scheme in LAME 3.88 is depicted in Figure 15. The variables in Figure 15 are defined as follows:

R	current reservoir size;
R_MAX	maximum allowable reservoir size;
PE	perceptual entropy calculated by the psychoacoustic model;
ch	channel index;
$mean_bits$	average bits of one frame derived from desired bit rate;
add_bits	temporary variable in reservoir control steps.

The allocated bits per channel, $allocation_bits[ch]$, is derived by

$$allocation_bits[ch] = mean_bits[ch] + add_bits[ch] . \quad (17)$$

In the mechanism, there are some heuristic number like 750 and 1.4, this number will lead to variation on the different bit rates and sample rates. Furthermore, there is no deposit mechanism for bit reservoir in addition to the silence frames. Hence in most situations, there is no bit deposited in the reservoir to regulate the bit demand from critical audio frames.

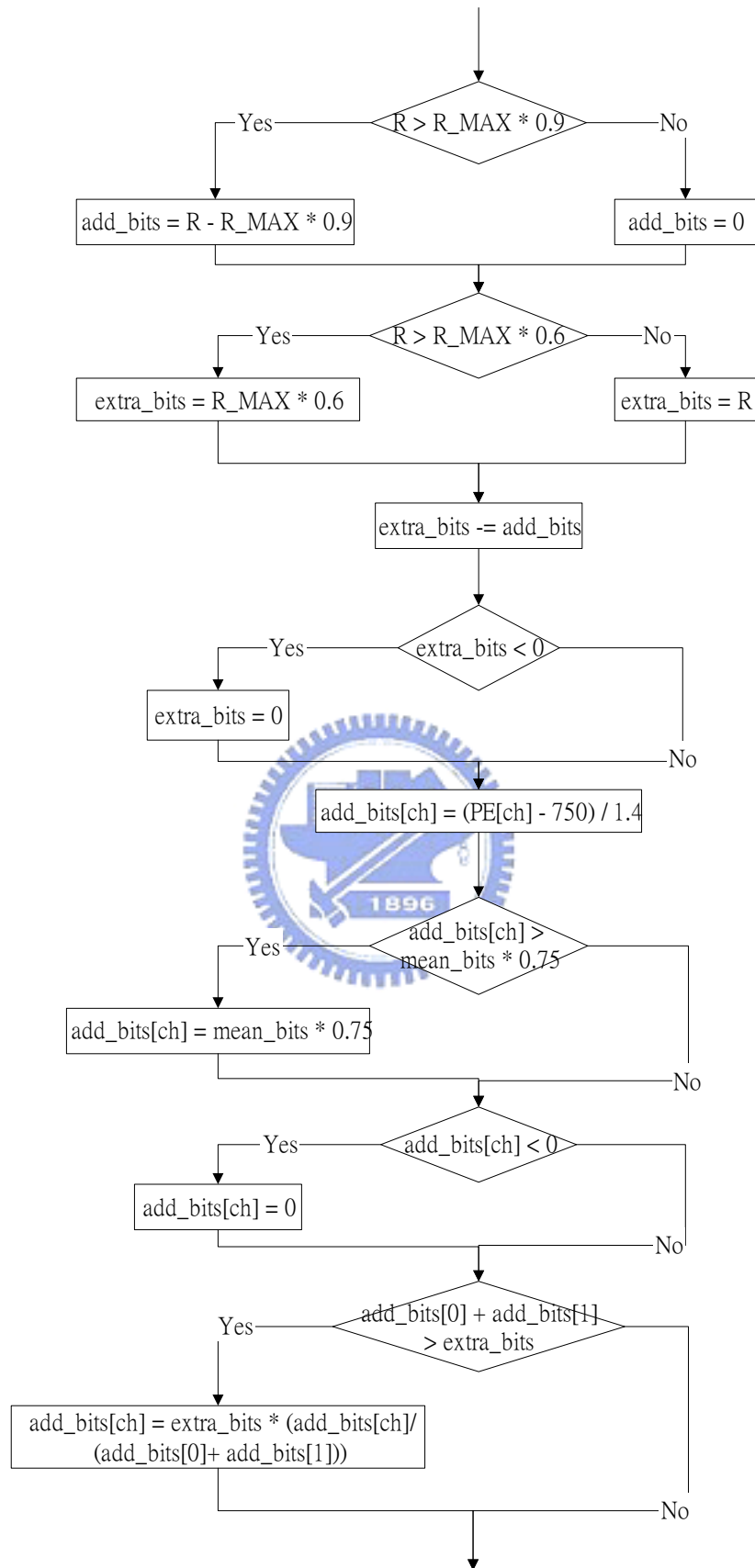


Figure 15: The flow chart of bit reservoir design in LAME 3.88.

3.2 Bit Reservoir Schemes in AAC

AAC follows the same basic paradigm as MP3 (high frequency resolution filterbank, non-uniform quantization, Huffman coding), but includes a lot of new coding tools to improve the coding efficiency. Also, with the flexible bitstream format, AAC can achieve a high bits variation among frames to control the frame quality and various bit rate scenario.

3.2.1 Bitstream Format

The MPEG-4 AAC [2] system has a very flexible bitstream syntax. There are two parts in AAC bitstream: Audio Data Interchange Format (ADIF) and Audio Data Transport Stream (ADTS). The type of each audio format is shown in Figure 16 and Figure 17. ADIF is not used in the on-line network environment because that is the storage mass, so playing is impossible at some point of bitstream. ADTS is possible playing at the network environment because there are syncword, CRC and frame length information in bitsream. Therefore, we focus on ADTS format as our research domain.

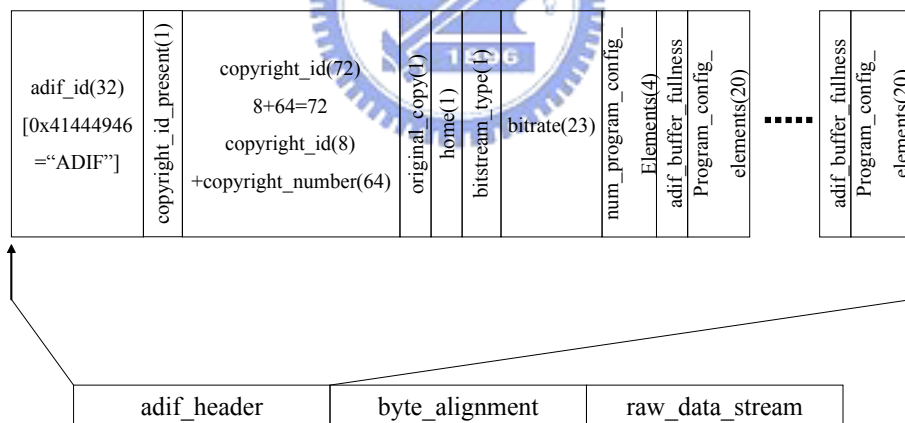


Figure 16: ADIF birstream [2].

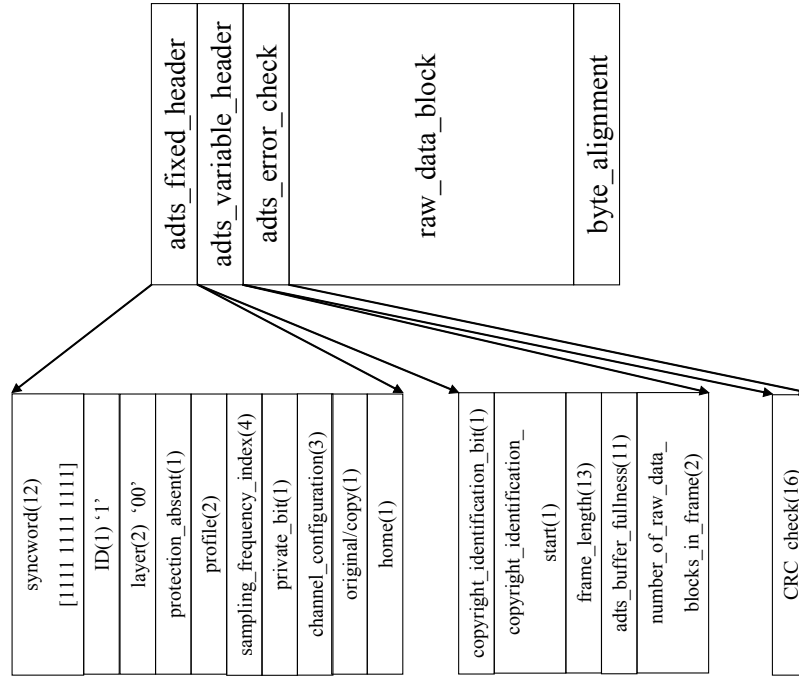


Figure 17: ADTS bitstream [2].

Since the AAC system has a data buffer that permits its instantaneous data rate to vary as required by the audio signal, the length of each frame is not constant. In this respect the AAC bitstream uses a variable rate headers that are byte-aligned so as to permit editing of bitstreams at any frame boundary. It is the main difference with MP3 discussing in section 3.1.1. Because the distance between headers is no longer fixed, the bitstream element, syncword, is used for searching header position. The syncword is composed of bit string '1111 1111 1111.' Once the decoder meets up syncword in bitstream, the position of header is derived quickly for decoding process.

The maximum bit reservoir size in AAC is limited by decoder input buffer size while in MP3 is determined by *main_data_begin* pointer. The maximum bit reservoir size for constant rate can be calculated by subtracting the mean number of bits from the minimum decoder input buffer size. Such as

$$max_bit_reservoir = max_channel_bits * 2 - mean_bits \quad (18)$$

where *max_bit_reservoir* is the maximum bit reservoir size, *max_channel_bits* is the maximum number of bits per channel that is 6144, and *mean_bits* is the average number of bits per frame. For variable bit rate the encoder must operate in a way that the input buffer requirements do not exceed the minimum decoder input buffer.

The limitation of bit reservoir size described above is used to make sure the accuracy of decoding process. As long as the input buffer requirement is maintained, the bit reservoir concept can be extended to an abstract bit reservoir design. The

reservoir control not only implies on short-term frames but also implies on whole track for bit rate control. The details are shown in later chapters in this thesis.

3.2.2 Recommended Schemes in Standard

The paper [2] provides a rough bit reservoir control method. Bits are saved to the bit reservoir when the bits are fewer than the *mean_bits* of one frame. If the reservoir is full, unused bits have to be encoded in the bitstream as fill-bits. The maximum amount of bits available for a frame is the sum of *mean_bits* and bits saved in the bit reservoir. The number of bits that should be used for encoding a frame depends on the maximum available bits and *more_bits* value, which is calculated by

$$\text{more_bits} = \text{bit_allocation} - (\text{mean_bits} - \text{side_info_bits}). \quad (19)$$

where *bit_allocation* is derived from psychoacoustic model, *side_info_bits* is bits used for side information. The actual rule is given in Figure 18.

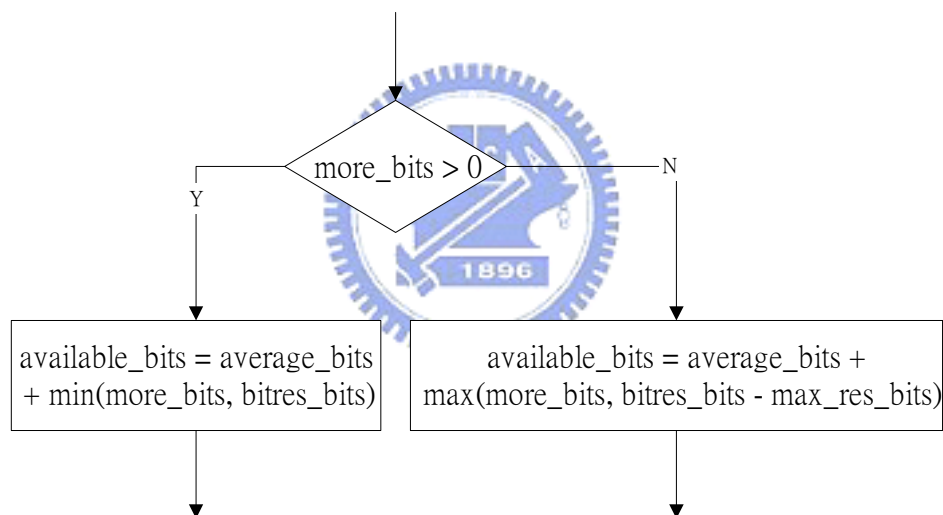


Figure 18: Flow chart of bit reservoir control in [2].

The variables in Figure 18 are defined as follow:

- available_bits* bits used for quantization;
- average_bits* the mean number of bits per frame derived from desired bit rate;
- min()* function that returns the minimum value within arguments;
- max()* function that returns the maximum value within arguments;
- bitres_bits* bits saved in bit reservoir;
- max_res_bits* maximum bit reservoir size calculated by (18).

3.2.3 Schemes in FAAC 1.24

FAAC 1.24 [15] is the reference encoder for AAC. Through our coding trace it

seems to have no bit reservoir control design in FAAC except the check of input buffer restriction. The only relative part is the function “MaxBitresSize” that is used for stuffing. Hence we have no any other discussion about FAAC here.

3.3 Bit Reservoir Schemes in HE-AAC

HE-AAC [3] is the combination of conventional AAC codec and Spectral Band Replication (SBR) module to achieve high audio quality at low bit rates. There are no definite statements concerning bit reservoir design for HE-AAC in [3]. Current limited literature seems to have no relative bit reservoir design in HE-AAC. We propose a novel design [16] and state it explicitly in this thesis.

3.3.1 Bitstream Overview

The bitstream of HE-AAC shown in Figure 19 is form by AAC frame and SBR frame. The decoder will identify the content of bitstream element to determine whether it is synword for next AAC frame header or header flag for SBR frame. The decoding process of SBR part is independent of AAC part. Hence the bits distribution between AAC and SBR in HE-AAC frame leads to dependent issue. How to allocate bits properly would affect the audio quality and bit rate control.

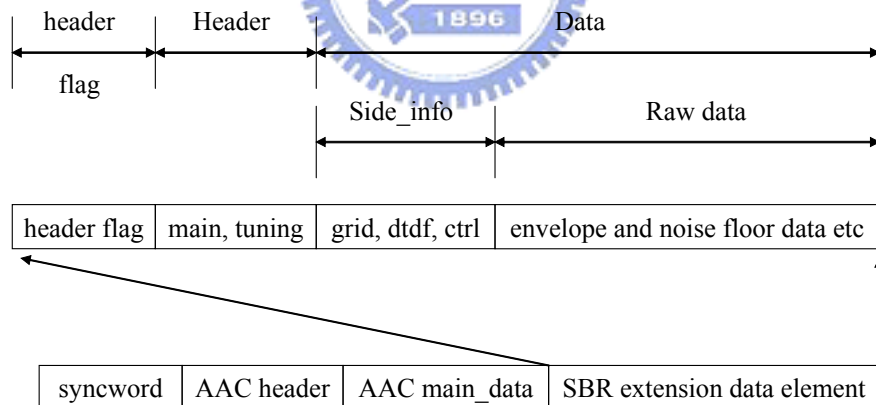


Figure 19: Bitstream organization of HE-AAC [3].

3.3.2 Schemes in 3GPP

The 3rd Generation Partnership Project (3GPP) provides a reference code for HE-AAC [17]. The flow chart of 3GPP is given in Figure 20. There is a bit reservoir but no bit reservoir control in 3GPP. The PE Correction and Bit Factor calculation are only used for available bits estimation without flexible bit rate control. If the available bits are not sufficient to support estimated demand, the masking threshold is

appropriately adjusted. The masking threshold to be adjusted is different from the one calculated by psychoacoustic model. The psychoacoustic model calculates the idea masking for human hearing. The adjusted masking is used to estimate the scale factor for quantization later. So the available bits for AAC part are derived from *Usable_bits*, which is calculated by subtracting SBR bits from mean bits of HE-AAC, and *Reservoir_bits*. Although this strategy in 3GPP maintains bit rate constrain but comes up against quality degradation in AAC part.

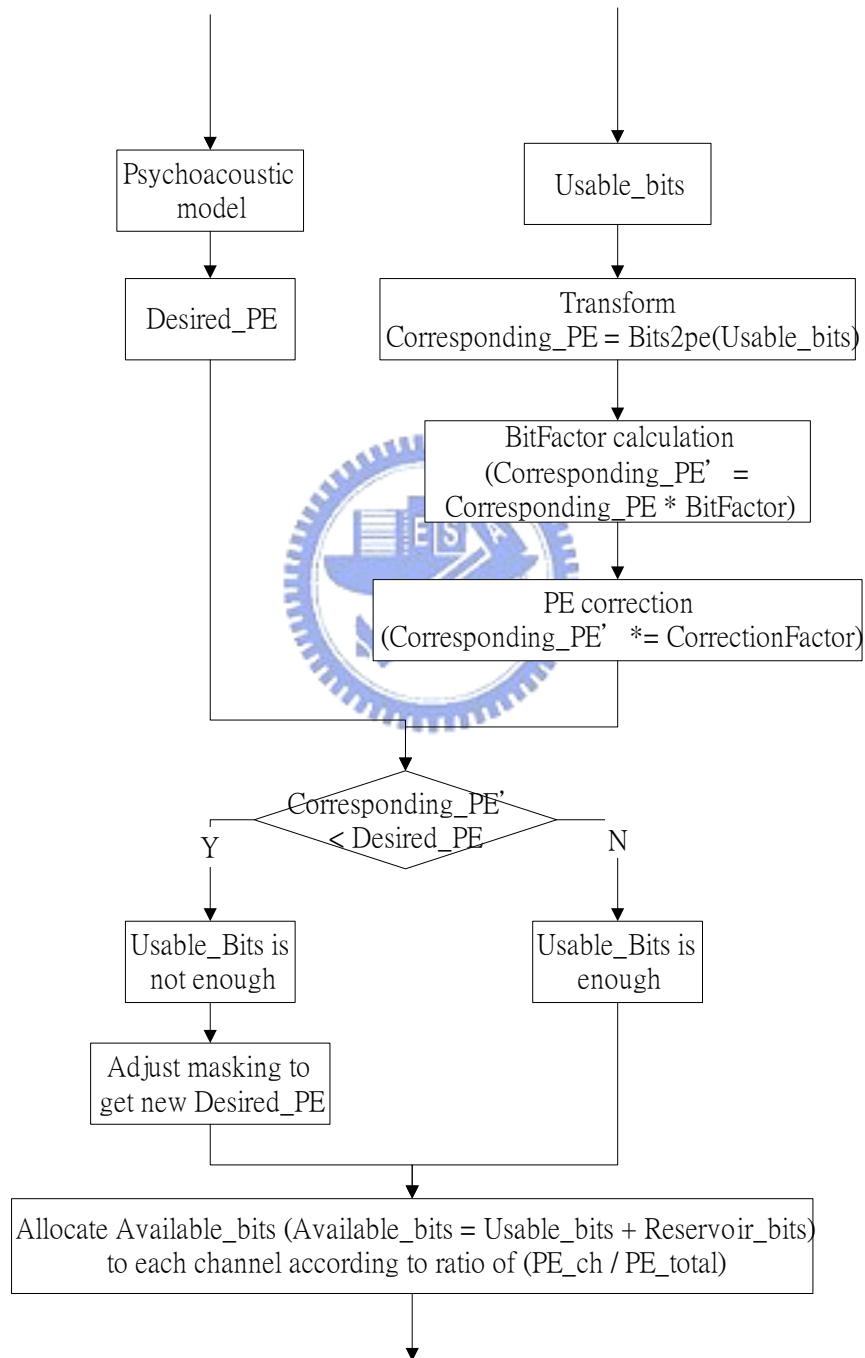


Figure 20: Flow chart of 3GPP HE-AAC.

Chapter 4

Efficient Bit Reservoir for MP3 and AAC

The bit reservoir deposits bits from “easy” frames and loans bits for “difficult” frames. The efficiency of bit reservoir depends on the accuracy to predict the demand bits with consideration to the bit rates, accumulated bits, audio contents, allowable bit-variation range, allowable bit-variation resolution, and tools used in the encoder. These factors are jointly considered through two modules: demand estimator and budget regulator. The demand estimator will take care of the factors like bit rates and bit-quality dynamics. Based on the demand bits, the budget regulator will decide the available bits from the tools used and the allowable range or resolution of bit variation. As mentioned in Chapter 3, there is limited information on the detailed bit reservoir implementation in current literature. However, the basic approach is to predict demand bits through the perceptual entropy and budget control through the fullness of bit reservoir. The variation with the bit rates and the statistics of the quality over frames is usually not well considered. On the other budget part, the loan range and the resolution are two spaces that can be extended in addition to the fullness control. Therefore, this chapter will present an efficient bit reservoir design based on the demand estimator and the budget regulator for MP3 and AAC.

4.1 Allocation Entropy

Johnston [7][13] defined the perceptual entropy (PE) to reflect the minimum bits required for transparent quality. The PE is defined as

$$PE_i = W_i * \log_{10}(SMR_i + 1), \quad (20)$$

where W_i is the number of spectrum lines in partition band i . The partition band has a bandwidth proportional to the critical bandwidth. The signal-to-masking ratio SMR_i is defined as

$$SMR_i = \begin{cases} E_i / M_i, & \text{if } (E_i > M_i) \\ 0, & \text{if } (E_i \leq M_i) \end{cases} \quad (21)$$

where E_i and M_i are the spectrum energy and masking threshold in partition band i .

Except the general perceptual entropy, there is another perceptual criterion, allocation entropy (AE) [18]. The PE does not reflect the bits required for the case where the transparent quality is not achievable under limited bit rates. For audio coding, the main issue is the tradeoff between bits-constrain and quality instead of achieving transparent quality. The AE could well reflect the bits required to have the graceful degradation and have put into consideration the bandwidth proportional noise-shaping criterion [19]. To derive the AE, we modify the SMR_i in (21) as SMR'_q :

$$SMR'_q = \begin{cases} \frac{E_q}{M_q * B_q}, & \text{if } (E_q \geq M_q * B_q) \\ 0, & \text{if } (E_q < M_q * B_q) \end{cases}, \quad (22)$$

where q is the index of quantization band and the effective bandwidth B_q defined in [19] is illustrated in Figure 21 and Figure 22. Therefore, the definition of AE in each band can be described as

$$AE_q = W_q * \log_{10}(SMR'_q + 1), \quad (23)$$

where W_q is the number of spectrum lines in quantization band q .

If the noise is higher than masking threshold, a noise proportional to effective bandwidth is the suitable bit allocation criterion for optimum graceful degradation according to [19]. The effective bandwidth is derived from the critical band with bandwidth about one-third to one-fourth of the critical bandwidth. In general, the higher spectrum bands often have wider effective bandwidth and should have a higher noise shape. Also, the AE is evaluated with the unit of the scale factor bands to match directly the units in quantization and encoding process.

