

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

適用於助聽器之低功率噪音消除設計

Low Power Noise Reduction Design for
Hearing Aids Application

研究生：蔡政君

指導教授：張添烜

中華民國 九十八年 九月

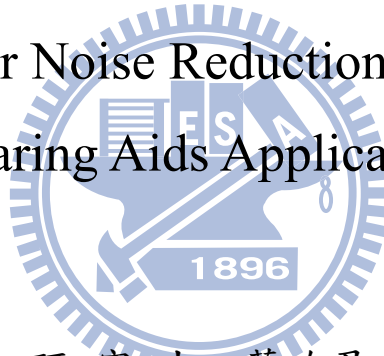
國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

適用於助聽器之低功率噪音消除設計

Low Power Noise Reduction Design for
Hearing Aids Application



研究生：蔡政君

指導教授：張添烜

中華民國 九十八年 九月

適用於助聽器之低功率噪音消除設計

Low Power Noise Reduction Design for
Hearing Aids Application

研究生：蔡政君

Student: Cheng-Chun Tsai

指導教授