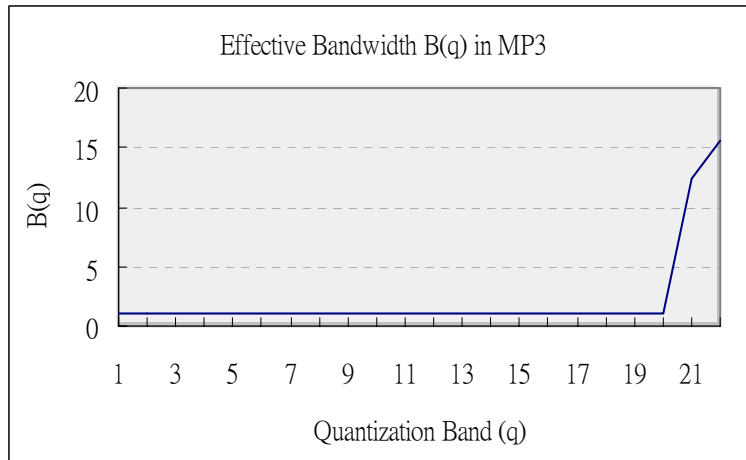


Figure 21: Effective bandwidth for MP3 (Long Window, Sample Rate: 44.1 KHz) [19].

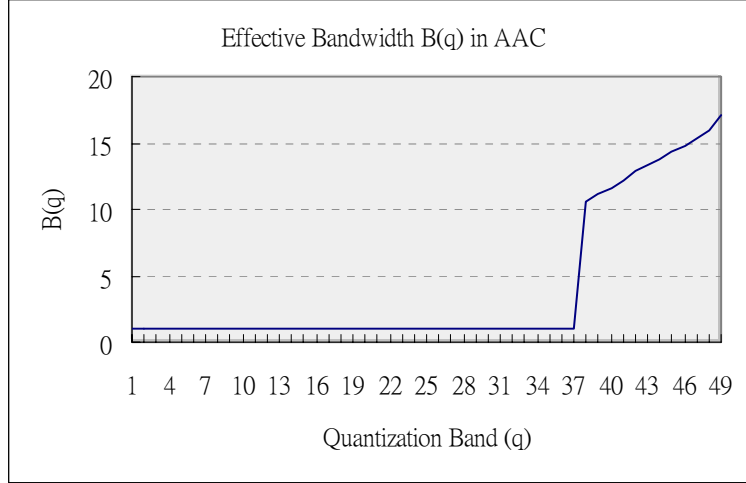


Figure 22: Effective bandwidth for AAC (Long Window, Sample Rate: 44.1 KHz) [19].

Since the granule in MP3 and the frame in AAC is the basic coding unit, we need to sum up the values of AE in each quantization band. The bits required for either the granule or the frame can be obtained from

$$AE(f) = \sum_{ch} \sum_q AE_{ch,q} = \sum_{ch} \sum_q W_{ch,q} * \log_{10}(SMR'_{ch,q} + 1), \quad (24)$$

where ch is the channel index, q is the quantization band index, $W_{ch,q}$ is the number of spectrum lines in quantization band q of channel ch , $SMR'_{ch,q}$ is the signal-to-masking ratio as shown in (22) for quantization band q of channel ch , and f is the frame or granule index. $AE(f)$ represents the bits required for the frame to have a specified quality.

In order to illustrate that AE is the representative of bits required for variable signals, we list some tracks with its corresponding spectrogram and AE as examples. These tracks shown in Figure 23 to Figure 26 are chosen from Table 2. From the results in Figure 23-Figure 26, the AE definitely indicates the transient of signals. Based on this information, the demand estimator design in later section is appropriate to evaluate the bits required for each frame.

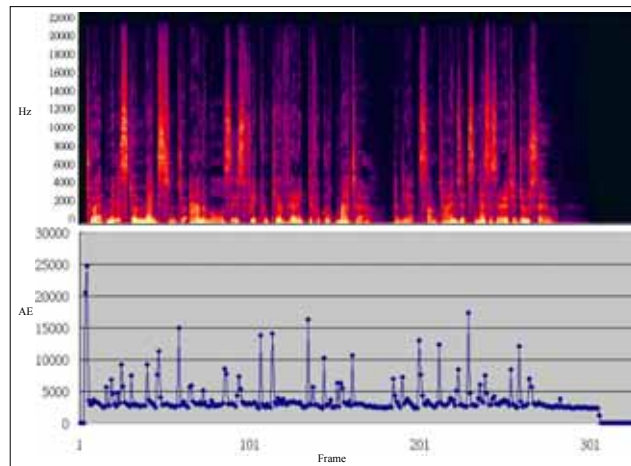


Figure 23: The spectrogram (top) and the values of AE (bottom) of natural vocal (es03).

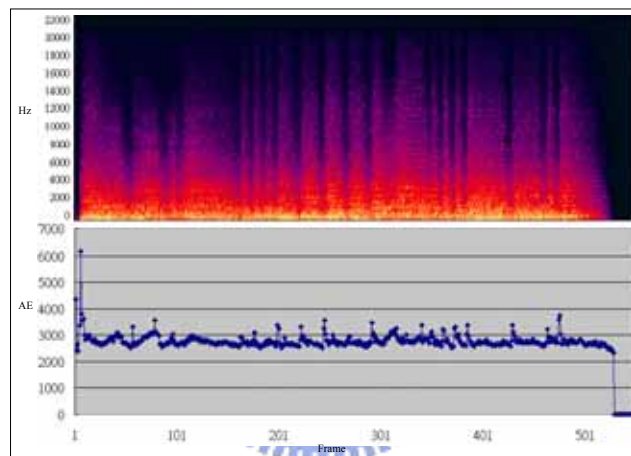


Figure 24: The spectrogram (top) and the values of AE (bottom) of complex sound (sc02).

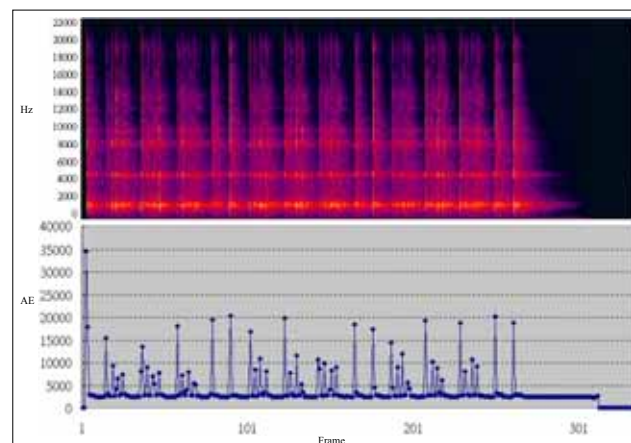


Figure 25: The spectrogram (top) and the values of AE (bottom) of transient (si02).

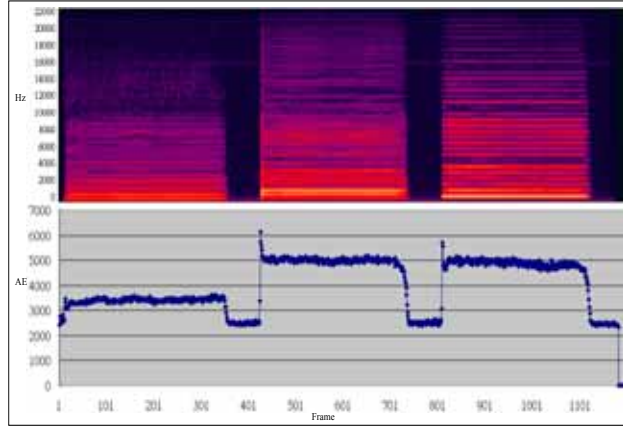


Figure 26: The spectrogram (top) and the values of AE (bottom) of harmonic (si03).

4.2 Demand Estimator

The responsibility for demand estimator is to actively allocate bits among frames instead of passively consumes accumulated bits. The demand estimator adaptively predicts the bits required for a frame according to the audio content. The proposed idea based on the allocation entropy is to predict demand bits through average in the past and demand ratio calculation. The demand estimator will also take care of the factors such as bit rates and bit-quality dynamics.

4.2.1 AE Average

In addition to AE, which represents the requirement for a frame, we should have the average demands aligned to the average bit rates to control the average quality. The average demand $AE_{average}$ can be estimated through the average over past N frames:

$$AE_{average} = \frac{\sum_{f=1}^N AE(f)}{N}, \quad \text{for } LBound < AE(f) < UBound, \quad (25)$$

where $LBound$ is the lower bound of AE and $UBound$ is the upper bound of AE. The boundary constraint comes from the definition of masking threshold in SMR'_q calculation as shown in (22). The masking threshold M_q for quantization band q is defined as

$$M_q = \max(qthr_q, \min(T_q, T_{l_q} * repelev)), \quad (26)$$

where $qthr_q$ is the threshold in quiet, T_q is the masking threshold of band q , T_{l_q} is the masking threshold of band q in the last block, and $repelev$ is set to '1' for short blocks and '2' for long blocks. If the $(K-1)^{th}$ signal is like quiet sound and the K^{th} signal is a

strong attack signal, the M_q of the K^{th} signal is the small value $T_{l_q} * repelev$, not T_q . So the corresponding AE of the K^{th} signal with strong energy will be extremely large. This kind of AE only indicates the occurrence of transient but the actual bits demand is drowned in that enormous value. In order to filter out this type of interference in $AE_{average}$ calculation, the $AE(f)$ larger than $UBound$ should not be put into reference frames. The similar reason is applied on $AE(f)$ smaller than $LBound$ to avoid the influence from silence frames. The flow chart of detailed $AE_{average}$ calculation processes is depicted in Figure 27. Num_AE denotes the number of processed frames. AE_{cur} denotes the AE of current frame. $Reference_Array$ denotes the temporary array that stores the AEs of past frames.

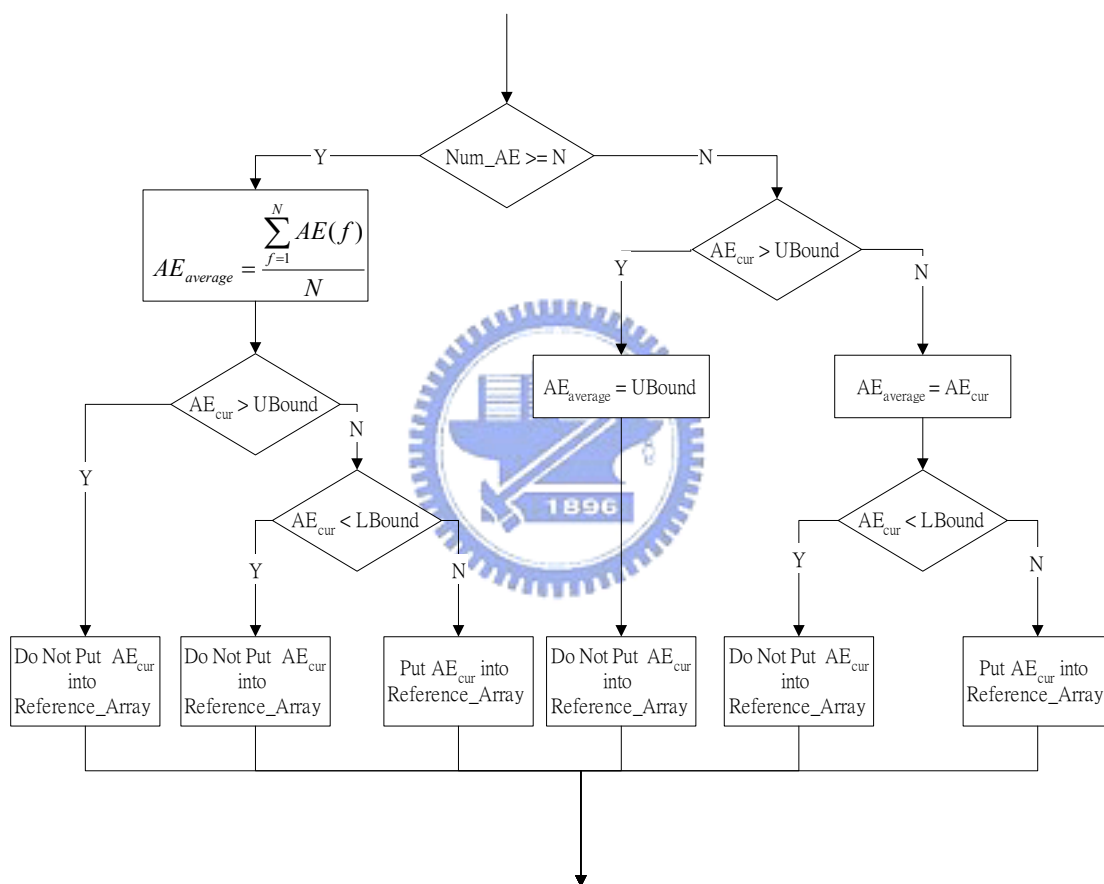


Figure 27: Flow chart of $AE_{average}$ calculation.

4.2.2 Demand Ratio

Through the average AE, we could evaluate the demand ratio D to represent the trend of demand variation.

$$D(f) = \frac{AE(f) - AE_{average}}{AE_{average}}. \quad (27)$$

$D(f)$ represents the current demand over the previous N coding units. The demand ratio $D(f)$ should be transformed into $R_{demand}(f)$ by a transform function to shape the curve and clip the upper/lower bounds:

$$R_{demand}(f) = \eta(D(f)). \quad (28)$$

The three $\eta(\cdot)$ examples used in MP3 [20] and AAC [21] encoders are illustrated in Figure 28 to Figure 30.

4.2.2.1 Demand Curve for MP3 CBR Mode

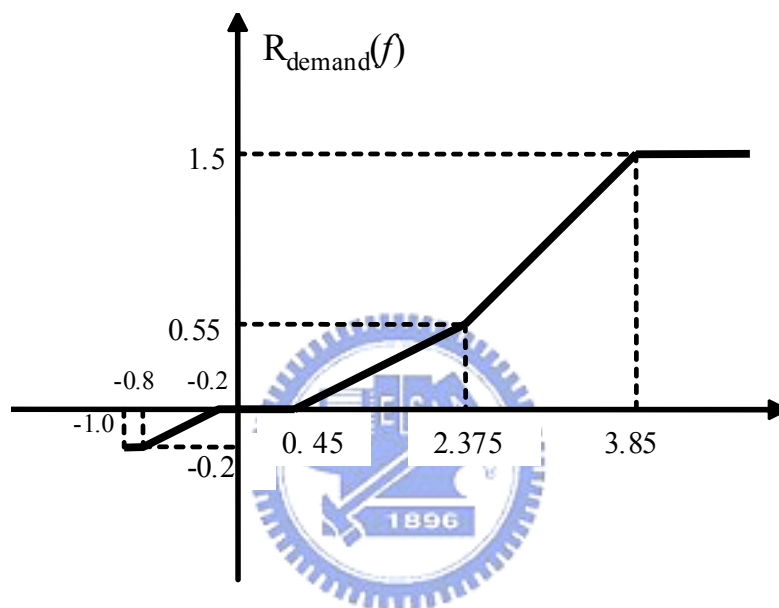


Figure 28: Demand curve for MP3 CBR mode.

Figure 28 illustrates the transform function of demand ratio for MP3 CBR mode. The slope is small due to the limited explicit budget buffer size. The “zero-zone” from -0.2 to 0.45 in $D(f)$ is used to neglect the slight demand variation come from the prediction accuracy based on AE and $AE_{average}$. The saturation value of $R_{demand}(f)$ is set to 1.5 with referring to the ratio of maximum bit rate (320 kbps) to reference bit rate (128 kbps). The lower bound of $R_{demand}(f)$ is set to -0.2 to prevent quality degradation caused by estimation error.

4.2.2.2 Demand Curve for MP3 ABR Mode

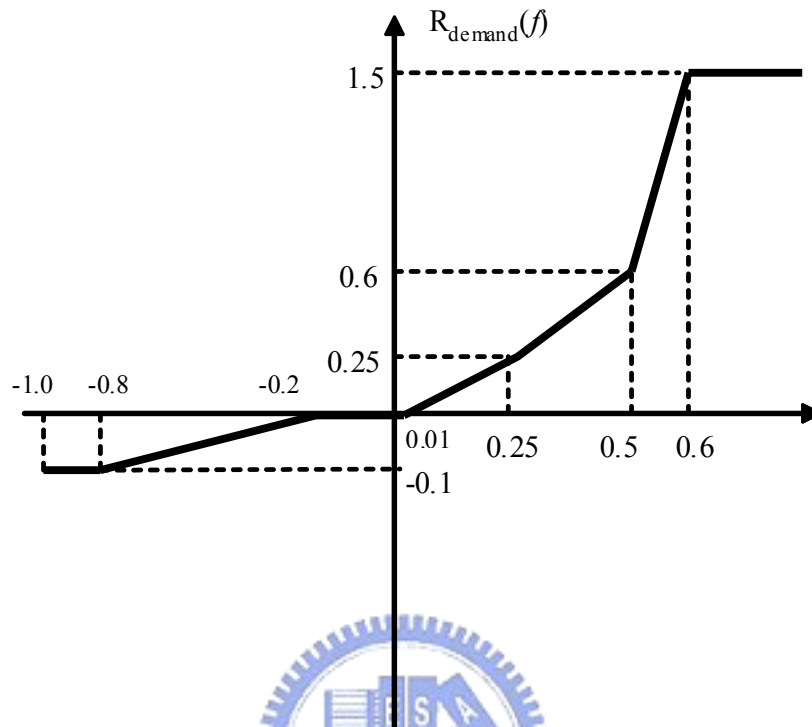


Figure 29: Demand curve for MP3 ABR mode.

Figure 29 illustrates the transform function of demand ratio for MP3 ABR mode. ABR mode has an implicit bit reservoir in addition to the explicit bit reservoir. With such flexible reservoir budget condition, the slope could increase with demands growing. Also the zero-zone of $D(f)$ in negative part could be stretched to tolerate much more negative demands. The saturation value of $R_{demand}(f)$ is set to 1.5 by the same reason in CBR mode. The lower bound of $R_{demand}(f)$ could be set to -0.1 to reduce the refund rate because of the elastic budget space.

4.2.2.3 Demand Curve for AAC

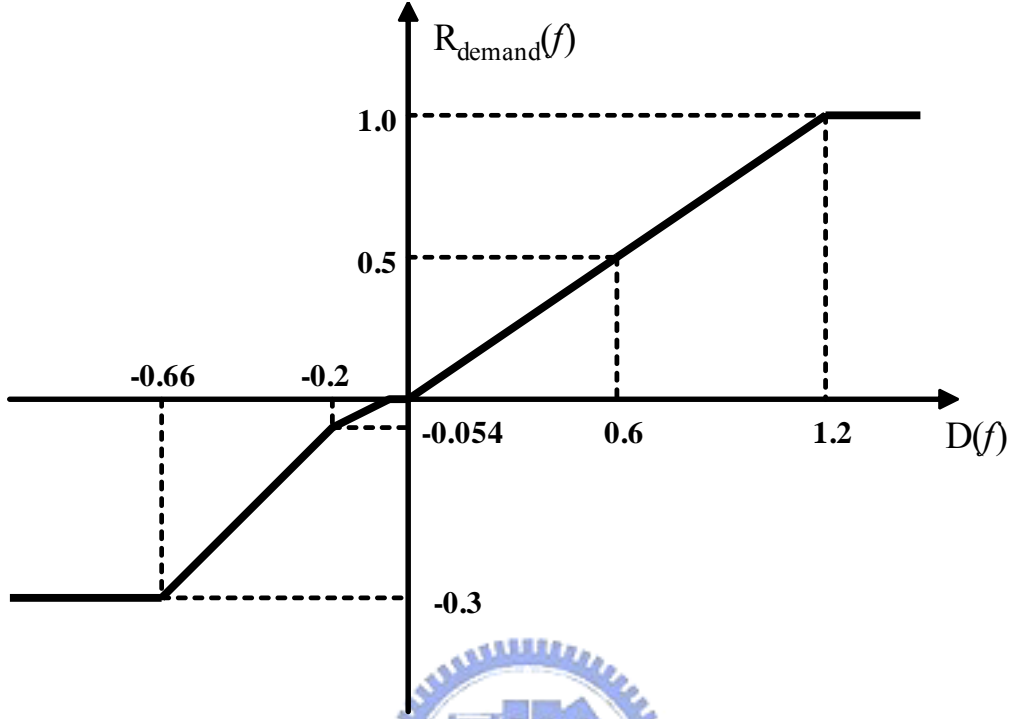


Figure 30: Demand curve for AAC ABR mode.

Figure 30 illustrates the transform function of demand ratio for AAC ABR mode. The slope is constant due to the intensive tools (e.g. M/S Coding [22], Window Switch [23], TNS [24]) used. Owing to these auxiliary modules the upper bound and lower bound setting of $R_{demand}(f)$ is more conservative.

4.3 Budget Regulator

The budget regulator decides the available bits according to the preferred scenario. We define a budget ratio and adjust the budget ratio with the fullness (denoted as F) of the bit reservoir. The variable F stands for the fullness of the bit reservoir budget. The fullness F is evaluated through

$$F = \frac{S}{S_{MAX}}, \quad (29)$$

where S is the current accumulated budget size, S_{MAX} is the maximum allowable bit reservoir size. S_{MAX} is defined as

$$S_{MAX} = mean_bits * B, \quad (30)$$

where $mean_bits$ is the average number of bits per frame in AAC or per granule in MP3, B is the control factor for determining budget buffer size. The concept of deposition and loan are used for flexible budget buffer design based on the nature of

ABR mode. It means that the budget buffer size is allowed to be negative. We can draw bits in advance and reimburse them in near future as long as conforming to the desired bit rate. Therefore, the eq. (29) could be redefined as

$$F = \begin{cases} 1, & \text{if } S > S_{MAX} \\ \frac{S}{S_{MAX}}, & \text{if } -S_{MAX} \leq S \leq S_{MAX} \\ -1, & \text{else} \end{cases} \quad (31)$$

The fullness F should be transformed into R_{budget} by a transform function to regulate the depositing or loaning rate:

$$R_{budget} = \varphi(F). \quad (32)$$

The three $\varphi(\cdot)$ examples used in MP3 [20] and AAC [21] encoder are illustrated in Figure 31 to Figure 33.

4.3.1 Budget Curve for MP3 CBR Mode

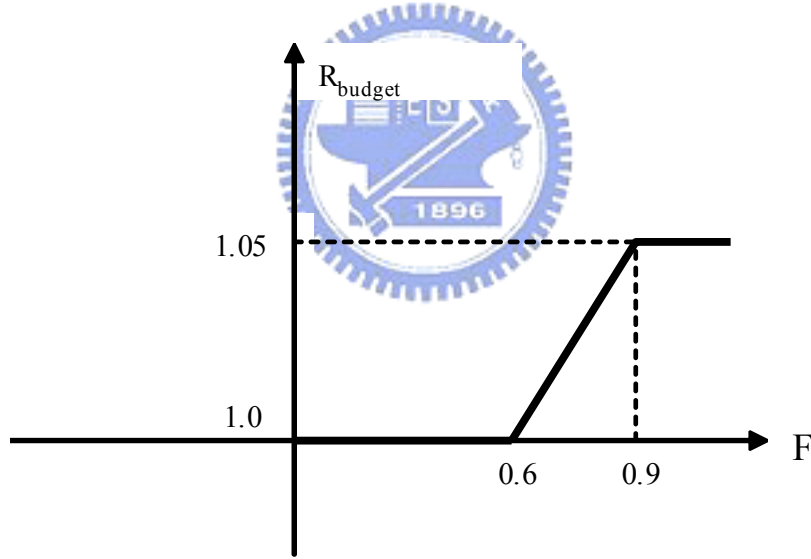


Figure 31: Budget curve for MP3 CBR mode.

Figure 31 illustrates the transform function of budget regulator for MP3 CBR mode. As described in section 3.1.1, the maximum bit reservoir size in MP3 CBR mode is limited by frame format. Since the maximum reservoir size is small and inflexible, the budget regulator only gives extra bonus for granules while reservoir budget is almost full. In other words the demand estimator plays a major role for bit reservoir controlling in MP3 CBR mode.

4.3.2 Budget Curve for MP3 ABR Mode

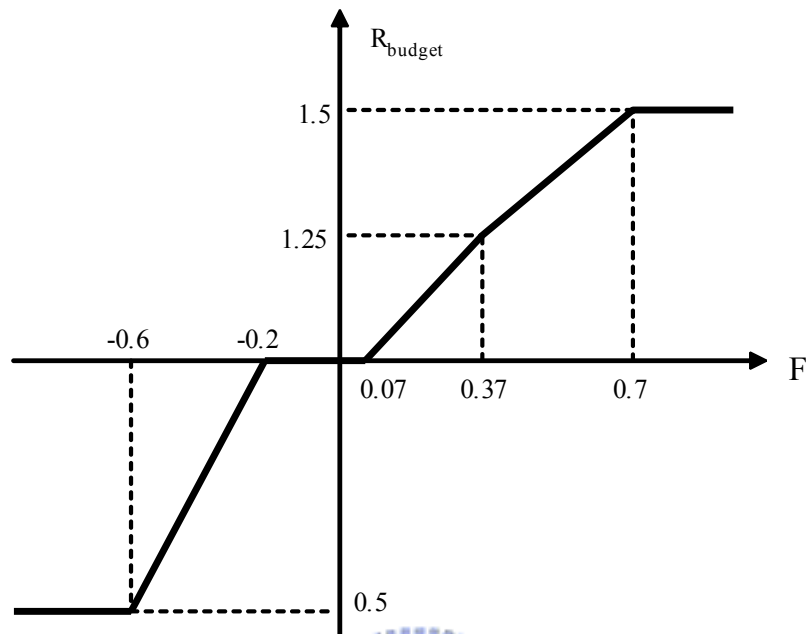


Figure 32: Budget curve for MP3 ABR mode.

Figure 32 illustrates the transform function of budget regulator for MP3 ABR mode. The “flat-zone” from -0.2 to 0.07 is used to eliminate the function of budget regulator while the bit reservoir is not full or deficient in order to prevent wasting budget or degrading quality. The increasing rate of R_{budget} slows down when the reservoir approaches full. The lower bound of R_{budget} is set to 0.5 to refund bits and maintain quality when the reservoir is almost exhausted.

4.3.3 Budget Curve for AAC

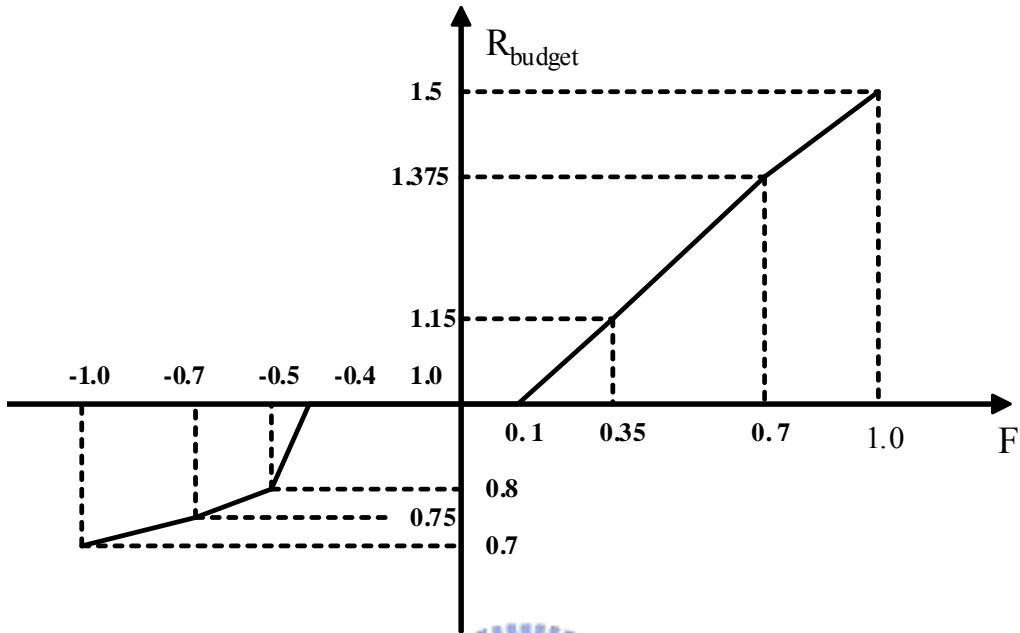


Figure 33: Budget curve for AAC ABR mode.

Figure 33 illustrates the transform function of budget regulator for AAC ABR mode. The trend is similar to that for MP3 ABR mode but with a little detailed due to adopt other auxiliary modules.

4.4 Allocated Bits Calculation

The proposed bit reservoir design not only manages accumulated bits but also determines allocated bits for each coding units based on novel concept of adaptive budget buffer size. Through the demand estimator and the budget regulator, the allocated bits for encoding units is derived from

$$Allocated_bits = mean_bits + R_{demand} * mean_bits * R_{budget}, \quad (33)$$

where $mean_bits$ is derived from the desired bit rate, R_{demand} comes from demand curve, and R_{budget} comes from budget curve. The $Allocated_bits$ is used for bit allocation and quantization later. The over estimated bits are reclaimed after quantization and feed into bit reservoir for next coding unit. There is an exception need to be noticed. If the R_{demand} is smaller than zero, the R_{budget} is revised to one to avoid undesired product of R_{demand} and R_{budget} . The flow chart of our efficient bit reservoir design for MP3 and AAC is shown in Figure 34.

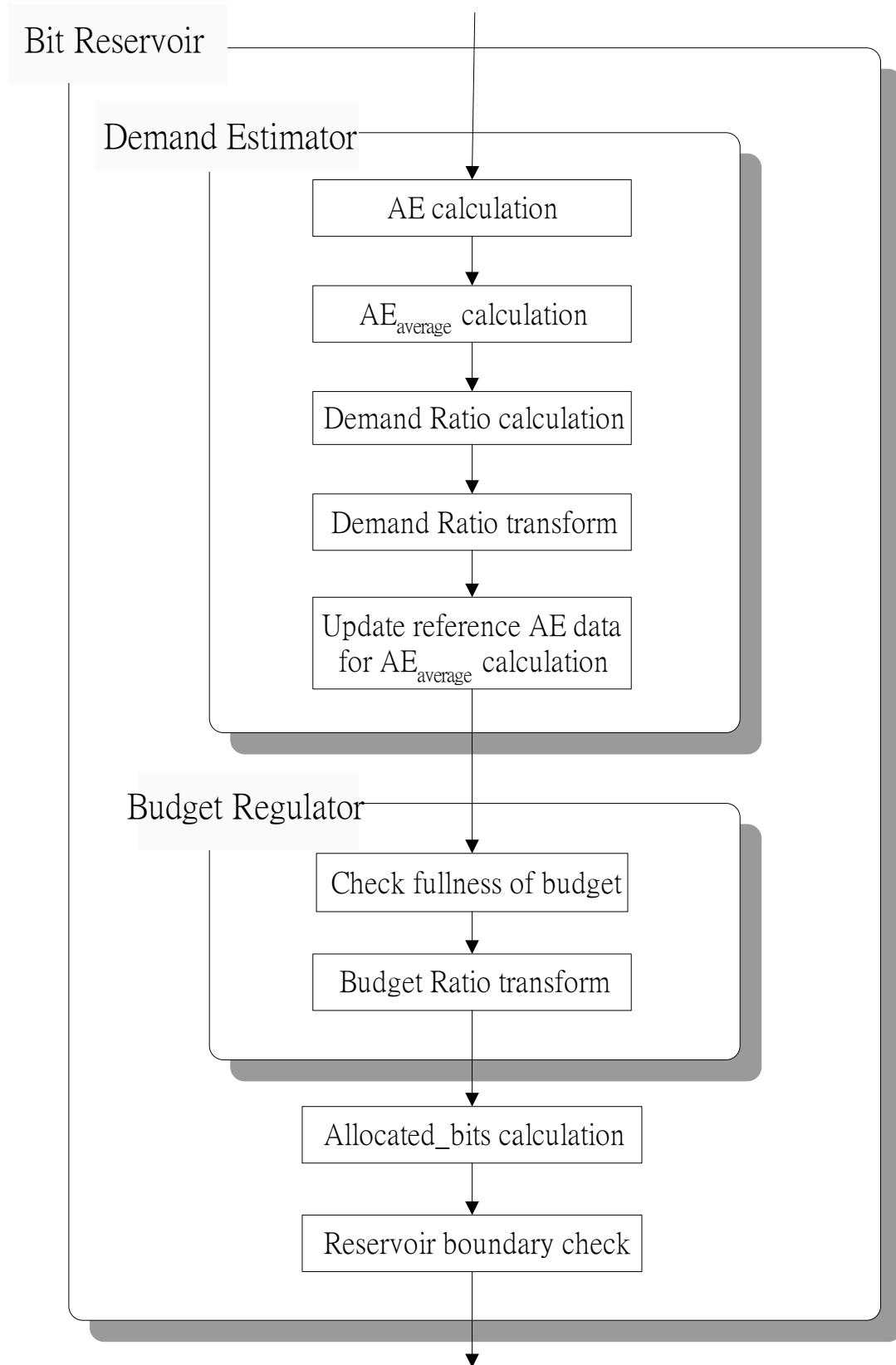


Figure 34: Flow chart of the efficient bit reservoir design.

4.5 Experiments for MP3 and AAC

This section focuses on quality measurement through objective and subjective methods. We adopt NCTU-MP3 [20] and NCTU-AAC [21] as platforms for MP3 and AAC to check the bit reservoir mechanisms. There are three primary experiments in this section. The first one is to evaluate the quality of two encoders with and without bit reservoir through objective measurement. The twelve test tracks recommended by MPEG are chosen as our default track testing database. The second one is the best parameters decision evaluations. The preceding sections are based on these well tuning parameters to evaluate the quality experiments. The third one is to prove the robustness of our proposed bit reservoir algorithm through PSPLAB audio database [26]. Through both objective and subjective tests, the efficiency and quality of our designs are well examined.

Table 2: The twelve test tracks recommended by MPEG.

Tracks		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	(b) (d)
5	sc02	Orchestral piece	Stereo	12	(d)
6	sc03	Contemporary pop music	Stereo	11	(d)
7	si01	Harpsichord	Stereo	7	(b)
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) (b)
12	sm03	Plucked strings	Stereo	13	(a) (b)
Remarks: (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.					

4.5.1 Objective quality evaluation

The aim of objective perceptual measurements is to predict the basic audio quality by using objective measurements incorporating psychoacoustic principles. For objective quality evaluation, we mainly adopt the PEAQ (Perceptual Evaluation of Audio Quality) system which is the recommendation scheme by ITU-R Tack Group 10/4. While PEAQ is based on a refinement of generally accepted psychoacoustic models, it also includes new cognitive components to account for higher-level processes that come to play a role in the judgment of audio quality. The objective difference grade (ODG) is the output variable from the objective measurement method via an artificial neural network. The ODG value should range from 0 to -4 , where 0 corresponds to an imperceptible impairment and -4 to impairment judged as very annoying. The improvement up to 0.1 is usually perceptually audible. 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 this PEAQ system [27].

We use NCTU-MP3 [20] as our code base for quality evaluation in MP3. Both CBR mode and ABR mode are conducted in following experiments. Other bit reservoir methods are also listed for comparison with our proposed design.

4.5.1.1 Objective quality evaluation for MP3 CBR mode

For CBR mode, there are another 4 bit reservoir methods for comparison. The first one is NoReservoir, which just uses the allocated mean bits without accumulating the bits left. The second one is Simple, which only preserves remaining bits from previous one frame to current frame without any managing scheme. The third one is ISO, which is the method proposed in MP3 standard [1]. The fourth one is LAME-3.88, which is the bit reservoir scheme used in [14]. We also consider coding environment with or without other modules, e.g. window switch and M/S coding. The results shown in Table 3 and Table 4 illustrate that the new bit reservoir design could gain 0.2384 (Long/Short window, M/S coding) and 0.3375 (Long window only, without M/S coding) improvement in average than those without any reservoir controlling. With comparing to other reservoir schemes, our new design is also superior, especially in si02, sm02 and sm03.

Table 3: Objective measurements through the ODGs for different bit reservoir designs in MP3 CBR mode (Long/Short window, M/S coding).

Coding Methods	1	2	3	4	5
Stereo Modes	M/S	M/S	M/S	M/S	M/S
Allow Short	Yes	Yes	Yes	Yes	Yes
es01	-0.41	-0.34	-0.34	-0.33	-0.31
es02	-0.23	-0.19	-0.19	-0.22	-0.2
es03	-0.29	-0.28	-0.28	-0.24	-0.26
sc01	-0.62	-0.55	-0.55	-0.59	-0.56
sc02	-1.08	-0.99	-0.99	-1.07	-0.99
sc03	-0.93	-0.79	-0.79	-0.83	-0.79
si01	-1.05	-0.9	-0.9	-0.92	-0.86
si02	-2.03	-1.75	-1.75	-1.33	-1.09
si03	-1.72	-1.48	-1.48	-1.63	-1.48
sm01	-2.06	-1.73	-1.73	-1.91	-1.74
sm02	-0.75	-0.68	-0.68	-0.46	-0.49
sm03	-1.6	-1.37	-1.37	-1.27	-1.14
Average	-1.0642	-0.9208	-0.9208	-0.9	-0.8258

Bit Rate : 128kbps
Sample Rate : 44100 Hz
Coding Method :
1 : NoReservoir; 2 : Simple; 3 : ISO; 4 : LAME-3.88; 5 : Our New Design

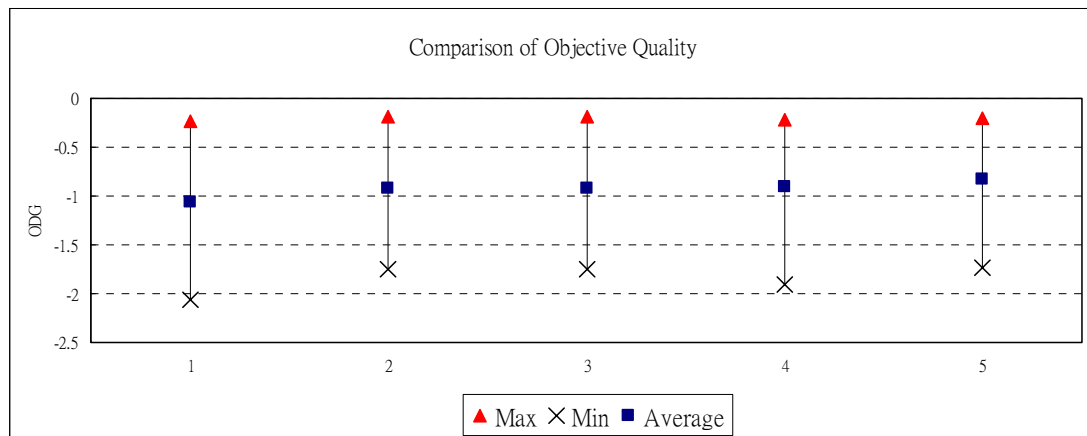


Figure 35: The ODG range comparison of Table 3. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.

Table 4: Objective measurements through the ODGs for different bit reservoir designs in MP3 CBR mode (Long window, without M/S coding).

Coding Methods	1	2	3	4	5
Stereo Modes	L/R	L/R	L/R	L/R	L/R
Allow Short	No	No	No	No	No
es01	-1.72	-1.46	-1.46	-1.35	-1.3
es02	-1.47	-1.33	-1.33	-1.29	-1.34
es03	-1.6	-1.35	-1.35	-1.28	-1.34
sc01	-0.84	-0.76	-0.76	-0.8	-0.76
sc02	-1.21	-1.06	-1.06	-1.17	-1.06
sc03	-1.3	-1.11	-1.11	-1.18	-1.1
si01	-1.19	-0.99	-0.99	-1.02	-0.99
si02	-2.97	-2.57	-2.31	-2.1	-1.7
si03	-1.79	-1.54	-1.54	-1.68	-1.54
sm01	-2.06	-1.75	-1.75	-1.91	-1.76
sm02	-1.09	-0.9	-0.9	-0.72	-0.71
sm03	-1.56	-1.33	-1.33	-1.26	-1.15
Average	-1.5667	-1.3458	-1.3242	-1.3133	-1.2292

Bit Rate : 128kbps
Sample Rate : 44100 Hz
Coding Method :
1 : NoReservoir; 2 : Simple; 3 : ISO; 4 : LAME-3.88; 5 : Our New Design

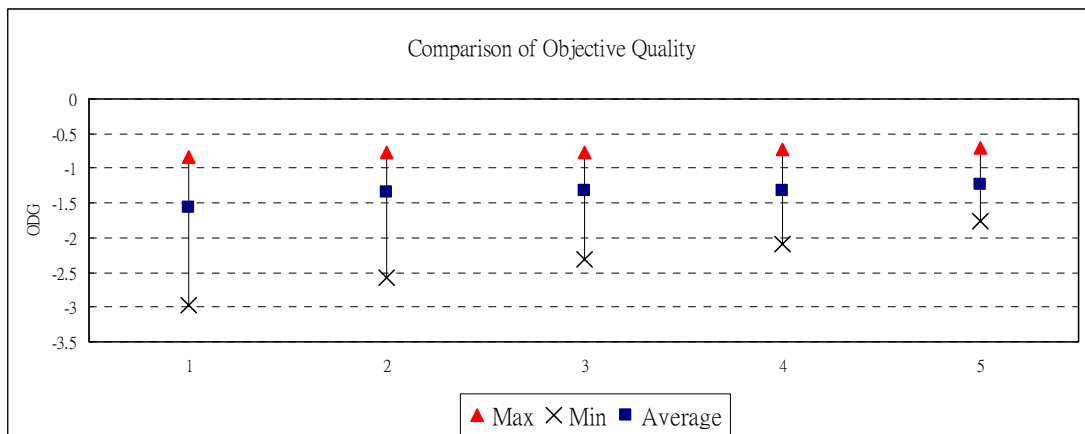


Figure 36: The ODG range comparison of Table 4. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.

4.5.1.2 Objective quality evaluation for MP3 ABR mode

For ABR mode, there are other two bit reservoir methods for comparison. The first one is NoReservoir, which just uses the allocated mean bits without accumulating the bits left. The second one is LAME-3.88, which is the bit reservoir scheme for ABR mode in [14]. We also consider coding environment with or without other modules, e.g. window switch and M/S coding. The results shown in Table 5 illustrate that the new bit reservoir design could gain 0.3309 (Long/Short window, M/S coding) and 0.4659 (Long window only, without M/S coding) improvement in average than those without any reservoir controlling. With comparing to reservoir schemes in LAME-3.88, our new design is also superior, especially in si01, si02 and sm01 with other assistant modules (window switch and M/S coding) or si01, si02, and natural vocal serious (es01, es02, and es03) without other assistant modules.

Table 5: Objective measurements through the ODGs for different bit reservoir designs in MP3 ABR mode.

Coding Methods	1	2	3	4	5	6
Stereo Modes	M/S	M/S	M/S	L/R	L/R	L/R
Allow Short	Yes	Yes	Yes	No	No	No
es01	-0.41	-0.36	-0.29	-1.72	-1.62	-1.09
es02	-0.23	-0.24	-0.16	-1.47	-1.43	-1.06
es03	-0.29	-0.27	-0.26	-1.6	-1.6	-1.09
sc01	-0.62	-0.63	-0.47	-0.84	-0.88	-0.68
sc02	-1.08	-1.12	-0.95	-1.21	-1.27	-1.07
sc03	-0.93	-0.97	-0.68	-1.3	-1.36	-1.04
si01	-1.05	-1.14	-0.71	-1.19	-1.27	-0.86
si02	-2.03	-1.47	-0.99	-2.97	-2.36	-1.48
si03	-1.72	-1.7	-1.42	-1.79	-1.75	-1.52
sm01	-2.06	-2.09	-1.45	-2.06	-2.07	-1.76
sm02	-0.75	-0.59	-0.32	-1.09	-0.82	-0.49
sm03	-1.6	-1.41	-1.1	-1.56	-1.39	-1.07
Average	-1.0642	-0.9992	-0.7333	-1.5667	-1.485	-1.1008
Bit Rate : 128kbps						
Sample Rate : 44100 Hz						
Coding Method :						
1, 4 : NoReservoir; 2, 5 : LAME-3.88; 3, 6 : Our New Design;						

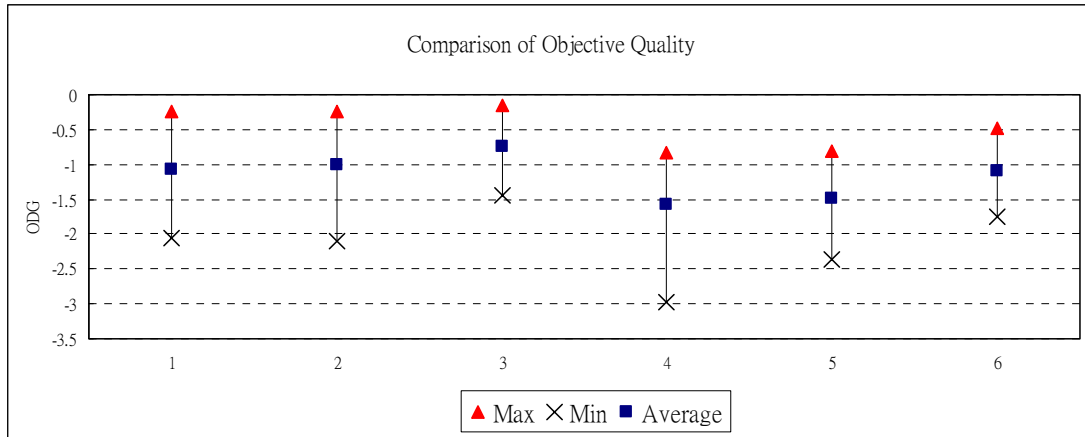


Figure 37: The ODG range comparison of Table 5. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.

4.5.1.3 Objective quality evaluation for AAC

For AAC, there are another 2 bit reservoir methods for comparison. The first one is NoReservoir, which just uses the allocated mean bits without accumulating the bits left. The second one is Simple, which only preserves remaining bits from previous one frame to current frame without any managing scheme. We also consider coding environment with or without other modules, e.g. Temporal Noise Shaping, window switch and M/S coding. The results shown in Table 6 illustrate that the new bit reservoir design could gain 0.1875 (TNS, Long/Short window, and M/S coding) and 0.4975 (Long window only, without M/S coding and TNS) improvement in average than those without any reservoir controlling. With comparing to other reservoir schemes, our new design is superior in si01, sm01 and sm02 with other assistant modules (TNS, window switch and M/S coding) or si02, sm02, sm03 and natural vocal serious (es01, es02, and es03) without other assistant modules.

Table 6: Objective measurements through the ODGs for different bit reservoir designs in AAC.

Coding Methods	1	2	3	4	5	6
Stereo Modes	M/S	M/S	M/S	L/R	L/R	L/R
Allow Short & TNS	Yes	Yes	Yes	No	No	No
es01	-0.41	-0.37	-0.32	-1.56	-1.42	-0.93
es02	-0.19	-0.14	-0.13	-1.98	-1.75	-1.6

es03	-0.27	-0.27	-0.22	-2.24	-2	-1.43
sc01	-0.54	-0.5	-0.45	-0.7	-0.65	-0.59
sc02	-0.82	-0.67	-0.63	-0.98	-0.79	-0.74
sc03	-0.44	-0.4	-0.37	-0.6	-0.52	-0.45
si01	-0.79	-0.68	-0.55	-1.08	-0.91	-0.7
si02	-0.63	-0.56	-0.52	-3.28	-3	-1.97
si03	-1.08	-0.94	-0.92	-1.21	-1.06	-1.04
sm01	-0.84	-0.69	-0.51	-0.81	-0.64	-0.46
sm02	-1.2	-1.08	-0.5	-1.54	-1.4	-0.67
sm03	-0.7	-0.61	-0.54	-1.2	-1.03	-0.63
Average	-0.6592	-0.5758	-0.4717	-1.4317	-1.2642	-0.9342

Bit Rate : 128kbps
Sample Rate : 44100 Hz
Coding Method :
1, 4 : NoReservoir; 2, 5 : Simple; 3, 6 : Our New Design;

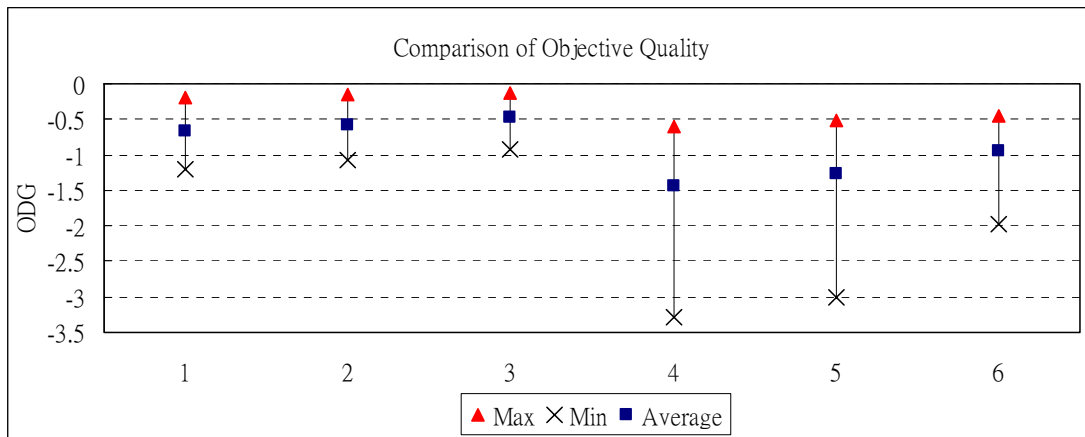


Figure 38: The ODG range comparison of Table 6. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.

4.5.2 Parameter Evaluation

In this section, we evaluate the parameters mentioning in this thesis. The best parameter combinations are adopted as the default setting for previous and proceeding sections.

4.5.2.1 Parameter evaluation for MP3 CBR mode

There are four parameters needed to evaluate. *UBound* is used for filtering out extreme value of AE as described in subsection 4.2.1. *Avg_length*, which is used for controlling reference data length, is the same as the variable *N* described in subsection 4.2.1. *Zero_R* and *Zero_L* are boundaries of zero-zone described in subsection 4.2.2.1. The evaluation results are shown in Table 7 and the best parameter choice is indicated by bold font type. In addition to these four parameters, the detailed complicated tuning process for demand curve and budget curve for MP3 CBR mode is omitted here for simplifying. We only show the tuning results in Figure 28 and Figure 31 for reference. From the results in Table 7, we choose *UBound* = 1200, *Avg_length* = 15, *Zero_R* = 0.45, and *Zero_L* = -0.2 as our default parameters setting in subsection 4.5.1.1 and 4.5.3.1.

Table 7: MP3 CBR mode parameters evaluation.

New Bit Reservoir Design for MP3 CBR Mode						
NCTU-MP3 (Long/Short window, M/S coding)	Step 1 : UBound					
	Ubound	1000	1100	1200	1300	1400
	Min	-1.75	-1.74	-1.74	-1.74	-1.74
	Max	-0.21	-0.2	-0.2	-0.21	-0.21
	Average	-0.8292	-0.8275	-0.8267	-0.8275	-0.8283
	Step 2 : fix the UBound as 1200					
	Avg_length	15	20	25	30	35
	Min	-1.74	-1.74	-1.75	-1.75	-1.74
	Max	-0.2	-0.21	-0.2	-0.2	-0.21
	Average	-0.8267	-0.8267	-0.8275	-0.8283	-0.8275
	Step 3 : fix the Avg_length as 15					
	Zero_R	0.35	0.4	0.45	0.5	0.55
	Min	-1.74	-1.74	-1.74	-1.74	-1.74
	Max	-0.2	-0.2	-0.2	-0.21	-0.2
	Average	-0.8292	-0.8258	-0.8258	-0.8267	-0.8267
	Step 4 : fix the Zero_R as 0.45					
	Zero_L	0	-0.1	-0.2	-0.25	-0.3
	Min	-1.73	-1.74	-1.74	-1.74	-1.74
	Max	-0.21	-0.21	-0.2	-0.2	-0.2
	Average	-0.8292	-0.8267	-0.8258	-0.8258	-0.8258

	Best Parameter : UBound = 1200; Avg_length = 15; Zero_R = 0.45; Zero_L = -0.2
Bit Rate : 128kbps	
Sample Rate : 44100 Hz	

4.5.2.2 Parameter evaluation for MP3 ABR mode

There are six parameters needed to evaluate. *UBound* is used for filtering out extreme value of AE as described in subsection 4.2.1. *Avg_length*, which is used for controlling reference data length, is the same as the variable *N* described in subsection 4.2.1. *Demand_zero_R* and *Demand_zero_L* are boundaries of zero-zone for demand curve described in subsection 4.2.2.2. *Budget_zero_R* and *Budget_zero_L* are boundaries of zero-zone for budget curve described in subsection 4.3.2. The evaluation results are shown in Table 8 and the best parameter choice is indicated by bold font type. In addition to these six parameters, the detailed complicated tuning process for demand curve and budget curve for MP3 ABR mode is omitted here for simplifying. We only show the tuning results in Figure 29 and Figure 32 for reference.

Table 8: MP3 ABR mode parameters evaluation.

New Bit Reservoir Design for MP3 ABR Mode						
NCTU-MP3 (Long/Short window, M/S coding)	Step 1 : UBound					
	Ubound	800	900	950	1000	1050
	Min	-1.54	-1.46	-1.45	-1.51	-1.53
	Max	-0.17	-0.16	-0.17	-0.17	-0.17
	Average	-0.7558	-0.7383	-0.735	-0.7375	-0.74
	Step 2 : fix the UBound as 950					
	Avg_length	25	30	35	40	45
	Min	-1.48	-1.45	-1.45	-1.46	-1.45
	Max	-0.17	-0.17	-0.16	-0.16	-0.17
	Average	-0.7392	-0.735	-0.7333	-0.7333	-0.7358
	Step 3 : fix the Avg_length as 35					
	Demand_zero_R	0	0.01	0.02	0.03	0.04
	Min	-1.46	-1.45	-1.46	-1.47	-1.47
	Max	-0.16	-0.16	-0.15	-0.16	-0.16
	Average	-0.74	-0.7333	-0.7375	-0.7408	-0.7425
	Step 4 : fix the Demand_zero_R as 0.01					

Demand_zero_L	0	-0.1	-0.15	-0.2	-0.25
Min	-1.48	-1.46	-1.46	-1.45	-1.45
Max	-0.16	-0.16	-0.16	-0.16	-0.16
Average	-0.7408	-0.735	-0.7342	-0.7333	-0.7333
Step 5 : fix the Demand_zero_L as -0.2					
Budget_zero_R	0.04	0.05	0.06	0.07	0.08
Min	-1.46	-1.46	-1.47	-1.45	-1.45
Max	-0.16	-0.16	-0.16	-0.16	-0.16
Average	-0.7342	-0.7342	-0.735	-0.7333	-0.7342
Step 6 : fix the Budget_zero_R as 0.07					
Budget_zero_L	-0.05	-0.1	-0.15	-0.2	-0.25
Min	-1.46	-1.46	-1.46	-1.45	-1.46
Max	-0.16	-0.16	-0.16	-0.16	-0.16
Average	-0.7342	-0.7342	-0.735	-0.7333	-0.735
Best Parameter : UBound = 950; Avg_length = 35; Demand_zero_R = 0.01; Demand_zero_L = -0.2; Budget_zero_R = 0.07; Budget_zero_L = -0.2					
Bit Rate : 128kbps					
Sample Rate : 44100 Hz					

From the results in Table 8, we choose $UBound = 950$, $Avg_length = 35$, $Demand_zero_R = 0.01$, $Demand_zero_L = -0.2$, $Budget_zero_R = 0.07$, and $Budget_zero_L = -0.2$ as our default parameters setting in subsection 4.5.1.2 and 4.5.3.2.

4.5.2.3 Parameter evaluation for AAC

There are six parameters needed to evaluate. $UBound$ is used for filtering out extreme value of AE as described in subsection 4.2.1. Avg_length , which is used for controlling reference data length, is the same as the variable N described in subsection 4.2.1. $Demand_zero_R$ and $Demand_zero_L$ are boundaries of zero-zone for demand curve described in subsection 4.2.2.3. $Budget_zero_R$ and $Budget_zero_L$ are boundaries of zero-zone for budget curve described in subsection 4.3.3. The evaluation results are shown in Table 9 and the best parameter choice is indicated by bold font type. In addition to these six parameters, the detailed complicated tuning process for demand curve and budget curve for AAC is omitted here for simplifying. We only show the tuning results in Figure 30 and Figure 33 for reference.

Table 9: AAC parameters evaluation.

New Bit Reservoir Design for AAC						
NCTU-AAC (Long/Short window, M/S coding, TNS)	Step 1 : UBound					
	Ubound	3900	4000	4100	4200	4300
	Min	-0.92	-0.92	-0.92	-0.92	-0.92
	Max	-0.13	-0.14	-0.14	-0.14	-0.14
	Average	-0.4775	-0.4758	-0.4758	-0.4767	-0.4775
	Step 2 : fix the UBound as 4100					
	Avg_length	40	42	45	47	50
	Min	-0.92	-0.93	-0.93	-0.92	-0.92
	Max	-0.13	-0.13	-0.13	-0.13	-0.13
	Average	-0.4767	-0.475	-0.4742	-0.4733	-0.475
	Step 3 : fix the Avg_length as 47					
	Demand_zero_R	0	0.01	0.02	0.03	0.04
	Min	-0.92	-0.92	-0.92	-0.92	-0.92
	Max	-0.13	-0.13	-0.13	-0.12	-0.12
	Average	-0.4733	-0.4742	-0.4758	-0.475	-0.4758
	Step 4 : fix the Demand_zero_R as 0					
	Demand_zero_L	0	-0.01	-0.02	-0.03	-0.04
	Min	-0.93	-0.92	-0.92	-0.92	-0.93
	Max	-0.12	-0.13	-0.13	-0.13	-0.13
	Average	-0.4792	-0.475	-0.4733	-0.4742	-0.475
	Step 5 : fix the Demand_zero_L as -0.02					
	Budget_zero_R	0	0.05	0.1	0.15	0.2
	Min	-0.92	-0.92	-0.92	-0.92	-0.92
	Max	-0.13	-0.13	-0.13	-0.13	-0.12
	Average	-0.4725	-0.4733	-0.4717	-0.4725	-0.4725
	Step 6 : fix the Budget_zero_R as 0.1					
	Budget_zero_L	-0.3	-0.35	-0.4	-0.45	-0.5
	Min	-0.93	-0.92	-0.92	-0.93	-0.92
	Max	-0.12	-0.12	-0.13	-0.12	-0.13
	Average	-0.4725	-0.4725	-0.4717	-0.4733	-0.4742
Best Parameter : UBound = 4100; Avg_length = 47; Demand_zero_R = 0; Demand_zero_L = -0.02; Budget_zero_R = 0.1; Budget_zero_L = -0.4						

Bit Rate : 128kbps
Sample Rate : 44100 Hz

From the results in Table 9, we choose $UBound = 4100$, $Avg_length = 47$, $Demand_zero_R = 0$, $Demand_zero_L = -0.02$, $Budget_zero_R = 0.1$, and $Budget_zero_L = -0.4$ as our default parameters setting in subsection 4.5.1.3 and 4.5.3.3.

4.5.3 Objective quality measurement based on music database

In order to verify the robustness of the proposed new bit reservoir design in different codecs and evaluate the possible risk for a variety of music categories, we adopt PSPLAB audio database [26] as our testing material. The database includes 327 tracks that are separated into 16 sets with different signal properties as shown in Table 10.

Table 10: The PSPLAB audio database.

Bitstream 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 critical 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 bitstream 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	First 8 Vobis testing sample from HA.
Total		327	

4.5.3.1 Objective quality measurement based on music database for MP3 CBR mode

The quality measurements for MP3 CBR mode are listed in this section. Because NCTU-MP3 only supports 8 and 16 bits PCM track format, the MP3 objective quality measurement just applies to 319 tracks in practice. The average ODG result of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database is illustrated in Figure 39. On average, our new bit reservoir design could gain 0.1956 ODG improvement than without applying bit reservoir scheme. Figure 40 and Figure 41 illustrate the enhancement and degradation tracks distribution for 319 tracks in different ODG difference range. The enhancement ratio is up to 94.4 %, and only 14 tracks (by removing duplicated tracks) get degradation. Our new bit reservoir for MP3 CBR mode is truly beneficial for great part of audio tracks.

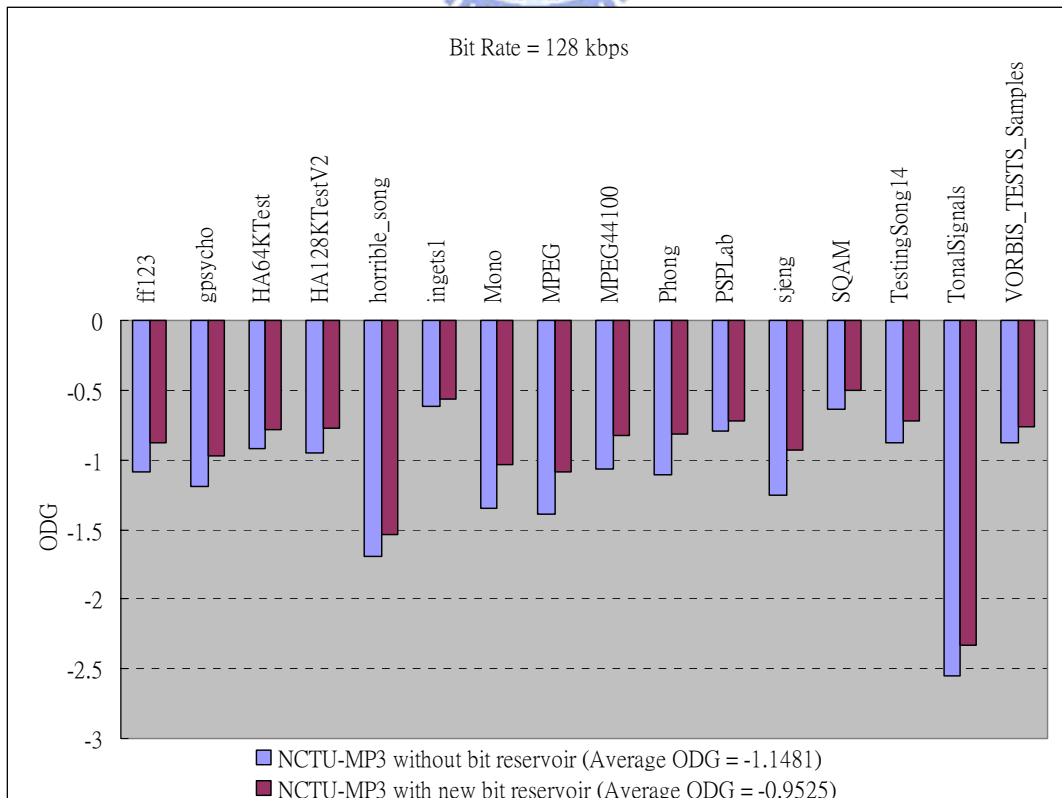


Figure 39: The average objective quality of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

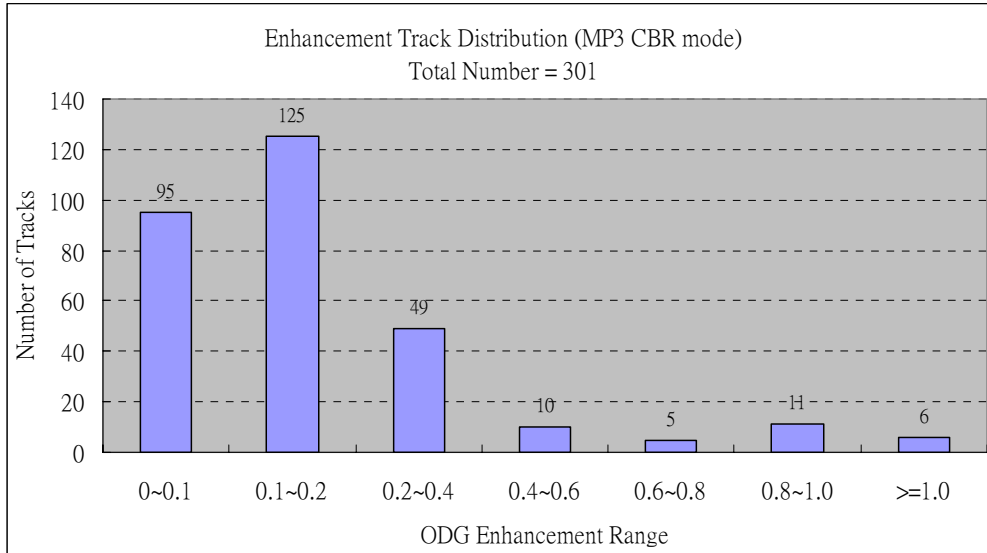


Figure 40: The enhancement tracks distribution of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

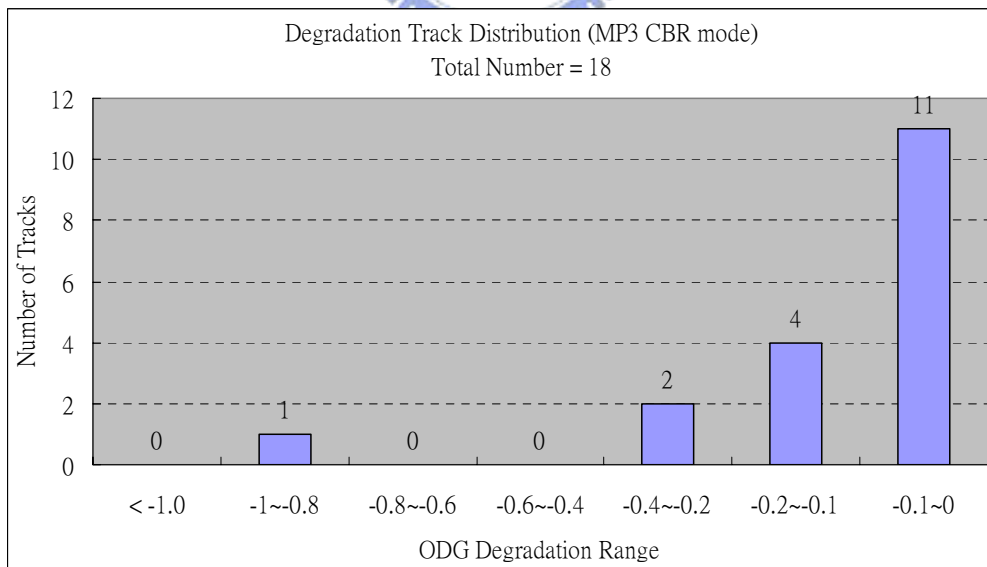


Figure 41: The degradation tracks distribution of NCTU-MP3 CBR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

4.5.3.2 Objective quality measurement based on music database for MP3 ABR mode

The average ODG result of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database is illustrated in Figure 42. On average, our new bit reservoir design could gain 0.253 ODG improvement than without applying bit reservoir scheme. Figure 43 and Figure 44 illustrate the enhancement and degradation tracks distribution for 319 tracks in different ODG difference range. The enhancement ratio is up to 93.1 %, and only 16 tracks (by removing duplicated tracks) get degradation. Our new bit reservoir for MP3 ABR mode is truly beneficial for great part of audio tracks.

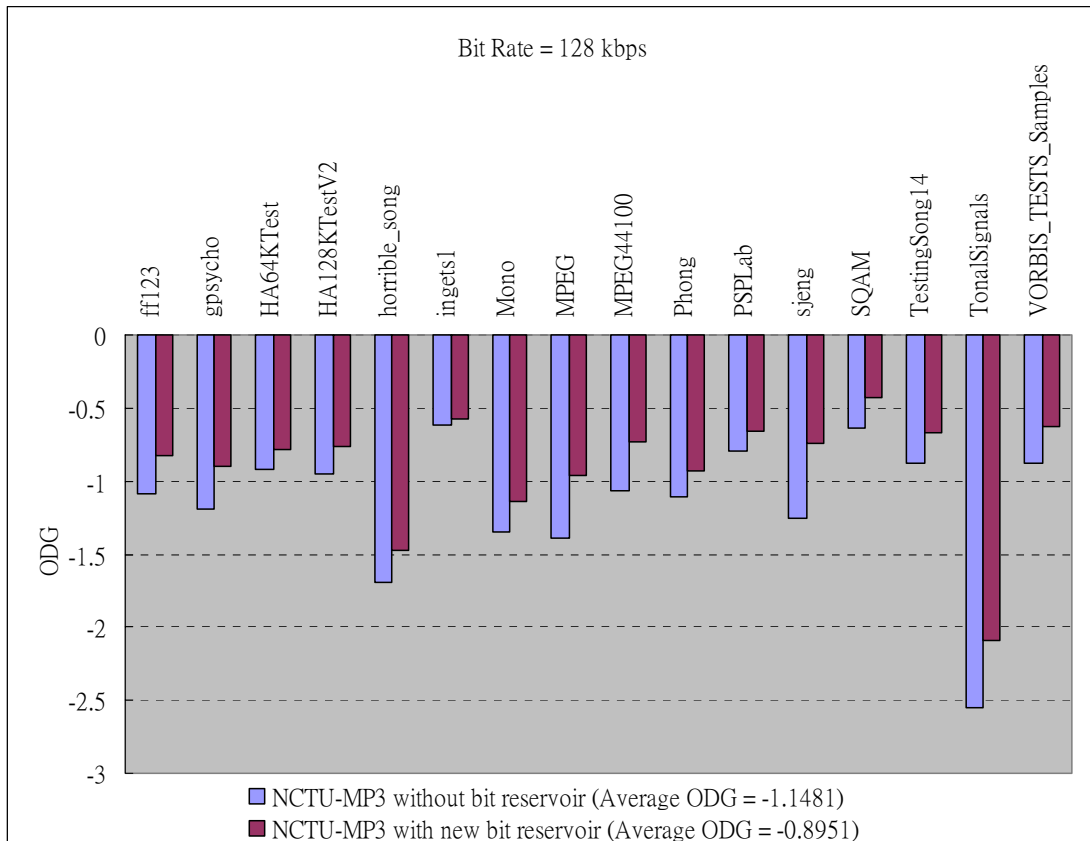


Figure 42: The average objective quality of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

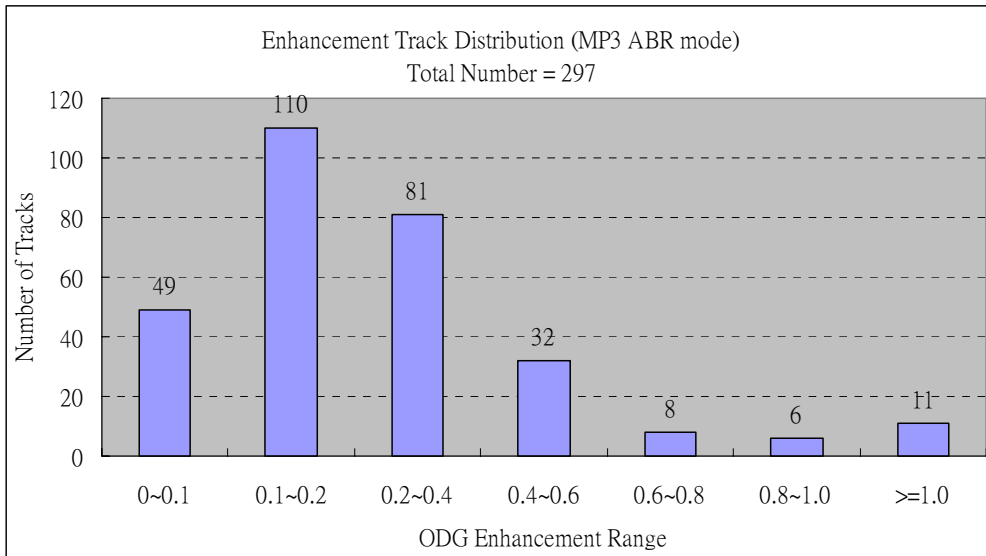


Figure 43: The enhancement tracks distribution of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

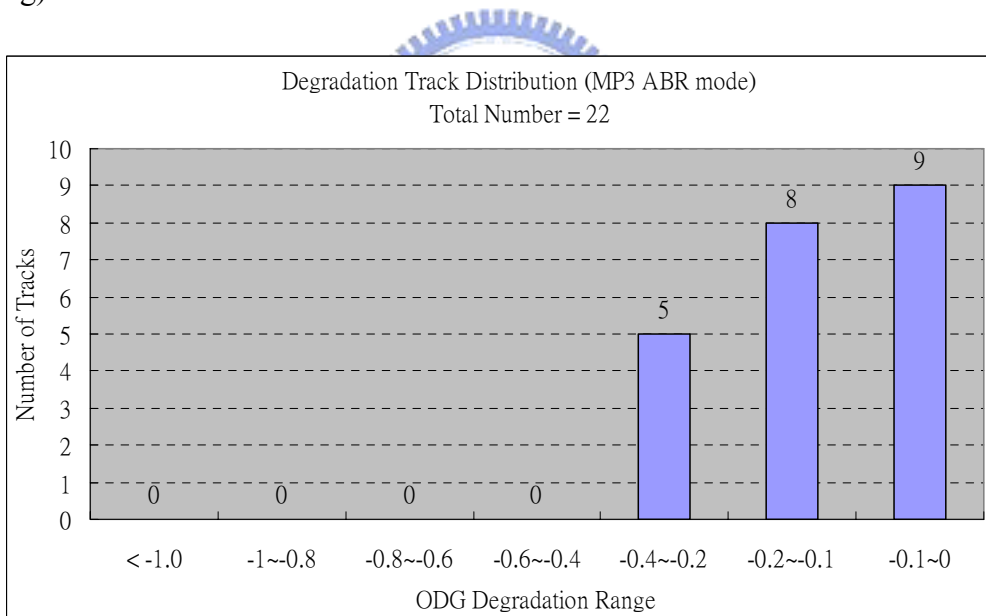


Figure 44: The degradation tracks distribution of NCTU-MP3 ABR mode without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding).

4.5.3.3 Objective quality measurement based on music database for AAC

The average ODG result of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database is illustrated in Figure 45. On average, our new bit reservoir design could gain 0.184 ODG improvement than without applying bit reservoir scheme. Figure 46 and Figure 47 illustrate the enhancement and degradation tracks distribution for 327 tracks in different ODG difference range. The enhancement ratio is up to 99.1 %, and only 3 tracks get degradation. Our new bit reservoir for AAC is truly beneficial for great part of audio tracks.

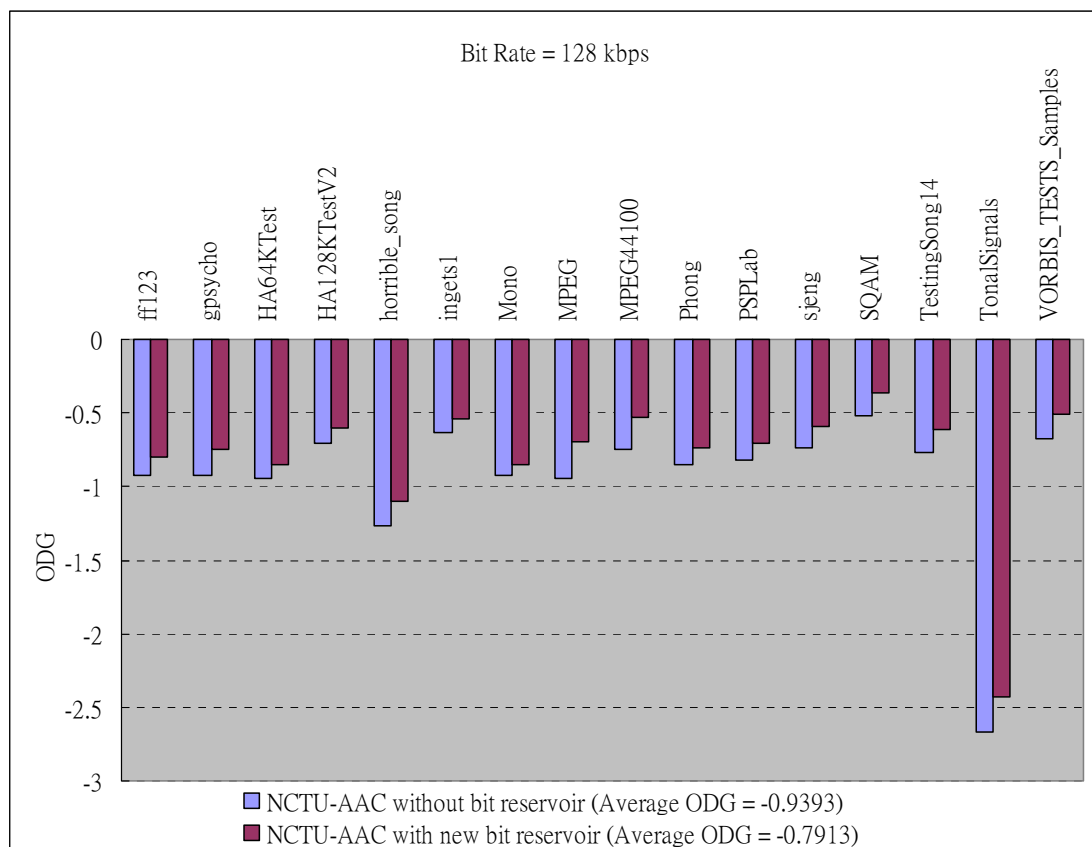


Figure 45: The average objective quality of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

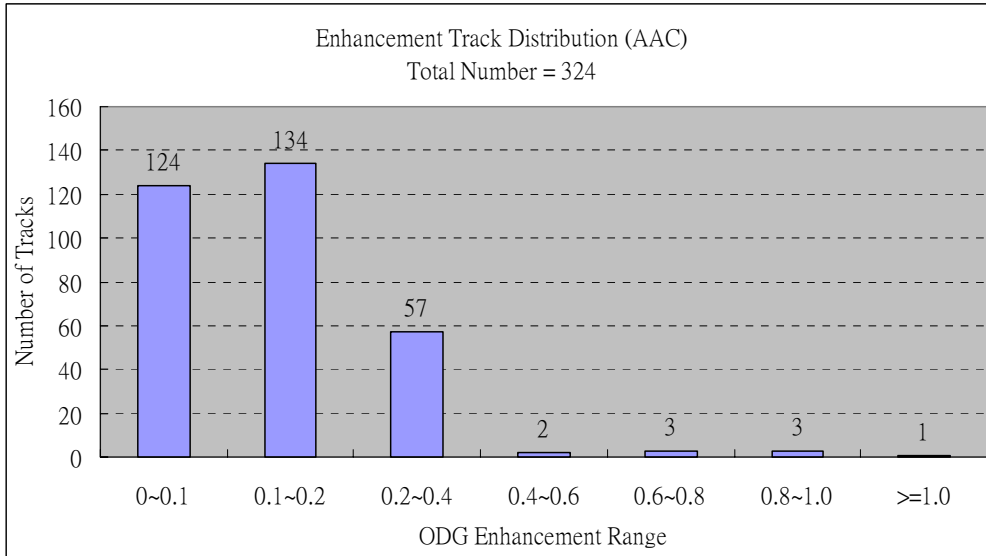


Figure 46: The enhancement tracks distribution of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

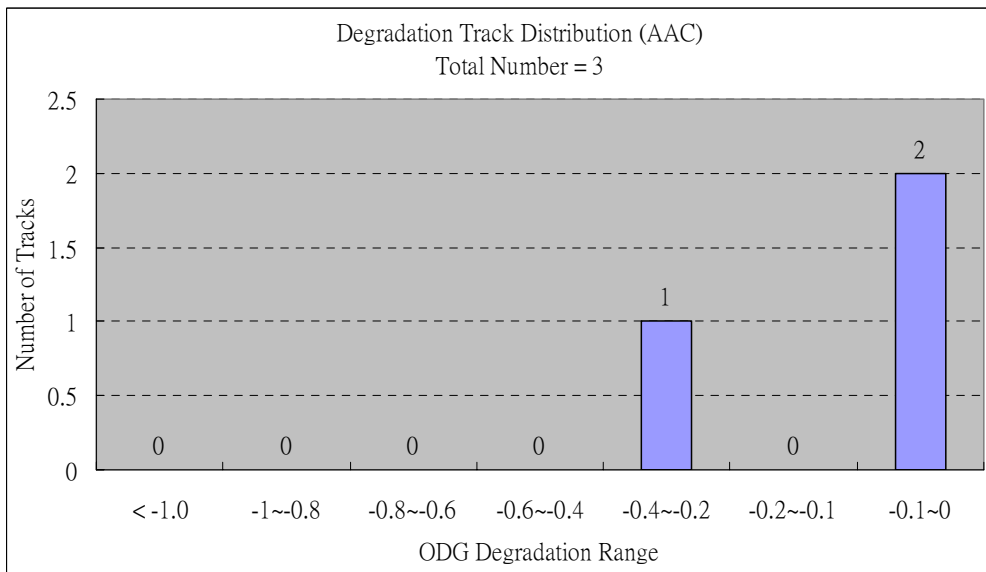


Figure 47: The degradation tracks distribution of NCTU-AAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 128 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

4.5.4 Objective quality measurement with existing codecs

In this section, we make a summary of our encoders with new bit reservoir design by comparing with other existing audio codecs. For MP3 encoder, we compare our NCTU-MP3 [20] with LAME-3.88 [14] in CBR mode and ABR mode at different

bit rates respectively. The experiment results are shown in Table 11 and Table 12. For AAC encoder, we compare our NCTU-AAC [21] with QuickTime 6.5.2 [28] and Nero 6.6.0.8 [29] at different low bit rates. The experiment results for AAC are illustrated in Table 13. Through observing these comparison results, our encoder with new bit reservoir design is superior to other encoders in average no matter what codec we evaluate.

Table 11: Objective quality comparison for MP3 CBR mode at different bit rates.

Bit Rates	96 kbps		112 kbps		128 kbps	
	NCTU-MP3	LAME-3.88	NCTU-MP3	LAME-3.88	NCTU-MP3	LAME-3.88
es01	-0.86	-1.59	-0.53	-0.87	-0.31	-0.56
es02	-0.57	-1.22	-0.29	-0.71	-0.2	-0.48
es03	-0.64	-1.45	-0.32	-0.81	-0.26	-0.56
sc01	-1.25	-1.38	-0.83	-0.98	-0.56	-0.69
sc02	-2.09	-2.02	-1.36	-1.47	-0.99	-1.09
sc03	-1.94	-2.19	-1.24	-1.38	-0.79	-0.96
si01	-2.58	-2.35	-1.71	-1.48	-0.86	-1.01
si02	-2.31	-2.11	-1.66	-1.46	-1.09	-1.02
si03	-3.5	-3.67	-2.73	-2.8	-1.48	-1.58
sm01	-3.5	-3.56	-2.81	-2.82	-1.74	-1.82
sm02	-1.33	-1.51	-0.73	-0.92	-0.49	-0.59
sm03	-2.37	-2.53	-1.82	-1.91	-1.14	-1.32
Average	-1.9117	-2.1317	-1.3358	-1.4675	-0.8258	-0.9733

Sample Rate : 44100 Hz
Encoder:
NCTU-MP3: with Long/Short window, M/S coding, and our new bit reservoir design.
LAME-3.88 : -b 128 -s 44100 -m j --cbr -k

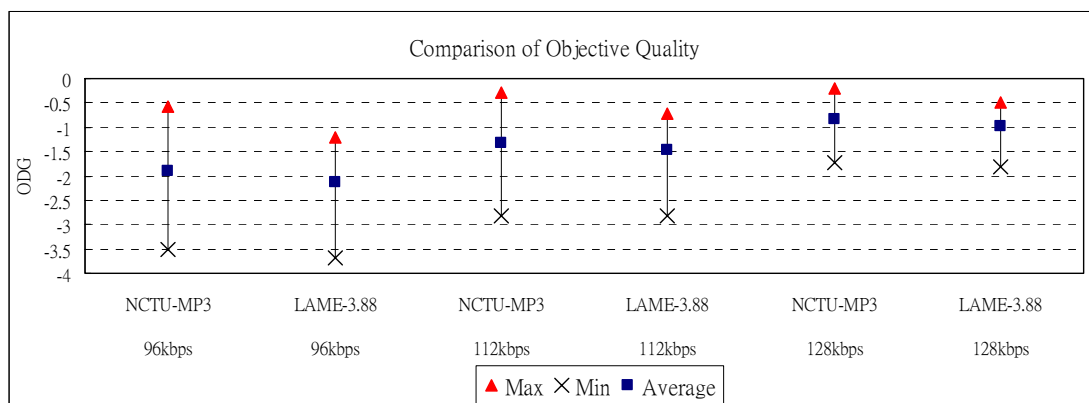


Figure 48: The ODG range comparison of Table 11.

Table 12: Objective quality comparison for MP3 ABR mode at different bit rates.

Bit Rates	96 kbps		112 kbps		128 kbps	
	NCTU-MP3	LAME-3.88	NCTU-MP3	LAME-3.88	NCTU-MP3	LAME-3.88
es01	-0.8	-1.73	-0.49	-1.05	-0.29	-0.71
es02	-0.4	-1.08	-0.22	-0.66	-0.16	-0.46
es03	-0.39	-1.32	-0.25	-0.8	-0.26	-0.61
sc01	-1.15	-1.63	-0.74	-1.22	-0.47	-0.88
sc02	-1.98	-2.52	-1.27	-1.91	-0.95	-1.48
sc03	-1.79	-2.44	-1.13	-1.63	-0.68	-1.17
si01	-2.31	-2.8	-1.46	-1.87	-0.71	-1.23
si02	-2.02	-1.89	-1.44	-1.28	-0.99	-0.88
si03	-3.45	-3.8	-2.67	-3.48	-1.42	-2.65
sm01	-3.34	-3.75	-2.55	-3.47	-1.45	-2.81
sm02	-0.93	-1.76	-0.44	-1.07	-0.32	-0.71
sm03	-2.21	-2.76	-1.63	-2.32	-1.1	-1.73
Average	-1.7308	-2.29	-1.1908	-1.73	-0.7333	-1.2767

Sample Rate : 44100 Hz
Encoder:
NCTU-MP3: with Long/Short window, M/S coding, and our new bit reservoir design.
LAME-3.88: --abr 128 -b 0 -B 320 -s 44100 -m j -k

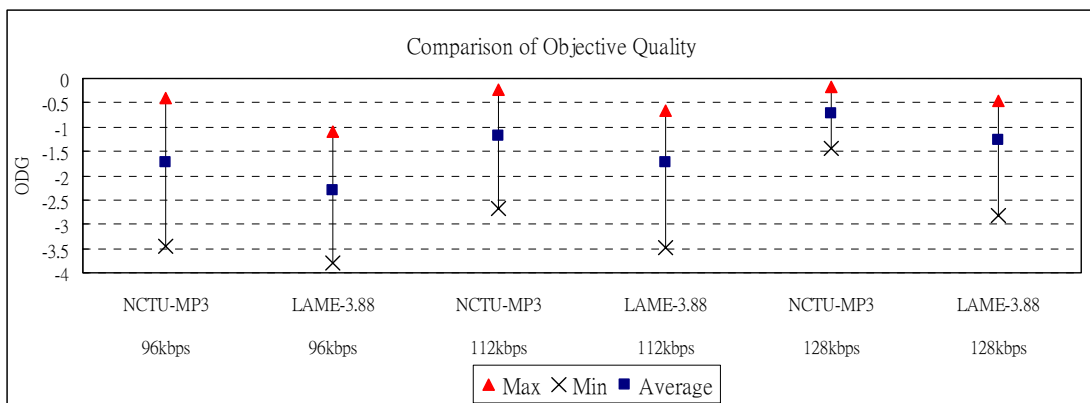


Figure 49: The ODG range comparison of Table 12.

Table 13: Objective quality comparison for AAC at different bit rates.

Bit Rates	96 kbps			112 kbps			128 kbps		
Encoders	NCTU-AAC	Quick Time 6.5.2	Nero 6.6.0.8	NCTU-AAC	Quick Time 6.5.2	Nero 6.6.0.8	NCTU-AAC	Quick Time 6.5.2	Nero 6.6.0.8
es01	-0.63	-0.54	-1.28	-0.41	-0.36	-0.93	-0.32	-0.24	-0.58
es02	-0.28	-0.32	-1.09	-0.2	-0.47	-0.63	-0.13	-0.11	-0.47
es03	-0.36	-0.19	-1.18	-0.29	-0.05	-0.72	-0.22	0.04	-0.48
sc01	-1.02	-0.96	-1.45	-0.63	-0.47	-1.14	-0.45	-0.2	-0.88
sc02	-1.74	-1.9	-2.47	-1.1	-1.2	-1.98	-0.63	-0.74	-1.38
sc03	-0.95	-1.57	-1.85	-0.49	-1.06	-1.34	-0.37	-0.61	-0.82
si01	-1.6	-1.81	-2	-0.88	-1.09	-1.87	-0.55	-0.65	-1.35
si02	-1.21	-1.36	-1.85	-0.76	-0.96	-1.14	-0.52	-0.64	-0.86
si03	-2.58	-2.05	-1.95	-1.48	-1.2	-2	-0.92	-0.7	-1.56
sm01	-2.19	-2.74	-2.44	-1.07	-1.65	-2.14	-0.51	-0.89	-1.25
sm02	-1.03	-1.48	-1.34	-0.72	-0.76	-1.18	-0.5	-0.37	-0.79
sm03	-1.53	-1.73	-1.98	-0.87	-1.18	-1.9	-0.54	-0.67	-1.29
Average	-1.26	-1.3875	-1.74	-0.7417	-0.8708	-1.4142	-0.4717	-0.4817	-0.9758

Sample Rate : 44100 Hz

Encoder:

NCTU-AAC: with Long/Short window, M/S coding, TNS, and our new bit reservoir design.

QuickTime 6.5.2 : Stereo, 44.1 kHz, Best Quality.

Nero 6.0.8 : LC AAC, CBR, Stereo, High Quality.

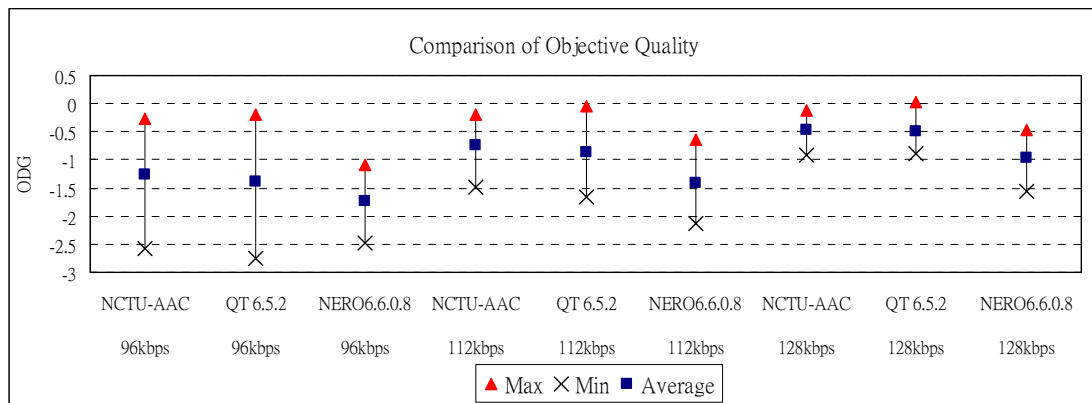


Figure 50: The ODG range comparison of Table 13.

Chapter 5

Bit Reservoir Design for HE-AAC

High Efficiency AAC (HE-AAC) has included the Spectral Band Replication (SBR) in combination with conventional AAC to achieve high audio quality at bit rates lower than 96 kbps. The bits allocated to AAC encoder and SBR module decides the quality and compression efficiency. In Chapter 4, we have designed the bit reservoir for AAC to reserve and predict the bits necessary for each time frame. The concept of bit reservoir should be extended for HE-AAC especially for the SBR module. To extend the bit reservoir to HE-AAC, we need to have the estimator and regulator for the SBR encoder. The estimator needs to predict the bit required for the SBR part while regulator needs to leave a budget for the SBR encoder. Also, the estimator and regulator in SBR should be suitably combined with the estimator and the regulator in AAC to have global control. In this chapter, we first introduce the fundamental concept of SBR and then propose a demand estimator for SBR and a global budget control between AAC and SBR.

5.1 Spectral Band Replication (SBR)

In order to obtain good perceptual quality under bit rate constraint, almost all audio codecs sacrifice the high frequency component of signals and put all available bits to the low frequency component that is more important for human hearing. However, the hearing perception becomes muffling as the audio bandwidth getting lower [6]. Under the tradeoff between bandwidth limit and quality, Spectral Band Replication (SBR) [3] has been proposed to compress high frequency contents with much less bits overheads.

The basic principle of SBR is to reconstruct the high frequency spectral bands by replicating the low frequency spectral bands and rescale the spectral envelope of the reconstructed part closely to the original signal according to the priori information extracted by the SBR encoder. The block diagram of SBR encoder is illustrated in Figure 51. The conventional AAC encoder only needs to encode the low frequency parts of the audio signals and consequently half of the original sample rate is enough to keep signal information based on Nyquist's theorem. Therefore, the HE-AAC codec is a dual rate system where the underlying AAC encoder/decoder is operated as

half the sample rate of the SBR encoder/decoder [5].

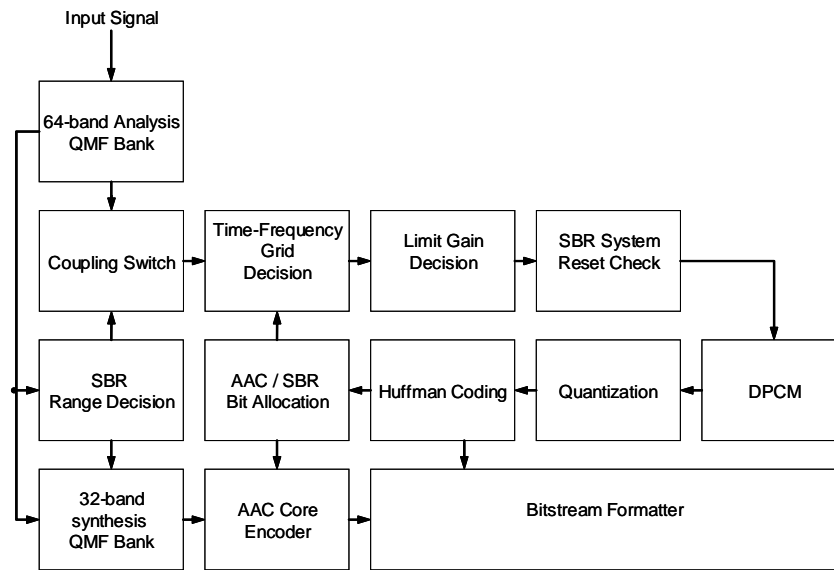


Figure 51: Block diagram of SBR encoder.

The responsibility of the SBR encoder is to extract the high bands information that includes the data of spectral envelope representation, tonal-to-noise ratio, and other control parameters. The control parameters are estimated to ensure that the high frequency reconstruction results in a reconstructed high band that is perceptually as similar as possible to the original high band. To extract this information, the original full bandwidth signal is separated into 64 subbands by a complex-valued quadratic mirror filter (QMF). The SBR range consisting of high bands is separated into several envelopes that are determined by some time points. The determination of these control parameters depends on the stable situation of the signal content. Moreover, several subbands are combined as non-uniform bands from frequency aspect. The frequency resolution table decides the segment points. Through the two dimensional segment, a time-frequency grid gathering the subband signals covered by the SBR range is constructed [6]. A grid example is shown in Figure 52. A spectrogram of the original signal is displayed with a superimposed time-frequency grid of the spectral envelope data transmitted to the decoder. It is obvious that the time-frequency resolution of the spectral envelope varies over time. Higher time resolution is used for transient passages while higher frequency resolution is used for the stationary passages.

On the other hand, the adding of noise or sinusoids with suitable energy ratio is considered to deal with the inconsistency of the tonal-to-noise ratio between the original spectral bands and the replicated ones. The extraction of the ratio information is based on the time-frequency grid units with different resolutions. Although the bit rate of SBR control data varies depending on coder setting, it is in general in the

region of somewhere about 1~3 kbits per channel per second [5]. This requirement is far lower than any conventional coding algorithm to encode the high bands.

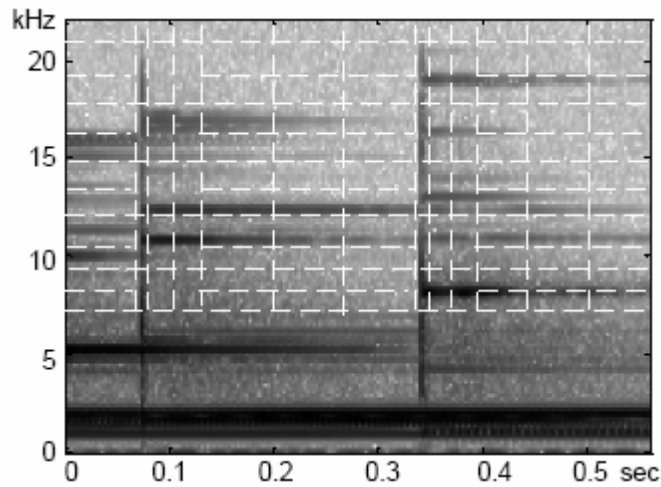


Figure 52: The spectrogram of an interval in input signal with the superimposed envelope time-frequency grid [5].

5.2 Demand Estimator for SBR

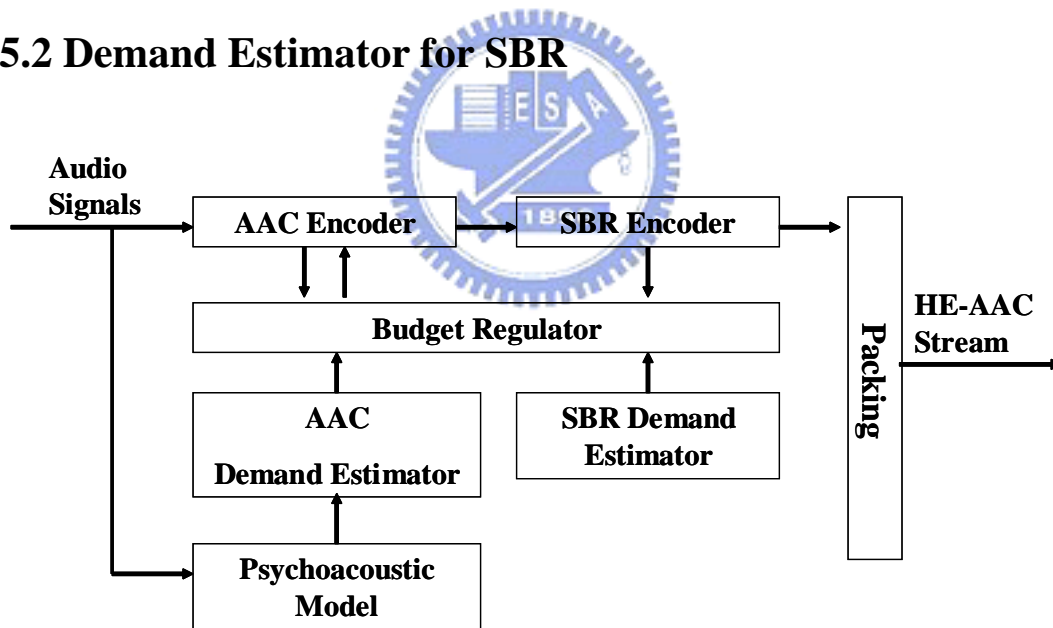


Figure 53: Block diagram of bit reservoir for HE-AAC.

We propose a bit reservoir design for HE-AAC [16] by extending the concept of estimator and regulator in AAC. The block diagram of our new design is depicted in Figure 53. The SBR demand estimator predicts the bits required for the SBR encoder. Also, the AAC demand estimator is the same as the estimator described in section 4.2. The budget regulator acts as a global distributor to assign bits to AAC encoder with leaving some budget for SBR encoder. On the coding sequence, we adopt proceeding first with the AAC encoder based on the allocated bits and then the SBR encoder. The

features of the algorithm can be considered from two aspects. First, we use one common budget regulator while two demand estimators for the AAC and SBR encoders. Second, we leave the budget for the SBR without directly regulating the encoded bits in SBR encoder. In this section, we propose the SBR demand estimator for a start and then combine this mechanism with global budget regulator for HE-AAC bit allocation in next section.

The bit rate of control parameters of SBR encoder varies with the SBR encoder modules inside, e.g. Grid and High Frequency Generation (HFG) [3]. Unlike conventional AAC encoder with psychoacoustic model, there is currently no entropy mechanism to represent or evaluate the bits required of SBR encoder. Hence the design of SBR bit estimator should start with observing bit consumption in SBR encoder over frames. Figure 54 to Figure 57 illustrates the consumed SBR bits with respect to audio frames for four different kinds of tracks. These four tracks selected from Table 2 are natural vocal (es03), complex sound (sc02), transient (si02), and harmonic (si03).

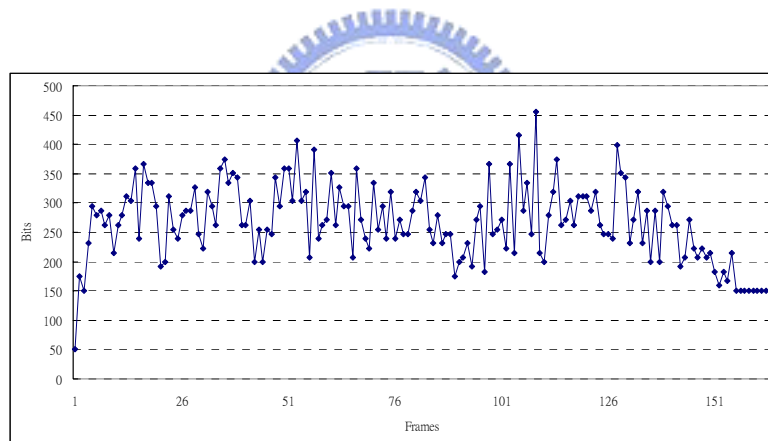


Figure 54: Natural vocal (es03).

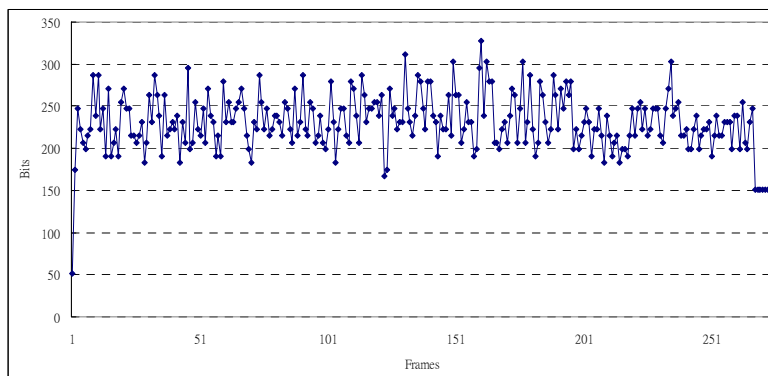


Figure 55: Complex sound (sc02).

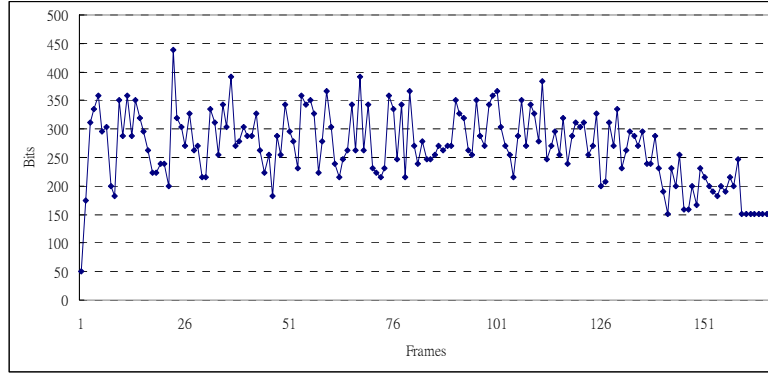


Figure 56: Transient (si02).

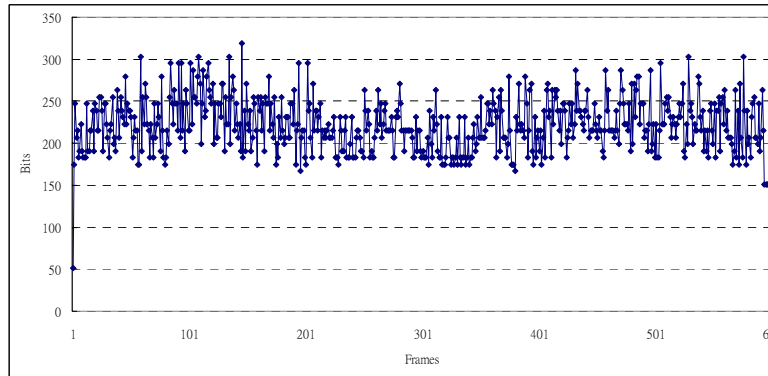


Figure 57: Harmonic (si03).

We also show the bits usage of other tracks within the twelve test tracks recommended by MPEG as listed in Table 2. The minimum, maximum, average, and standard deviation value of bits usage of SBR encoder in each track at bit rate 80kbps for HE-AAC is shown in Table 14. The percentage of these values in the mean bits at 80 kbps is listed, too. The standard deviation is derived by

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad \text{for } \forall x_i > 151, \quad (34)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{for } \forall x_i > 151, \quad (35)$$

where x_i is the number of bits in each SBR frame, n is the number of frames exclusive the silence frames where a threshold 151 bits have been use as the selection criterion.

Table 14: The minimum, maximum, average, and standard deviation of bits usage at 80 kbps. The “Minimum” and “Maximum”, “Average”, and “Standard Deviation” columns denote respectively the minimum, the maximum bits, the average bits, and the standard deviation used in the SBR encoder among all the frames in the correspondent track. The percentage in each the above category column is the bit percentage for the budget in a frame at bit rate 80 kbps.

80 kbps	Minimum		Maximum		Average		Standard Deviation	
	Bits	Percentage	Bits	Percentage	Bits	Percentage	Bits	Percentage
es01	151	4.29%	423	12.03%	276.02	7.85%	47.69	1.36%
es02	151	4.29%	391	11.12%	256.91	7.30%	51.49	1.46%
es03	151	4.29%	455	12.94%	265.76	7.56%	47.53	1.35%
sc01	151	4.29%	279	7.93%	214.58	6.10%	29.82	0.85%
sc02	151	4.29%	327	9.30%	230.14	6.54%	31.15	0.89%
sc03	151	4.29%	383	10.89%	261.44	7.43%	41.31	1.17%
si01	151	4.29%	399	11.34%	244.71	6.96%	48.09	1.37%
si02	151	4.29%	439	12.48%	267.62	7.61%	57.53	1.64%
si03	151	4.29%	319	9.07%	220.96	6.28%	30.92	0.88%
sm01	151	4.29%	343	9.75%	249.37	7.09%	33.42	0.95%
sm02	151	4.29%	503	14.30%	241.33	6.86%	73.31	2.08%
sm03	151	4.29%	399	11.34%	248.02	7.05%	44.74	1.27%

From the data, the bits used in SBR have small deviation. Also, the percentage of SBR is small compared to AAC encoder. Due to the relatively stable on the bit variation, we could reserve bits for SBR encoder through demand estimator of SBR. Hence the proposed bit reservoir in Figure 53 has not regulated the bits used in a frame, but the bits consumption will be taken into account in the left budget. Although this kind of mechanism may lead to a situation that the bits used may be greater than the allowable budget originally used in the bit reservoir but the exceeding amount is small and can be consumed in the budget of the proceeding frames. So, from the viewpoint of two frames, the budget has been constrained to the allowable amount.

Here we propose three designs for the SBR demand estimators. The first method is to have a fixed estimation. We can just provide a fixed value for the SBR demand estimator. The fixed value can be obtained by given experiment data or tuning results. This method is easy for implementation but may encounter the quality risk from different test tracks sets. The second method is to predict the demand from the consumed SBR bits in the previous one frame. This idea comes from the intuition that the bits consumption in consecutive frames should be similar. The third method is to have the estimation from the average of a period of the previous frames. It is the extension of method 2 but without risks caused by sudden bits attacks through much longer reference length.

5.3 HE-AAC Bit Allocation

For global bit allocation on HE-AAC, the formula applied to AAC in (33) must be modified to consider both AAC encoder and SBR encoder. Hence the allocated bits for whole HE-AAC frame is derived from

$$\text{Allocated_bits_for_HEAAC} = \text{mean_bits}' + R'_{demnd} * \text{mean_bits}' * R_{budgt}, \quad (36)$$

where $\text{mean_bits}'$ is derived from the desired average bit rate for HE-AAC encoder, R_{budget} comes from the budget curve for AAC encoder as depicted in Figure 33, and R'_{demand} is derived from

$$R'_{demnd}(f) = \eta(D'(f)), \quad (37)$$

$$D'(f) = \frac{AE(f) - AE_{average} + B}{AE_{average} + B}, \quad (38)$$

where $\eta(\cdot)$ is the transform function comes from the proposed demand curve for AAC as shown in Figure 30, $AE(f)$ is derived from (24), $AE_{average}$ is obtained through the same strategy as described in (25), and B is calculated by

$$B = AE(f) \frac{SBR_bits}{\text{mean_bits}' + SBR_bits}, \quad (39)$$

where SBR_bits is estimated by our SBR demand estimator. Since the traditional perceptual entropy for SBR encoder is not yet available in current literature, we need to transform SBR_bits from bit domain to entropy domain first. Then the modified demand ratio $D'(f)$ could include the entropy factor B to express the bits requirement of SBR encoder in HE-AAC. Therefore, the bits reserved for SBR encoder is SBR_bits and allocated bits for AAC encoder is

$$\text{Allocated_bits_for_AAC} = \text{Allocated_bits_for_HEAAC} - SBR_bits. \quad (40)$$

The flow chart of the bit reservoir design for HE-AAC is illustrated in Figure 58. Both demand estimator and budget regulator for AAC encoder and SBR encoder are covered in HE-AAC bit reservoir structure.

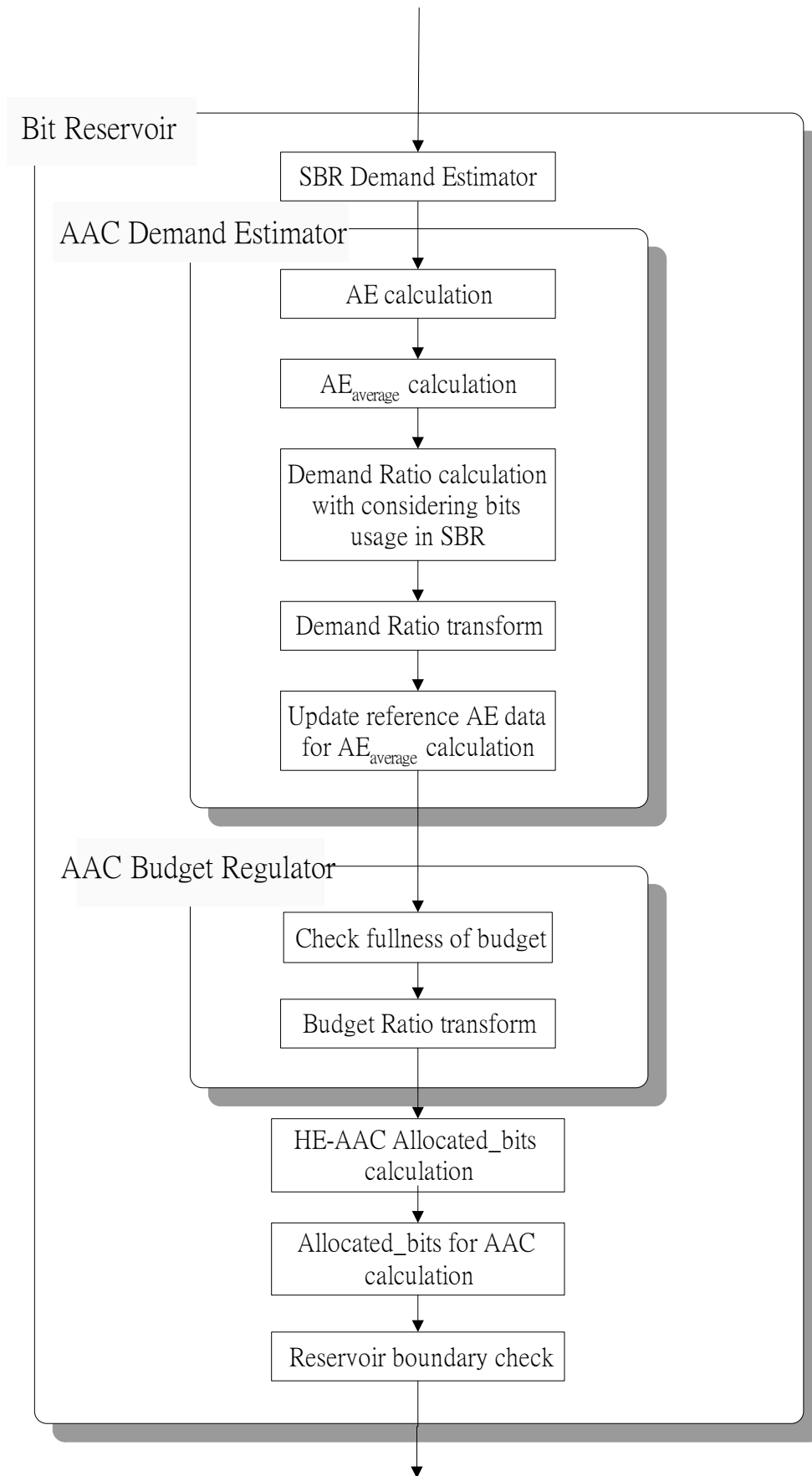


Figure 58: The flow chart of bit reservoir design for HE-AAC.

5.4 Experiments for HE-AAC

This section focuses on quality measurement through objective and subjective methods. We adopt NCTU-HEAAC [25] as platforms for HE-AAC to exhibit the quality enhancement with bit reservoir. There are four primary experiments in this section. The first one is to evaluate the quality of the encoder with and without bit reservoir through objective measurement. The second one is the best parameters decision evaluations. The preceding experiment is based on these well tuning parameters to evaluate the quality. The third one is to prove the robustness of our proposed bit reservoir algorithm through PSPLAB audio database [26] as described in section 4.5.3. The fourth one is to evaluate the audio quality through subjective measurement. Through both objective and subjective tests, the efficiency and quality of our designs are well examined.

5.4.1 Objective quality evaluation for HE-AAC

The following experiments are based on PEAQ system [27]. The detail of PEAQ system is described in section 4.5.1. We first illustrate the objective quality of three designs for the SBR demand estimators in our new bit reservoir for HE-AAC. These three designs are Constant, Previous, and Average method. Constant method uses a fixed estimation, Previous method refers to the consumed bits of previous one frame for estimation, and Average method adopts the average of several previous frames to estimate demand. The detailed description of three designs is listed in section 5.2. Figure 59 - Figure 62 demonstrate the objective quality measurements of three designs at bit rate 48kbps, 64kbps, 80kbps, and 96kbps. In these tests, the constant value for Constant method is set as 325 and the reference length of Average method is set as 25. The parameters evaluating process of three designs is stated in subsection 5.4.2. From the results, the qualities of three designs are similar. But Average method is beneficial for bit rates lower than 96kbps. Hence we choose Average method as our default SBR demand estimator in later experiments for HE-AAC.

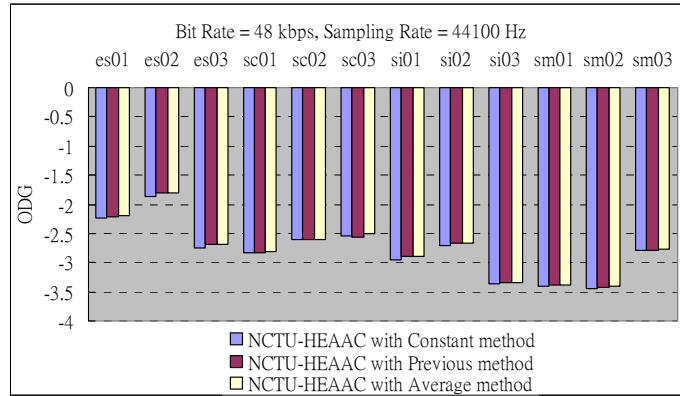


Figure 59: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 48kbps.

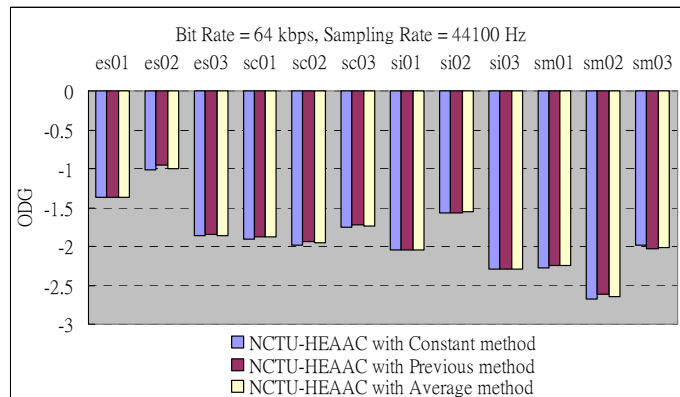


Figure 60: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 64kbps.

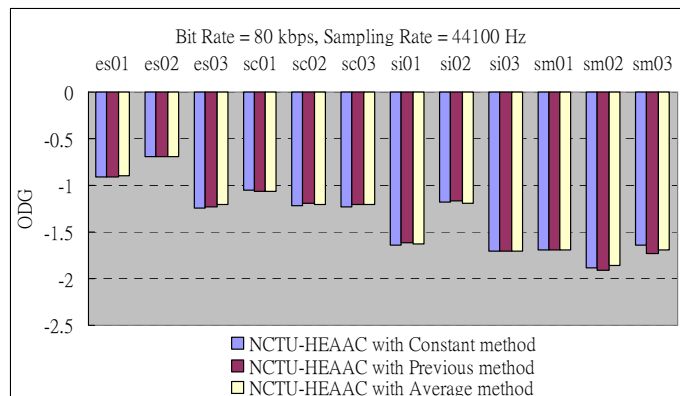


Figure 61: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 80kbps.

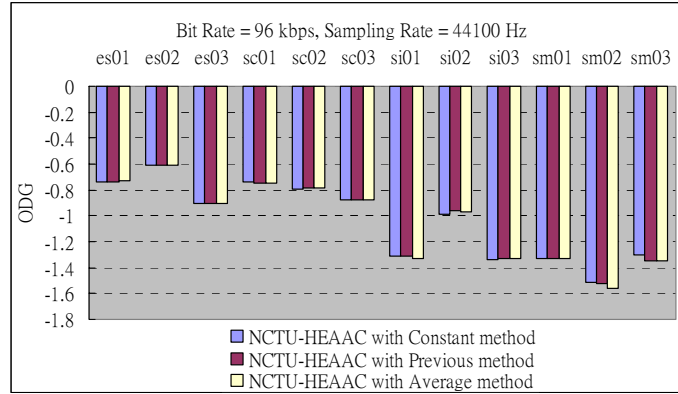


Figure 62: Objective measurements through the ODGs for three kinds of SBR demand estimator designs at 96kbps.

Furthermore, we illustrate the objective quality of HE-AAC with different bit reservoir schemes. In addition to our new design, there are another 2 bit reservoir methods for comparison. The first one is NoReservoir, which just uses the allocated mean bits without accumulating the bits left. The second one is Simple, which only preserves remaining bits from previous one frame to current frame without any managing scheme. The results shown in Table 15 illustrate that the new bit reservoir design could gain improvement on average up to 0.2683 at 48kbps, 0.2858 at 64kbps, 0.2325 at 80kbps, and 0.1367 at 96kbps than those without any bit reservoir controlling.

Table 15: Objective measurements through the ODGs for different bit reservoir designs at different bit rates in HE-AAC (Long/Short window, M/S coding, and TNS).

Bit Rates	48kbps			64kbps		
	1	2	3	1	2	3
es01	-2.51	-2.33	-2.19	-1.56	-1.51	-1.37
es02	-2.67	-2.33	-1.81	-1.27	-1.12	-1
es03	-2.97	-2.82	-2.68	-2.08	-2.02	-1.86
sc01	-3.02	-2.91	-2.82	-2.13	-1.99	-1.88
sc02	-2.66	-2.69	-2.61	-2.25	-2.12	-1.95
sc03	-2.69	-2.62	-2.51	-1.98	-1.85	-1.74
si01	-3.14	-3.05	-2.9	-2.43	-2.26	-2.05
si02	-3.12	-2.9	-2.66	-1.89	-1.63	-1.55
si03	-3.47	-3.39	-3.35	-2.52	-2.36	-2.29
sm01	-3.53	-3.46	-3.38	-2.55	-2.4	-2.24
sm02	-3.62	-3.54	-3.4	-3.06	-2.99	-2.64
sm03	-2.9	-2.82	-2.77	-2.3	-2.22	-2.02

Average	-3.025	-2.905	-2.7567	-2.1683	-2.0392	-1.8825
Bit Rates	80kbps			96kbps		
Coding Methods	1	2	3	1	2	3
es01	-1.02	-1	-0.9	-0.8	-0.77	-0.73
es02	-0.78	-0.71	-0.69	-0.65	-0.62	-0.61
es03	-1.37	-1.29	-1.2	-0.96	-0.92	-0.9
sc01	-1.23	-1.15	-1.06	-0.85	-0.8	-0.75
sc02	-1.41	-1.29	-1.2	-0.89	-0.85	-0.78
sc03	-1.39	-1.29	-1.21	-0.96	-0.92	-0.88
si01	-1.92	-1.78	-1.63	-1.47	-1.4	-1.33
si02	-1.35	-1.18	-1.19	-1.1	-0.98	-0.97
si03	-1.86	-1.75	-1.7	-1.45	-1.35	-1.33
sm01	-1.92	-1.79	-1.69	-1.44	-1.4	-1.33
sm02	-2.51	-2.37	-1.86	-2	-1.87	-1.56
sm03	-2.05	-1.98	-1.69	-1.59	-1.54	-1.35
Average	-1.5675	-1.465	-1.335	-1.18	-1.1183	-1.0433

Sample Rate : 44100 Hz
Coding Method :
1 : NoReservoir; 2 : Simple; 3 : Our New Design

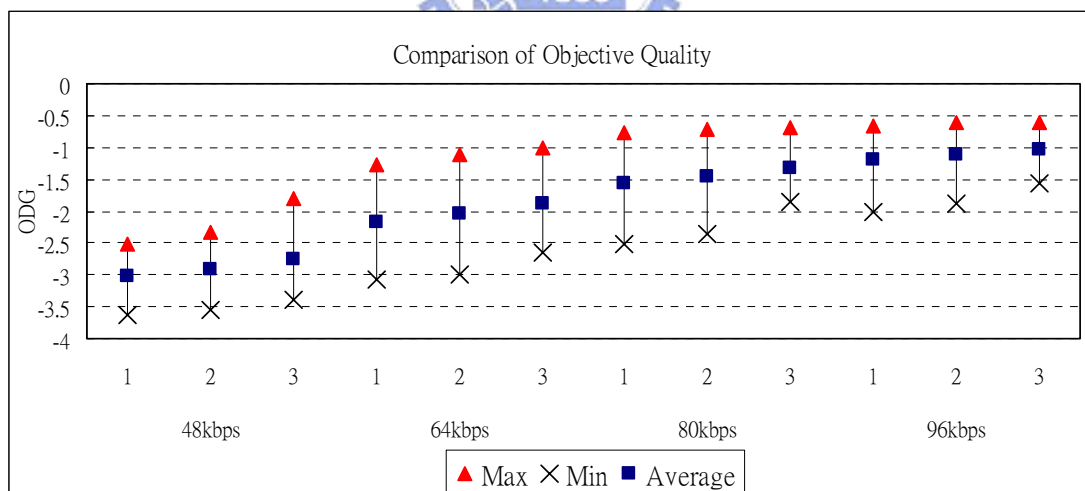


Figure 63: The ODG range comparison of Table 15. The top arrow represents the maximum ODG value, the down cross represents the minimum ODG value, and the middle square represents average ODG value among the twelve test tracks.

5.4.2 Parameter evaluation for HE-AAC

In this section, we evaluate the parameters for HE-AAC mentioning in this thesis. The best parameter combinations are adopted as the default setting for previous and proceeding sections.

In section 5.2, we proposed three designs for SBR demand estimator. In order to compare the quality of different designs, we need to evaluate the fixed estimation value for Constant method and the reference length for Average method respectively. Through following evaluation, the best parameter for each method is chosen as default setting for experiments in this chapter. Although there are several bit rate settings for HE-AAC encoder, we select only 80kbps that is the major discussing bit rate range as our evaluation bit rate to simplify the process. The evaluation results are shown in Table 16 and the best parameter choice is indicated by bold font type.

Table 16: HE-AAC parameters evaluation.

SBR Demand Estimator Design for HE-AAC						
NCTU-HEAAC (Long/Short window, M/S coding, TNS)	Constant method					
	Fixed value	275	300	325	350	375
	Min	-1.88	-1.92	-1.88	-1.9	-1.91
	Max	-0.69	-0.69	-0.69	-0.69	-0.69
	Average	-1.345	-1.345	-1.3392	-1.3425	-1.3442
	Average method					
	Reference length	15	20	25	30	35
	Min	-1.9	-1.89	-1.86	-1.9	-1.91
	Max	-0.69	-0.69	-0.69	-0.68	-0.69
	Average	-1.3392	-1.3375	-1.335	-1.34	-1.3375
	Best Parameter for Constant method : 325					
	Best Parameter for Average method : 25					
	Bit Rate : 80kbps					
Sample Rate : 44100 Hz						

From the results in Table 16, we choose *Fix value* = 325 and *Reference length* = 25 as our default parameters setting in subsection 5.4.1 and 5.4.3.

5.4.3 Objective quality measurement based on music database for HE-AAC

In order to verify the robustness of the proposed new bit reservoir design in HE-AAC and evaluate the possible risk for a variety of music categories, we adopt PSPLAB audio database [26] as our testing material. The database includes 327 tracks that are separated into 16 sets with different signal properties as shown in Table 10.

The average ODG result of NCTU-HEAAC without bit reservoir and with new bit reservoir at different bit rates for the 16 bitstream sets in PSPLAB audio database is illustrated through Figure 64 to Figure 67. On average, our new bit reservoir design could gain ODG improvement up to 0.2073 at 48 kbps, 0.2038 at 64 kbps, 0.1711 at 80 kbps, and 0.1037 at 96 kbps than without applying bit reservoir scheme. Figure 68 and Figure 69 illustrate the enhancement and degradation tracks distribution for 327 tracks at different bit rates. The enhancement ratio is up to 97.2 % at 48kbps, 98.4% at 64kbps, 96% at 80kbps, and 99.7% at 96kbps. From these results, our new bit reservoir design for HE-AAC is truly beneficial for great part of audio tracks.

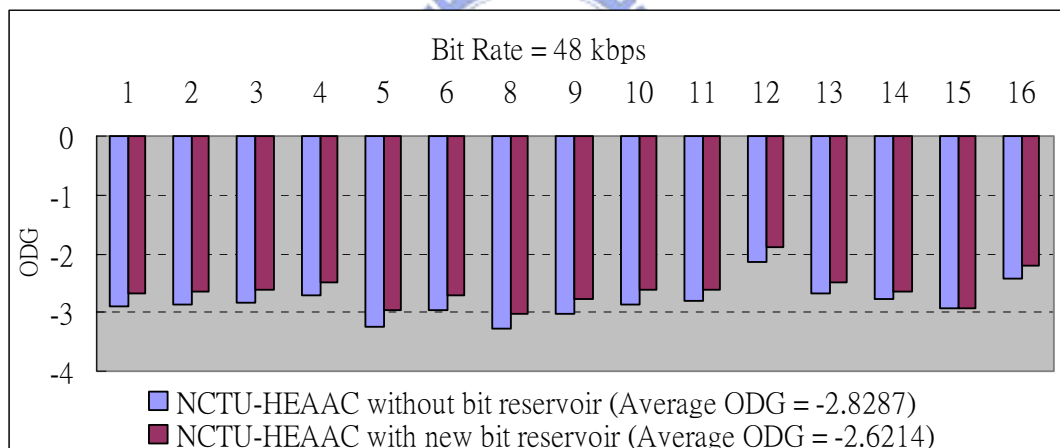


Figure 64: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 48 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

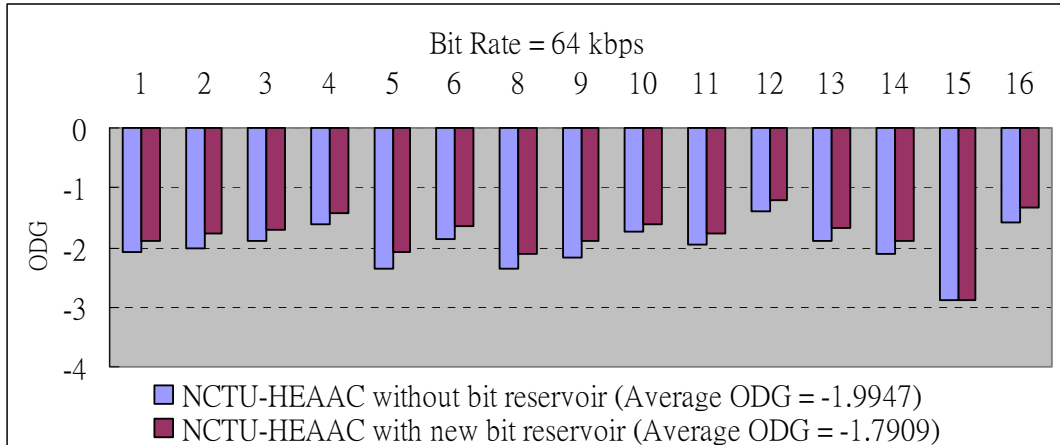


Figure 65: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 64 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

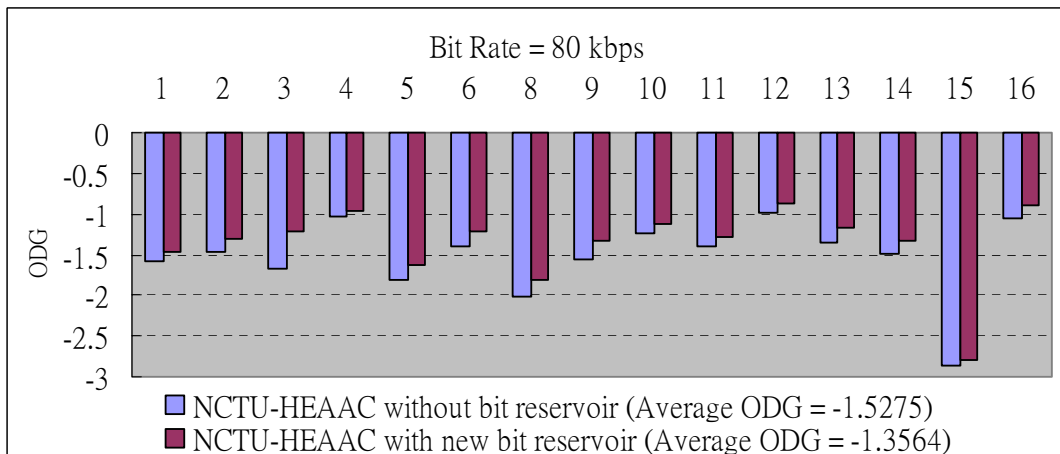


Figure 66: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 80 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

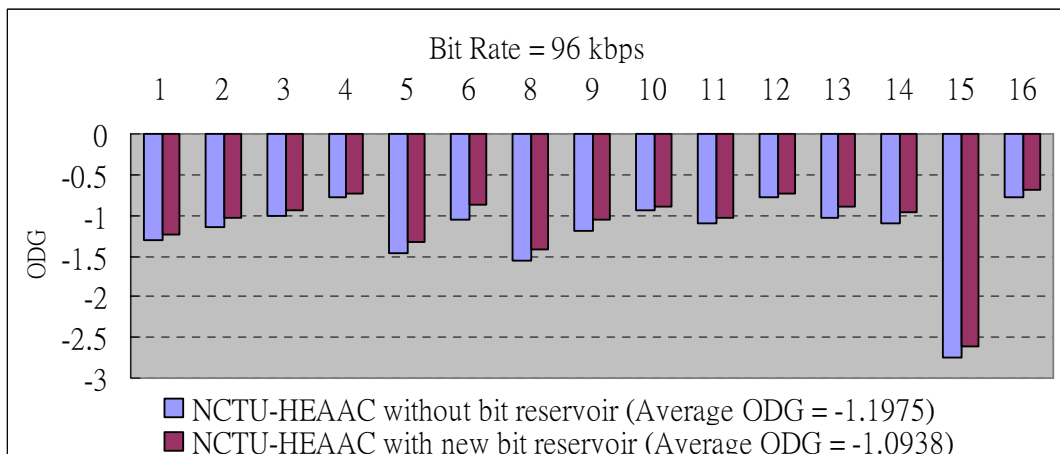


Figure 67: The average objective quality of NCTU-HEAAC without bit reservoir and with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 96 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

with new bit reservoir for the 16 bitstream sets in PSPLAB audio database. Bit rate: 96 kbps; Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

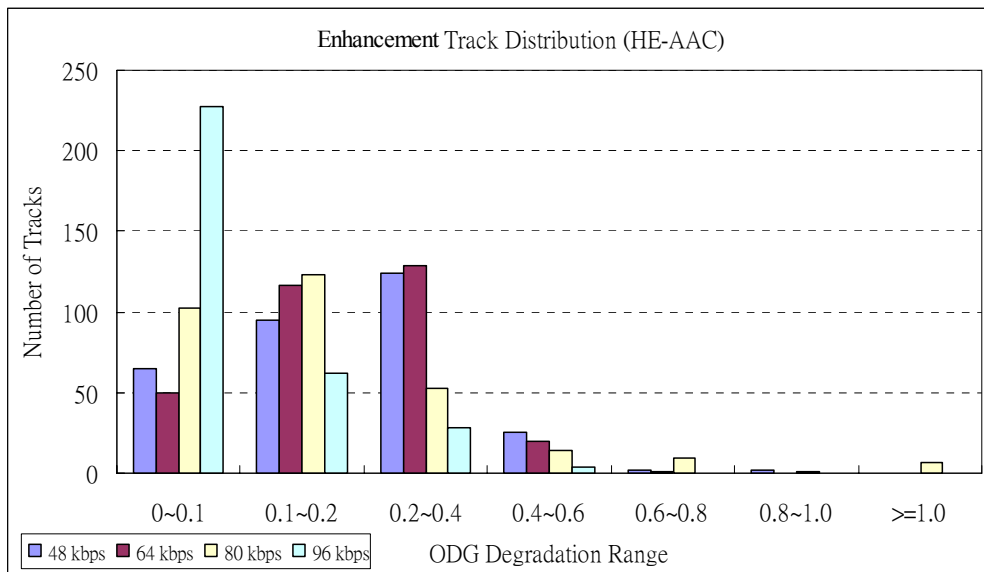


Figure 68: The enhancement tracks distribution of NCTU-HEAAC without bit reservoir and with new bit reservoir at different bit rates for the 16 bitstream sets in PSPLAB audio database. Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

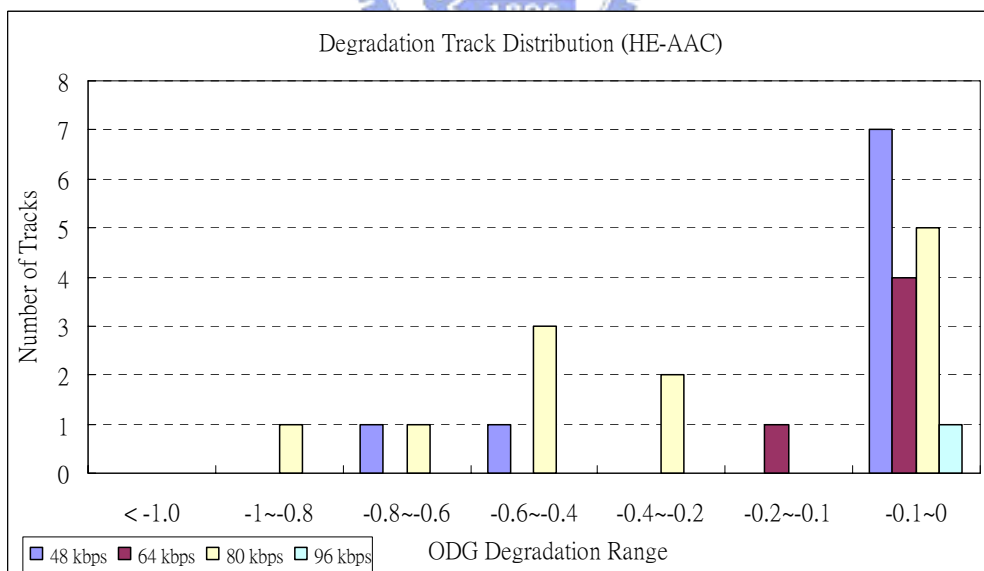


Figure 69: The degradation tracks distribution of NCTU-HEAAC without bit reservoir and with new bit reservoir at different bit rates for the 16 bitstream sets in PSPLAB audio database. Sample rate: 44100 Hz (Long/Short window, M/S coding, TNS).

5.4.4 Objective quality measurement with existing codecs

In this section, we make a summary of our encoder with new bit reservoir design by comparing with other existing HE-AAC encoders. We compare our NCTU-HEAAC [25] with Coding Technologies 7.0.5 [30] and Nero 6.6.0.8 [29] at different low bit rates. The experiment results for HE-AAC are listed in Table 17. Through observing these comparison results, our HE-AAC encoder with new bit reservoir design is superior to other encoders in average.

Table 17: Objective quality comparison for HE-AAC at different bit rates.

Bit Rates	48kbps			64kbps		
Encoders	NCTU-HEAAC	Coding Technologies 7.0.5	Nero 6.6.0.8	NCTU-HEAAC	Coding Technologies 7.0.5	Nero 6.6.0.8
es01	-2.19	-2.24	-1.84	-1.37	-1.02	-1.54
es02	-1.81	-2.65	-2.49	-1	-1.49	-2.27
es03	-2.68	-2.41	-2.64	-1.86	-1.1	-2.18
sc01	-2.82	-2.65	-2.9	-1.88	-1.65	-2.21
sc02	-2.61	-3.1	-2.72	-1.95	-2.33	-1.83
sc03	-2.51	-3.05	-2.87	-1.74	-1.62	-1.84
si01	-2.9	-3.42	-3.61	-2.05	-2.83	-2.61
si02	-2.66	-2.73	-3.4	-1.55	-1.56	-1.84
si03	-3.35	-2.86	-3.82	-2.29	-2.29	-2.76
sm01	-3.38	-3.61	-3.83	-2.24	-2.6	-3.01
sm02	-3.4	-3.31	-3.39	-2.64	-2.79	-2.25
sm03	-2.77	-3.14	-3.02	-2.02	-1.7	-2.1
Average	-2.7567	-2.9308	-3.0442	-1.8825	-1.915	-2.2033

Bit Rates	80kbps			96kbps		
Encoders	NCTU-HEAAC	Coding Technologies 7.0.5	Nero 6.6.0.8	NCTU-HEAAC	Coding Technologies 7.0.5	Nero 6.6.0.8
es01	-0.9	-0.79	-1.42	-0.73	-0.69	-1.34
es02	-0.69	-0.94	-2.07	-0.61	-0.83	-1.95
es03	-1.2	-0.88	-2.36	-0.9	-0.78	-2.26
sc01	-1.06	-1.06	-1.28	-0.75	-0.71	-0.88
sc02	-1.2	-1.53	-1.24	-0.78	-1.03	-1.02
sc03	-1.21	-1.15	-1.19	-0.88	-0.86	-0.95
si01	-1.63	-2.3	-1.94	-1.33	-1.79	-1.55

si02	-1.19	-1.01	-1.71	-0.97	-0.82	-1.51
si03	-1.7	-1.99	-2.19	-1.33	-1.5	-1.45
sm01	-1.69	-1.85	-2.15	-1.33	-1.58	-1.45
sm02	-1.86	-2.36	-1.65	-1.56	-2.04	-1.43
sm03	-1.69	-1.27	-1.37	-1.35	-0.98	-0.97
Average	-1.335	-1.4275	-1.7142	-1.0433	-1.1342	-1.3967

Sample Rate : 44100 Hz

Encoder:

NCTU-HEAAC: with Long/Short window, M/S coding, TNS, and our new bit reservoir design.

Coding Technologies 7.0.5 : -bsformat mp4 -sig 1 -chmode stereo.

Nero 6.0.8 : HE AAC, CBR, Stereo, High Quality.

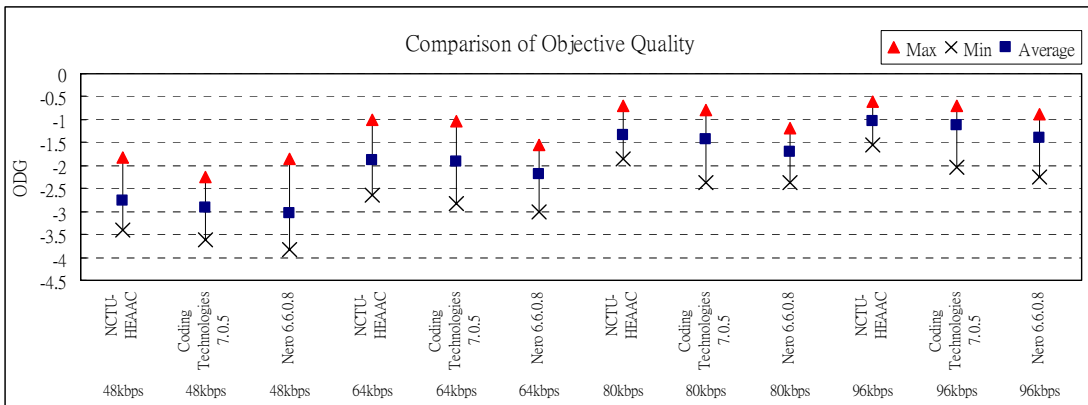


Figure 70: The ODG range comparison of Table 17.

5.4.5 Subjective quality evaluation for HE-AAC

The audio quality of an encoder can be considered as the perceived difference between the output of the testing encoder and a known reference signal. Traditional quality measurement such as the signal to noise ratio provides simple, objective measures of audio quality but they ignore psychoacoustic effects that can lead to large differences in perceived quality. Because human listeners are the ultimate judges of quality in any application, the formal listening tests are required to assess audio quality when a highly accurate assessment is needed [12].

In order to assess the audio quality, a grading scale is used. The grading scale used in ITU-R BS.1116 [31] listening tests is based on the five-stage impairment scale as defined by ITU-R BS.562-3 [32] and shown in Figure 71. The ratings in BS.1116 are represented on a continuous scale between grades of 5.0 for transparent coding down to 1.0 for highly annoying impairments [12].

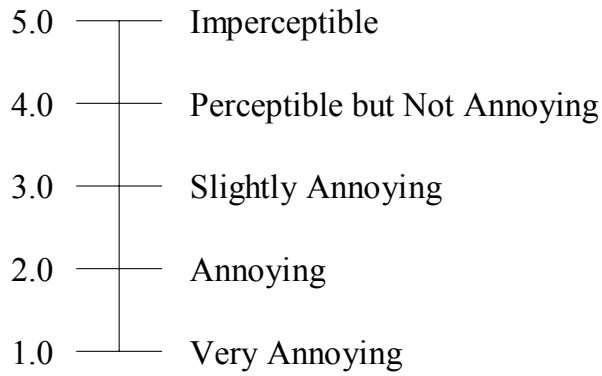


Figure 71: ITU-R five-grade impairment scale.

The most widely accepted manner for testing systems with small impairment is the so-called “double-blind, triple-stimulus with hidden reference” method. In this method the listener is presented with three signals: the reference signal, R, and the test signals A and B. One of the two test signals will be identical to the reference signal and the other one will be the coded signal. The “double-blind” is performed through neither the listener nor the administrator should know beforehand which test signal is presented just now. The assignments of signals A and B should be done randomly so that neither the test administrator nor the subject has any basis to predict which test signal is the coded one [12].

The listener is asked to assess the impairments of A compared to R, and of B compared to R according to the grading scale. Since one of the stimuli is actually the reference signal, one of them should receive a grade equal to five while the other stimulus may receive a grade that describes the listener’s assessment of the impairment. The resolution achieved by the listening test is reflected in the confidence interval. This interval contains the SDG values [12] with a specified degree of confidence, $1-\alpha$, where α represents the probability that inaudible differences are labeled as audible. A value of 0.05 is chosen for α in practice, which corresponds to a 95% confidence interval.

The “double-blind, triple-stimulus with hidden reference” method has been employed worldwide for many formal listening tests of perceptual audio codecs. The consensus is that it provides a very sensitive, accurate, and stable way for assessing small impairments in audio systems [12].

5.4.5.1 MUSHRA

The “double-blind, triple-stimulus with hidden reference” method has been implemented in different ways. For reliable and repeatable measure of the audio quality of intermediate-quality signals, a method termed MUSHRA (Multiple

Stimulus with Hidden Reference and Anchors) is proposed. It was recently recommended by the ITU-R [33]. MUSHRA is a “double-blind multi-stimulus” test method with hidden reference and one or more hidden anchors [12]. MUSHRA has the advantage that it provides an absolute measure of the audio quality of a codec, which can be compared directly with the reference, i.e. the original audio signal as well as the anchors [34].

According to the MUSHRA guidelines, the listening subjects are required to grade the stimuli based on a continuous quality scale that is divided in five equal intervals labeled excellent, good, fair, poor and bad. The scores are then normalized in the range between 0 and 100 where 0 corresponds to the bad quality of the scale. Then the data analysis is performed as the average across subjects of the differences between the score associated to the hidden reference and the score associated to each other stimulus. Typically a 95% confidence interval is utilized in MUSHRA.

Here we use ABC/Hidden Reference Audio Comparison Tool [35] as the testing environment for our listening test in this thesis. The main dialog box of ABC/Hidden Reference Audio Comparison Tool is shown in Figure 72.

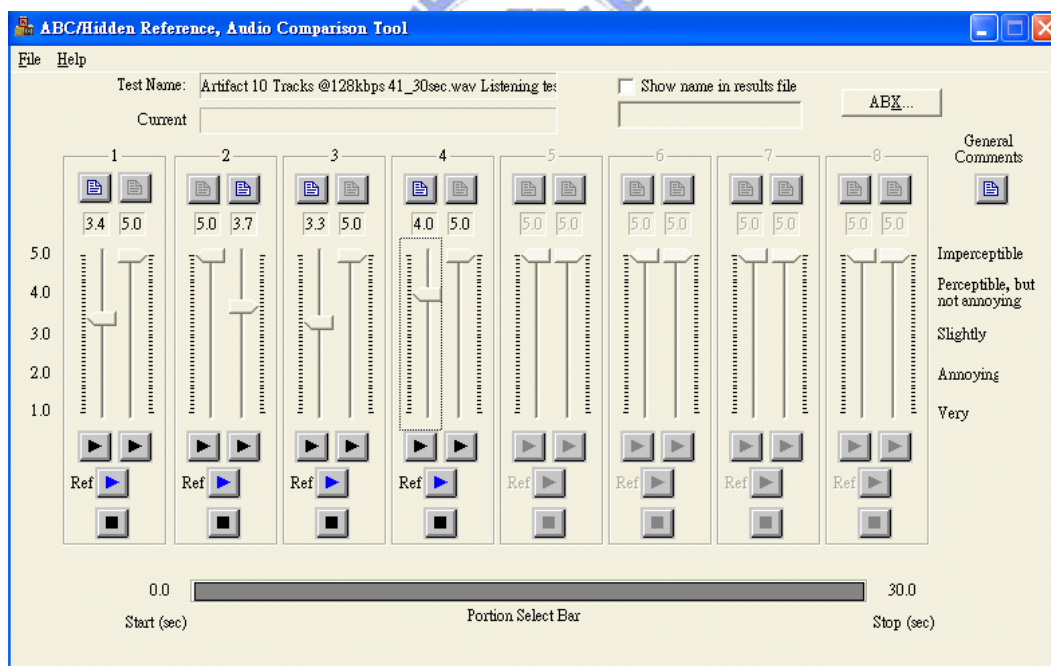


Figure 72: Main dialog box of ABC/Hidden Reference Audio Comparison Tool.

The listener first fills out a setup dialog box or loads a configuration file about tested sample. Each group consists of two files: the reference file and a sample. The listener doesn't know which is which because the program has shuffled them within the group. The groups themselves have been also shuffled, so that the listener doesn't know which sample has been assigned to which group. For each group, the listener

listens to both the left and right sides, and to the reference (labeled and colored blue). Then he/she rates what he/she thinks is the sample. Once the listener has moved the slider he/she is allowed to write a comment describing the degradation he/she heard. The bar at the bottom allows the listener to choose a portion of the file to listen to. After rating all files in the test, the listener saves the results to file. The program reveals which sample was assigned to which group inside the results file [35].

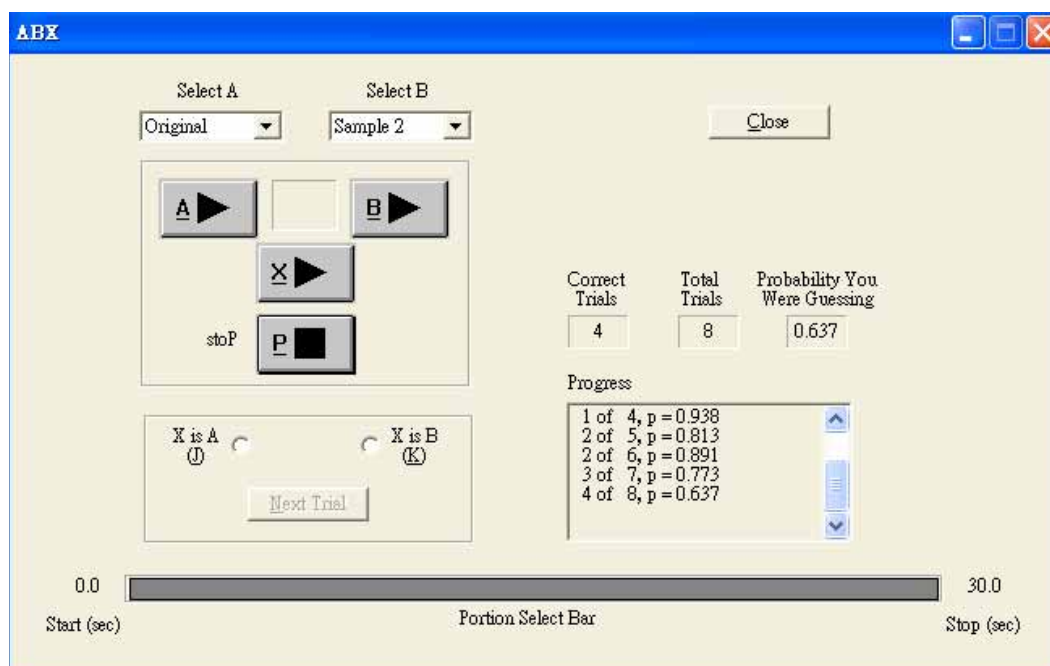


Figure 73: ABX dialog box of ABC/Hidden Reference Audio Comparison Tool.

Before rating through above steps, we should evaluate the confidence level of listener first. The ABX dialog box of ABC/Hidden Reference Audio Comparison Tool is shown in Figure 73. The ABX mode is meant to allow the listener to identify subtle differences between any two samples. Typically, one of the samples will be the original file. One of the samples is assigned to be ‘A’ and the other one is assigned to be ‘B’. Either sample A or sample B is randomly assigned to be ‘X’. The listener must determine whether X is the same as A or B. By performing this test many times, the listener can show with a specified level of confidence that his/her results are not by chance alone [35].

5.4.5.2 Results of listening test

In order to illustrate the consistency between objective and subjective measure, we choose the default test tracks in Table 2 as our samples. There are ten subjects selected to carry out this listening test. The results of subjects with a specified level of confidence are collected. The result of NCTU-HEAAC [25] with and without bit

reservoir control is shown in Figure 74. In general the quality of NCTU-HEAAC with new bit reservoir control is better than without bit reservoir control, especially in transient and harmonic signals, e.g. si01, sm02, sm03. From this experiment, we could conclude that the subjective quality of HE-AAC is consistent with objective quality in Table 15.

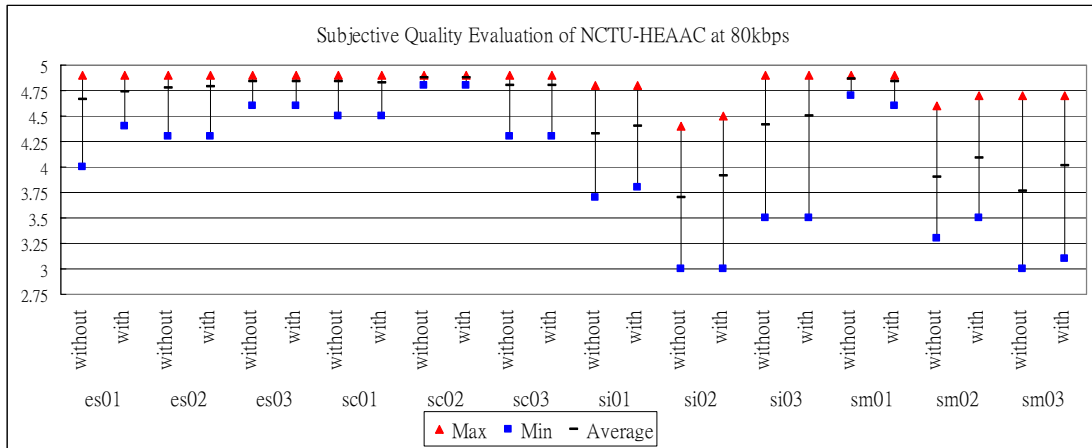


Figure 74: Subjective quality evaluation of NCTU-HEAAC with and without new bit reservoir design at bit rate 80kbps.



Chapter 6

Conclusion

This thesis has proposed a new bit reservoir design for MP3 and AAC to enhance compression quality and maintain bit rate constraint. We not only combine the demand-driven and budget-driven approaches but also consider the design through two novel modules: the demand estimator and the budget regulator. The demand estimator calculates Allocation Entropy, average AE in the past and demand ratio, and transforms the ratio by demand curve. The budget regulator utilizes the flexible abstract budget buffer to decide available bits via budget curve. Therefore, we could adaptively predict the bits required to achieve a specific quality and control the bits budget according to preferred scenario.

We also extend the bit reservoir design in AAC to HE-AAC through the SBR demand estimator and the global budget regulator. Our single iteration design reduces the complexity come from the interdependent issue of bits distribution between AAC encoder and SBR encoder. Two features of our algorithm are: using one common budget regulator while two demand estimators for AAC and SBR encoders, and leaving budget for SBR without directly regulating the encoded bits in SBR.

Objective experiments based on the recommendation system by ITU-R Task Group 10/4 have been conducted on intensive tracks to prove the quality improvement of our new bit reservoir design. Through both subjective and objective measure on tremendous music database, the quality and efficiency of our new design is verified. These experiments have shown that our bit reservoir design in MP3, AAC, and HE-AAC could well fit the encoders for various bit rates and preferred scenario.

References

- [1] ISO/IEC JTC1/SC2/WGII MPEG, International Standard ISO 11172-3 “Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s.”
- [2] ISO/IEC 14496-3:1999, “Information Technology–Coding of Audiovisual objects, Part3: Audio.”
- [3] ISO/IEC, “Text of ISO/IEC 14496-3:2001/FDAM1, Bandwidth Extension,” ISO/IEC JTC1/SC29/WG11/N5570, March 2003, Pattaya, Thailand.
- [4] M. Dietz, L. Liljeryd, K. Kjörling, O. Kunz, “Spectral Band Replication, a novel approach in audio coding,” at the 112th AES Convention, Munich, May 10–13, 2002.
- [5] M. Wolters, K. Kjörling, D. Himm, H. Purnhagen, “A closer look into MPEG-4 High Efficiency AAC,” at the 115th AES Convention, New York, USA, October 10–13, 2003.
- [6] H.W. Hsu, C.M. Liu, and W.C. Lee, “Audio Patch Method in MPEG-4 HE-AAC Decoder,” at the 117th AES Convention, San Francisco, USA, October 28–31, 2004.
- [7] J.D. Johnston, “Estimation of Perceptual Entropy Using Noise Masking Criteria,” *ICASSP*, 1988, pp.2524-2527.
- [8] E. Zwicker and H. Fastl, *Psychoacoustics Facts and Models*, Berlin, Germany: Springer-Verlag, 1990.
- [9] H. Fletcher, “Auditory Patterns,” *Rev. Mod. Phys.*, Vol. 12, pp. 47–65, Jan. 1940.
- [10] E. Terhardt, “Calculating Virtual Pitch,” *Hearing Res.*, Vol. 87, pp. 155-182, 1979.
- [11] T. Painter and A. Spanias, “Perceptual Coding of Digital Audio,” *Proceedings of the IEEE*, Vol. 88, No. 4, pp.451-515, April 2000.
- [12] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards*, Kluwer Academic Publishers, 2003.
- [13] J.D. Johnston, “Transform coding of audio signals using perceptual noise criteria,” *IEEE J. Select. Areas Commun.*, vol. 6, Feb. 1988, pp.314-323.

- [14] LAME, website <http://www.mp3dev.org/mp3> .
- [15] FAAC, website <http://www.audiocoding.com> .
- [16] C.M. Liu, L.W. Chen, H.W. Hsu, and W.C. Lee, "Bit Reservoir Design for HE-AAC," at the 118th AES Convention, Barcelona, Spain, May 28~31, 2005.
- [17] 3GPP TS 26.410 V6.0.0, website <http://www.3gpp.org> .
- [18] C.M. Liu, W.C. Lee, and Y.H. Hsiao, "M/S Coding Based on Allocation Entropy," *Digital Audio Effect*, London, UK, September 8-11, 2003.
- [19] C.M. Liu, W.J. Lee, and R.S. Hong, "A Bandwidth-Proportional Noise-Shaping Criterion and the Associated Fast Bit Allocation Method for Audio Coding," *Digital Audio Effect (DAFX-02)*, Sep. 2002, pp. 26-28.
- [20] NCTU-MP3, website <http://psplab.csie.nctu.edu.tw/projects/index.pl/nctu-mp3.html> .
- [21] NCTU-AAC, website <http://psplab.csie.nctu.edu.tw/projects/index.pl/nctu-aac.html> .
- [22] Yo-Hua Hsiao, "M/S Coding Enhancement in MP3 and AAC," *CSIE Master Thesis of NCTU*, 2004.
- [23] Kan-Yan Peng, "Design of Window Switch Method in AAC," *CSIE Master Thesis of NCTU*, 2004.
- [24] Tzu-Wen Chang, "Temporal Noise Shaping Design for MPEG 4 Advanced Audio Coding," *CSIE Master Thesis of NCTU*, 2004.
- [25] NCTU-HEAAC, website <http://psplab.csie.nctu.edu.tw/projects/index.pl/nctu-heaac.html> .
- [26] PSPLAB audio database, website <http://psplab.csie.nctu.edu.tw/projects/index.pl/testbitstreams.html> .
- [27] ITU Radiocommunication Study Group 6, "Draft Revision to Recommendation ITU-R BS.1387- Method for objective measurements of perceived audio quality".
- [28] QuickTime, website <http://www.apple.com/quicktime>.
- [29] Nero, website <http://www.nero.com>.
- [30] Coding Technologies, aacPlusEval v2 Evaluation Package, Version 7.0.5, website <http://portal.codingtechnologies.de/eval/aacPlusEval/>.
- [31] International Telecommunications Union, Radiocommunication Sector BS.1116 (rev. 1), "Methods for the Subjective Assessment of Small Impairment in Audio

Systems Including Multichannel Sound Systems”, Geneva 1997.

[32] International Telecommunications Union, Radiocommunication Sector BS.562-3, “Subjective Assessment of Sound Quality”, Geneva 1978-1982-1984-1990.

[33] International Telecommunications Union, Radiocommunication Sector BS.1534, “Method for the Subjective Assessment of Intermediate Quality Level Coding Systems – General requirements”, Geneva 2001.

[34] G. Stoll and F. Kozamernik, “EBU Listening Tests on Internet Audio Codecs”, EBU TECHNICAL REVIEW, June. 2000.

[35] ABC/Hidden Reference Audio Comparison Tool, website
<http://ff123.net/abchr/abchr.html>.

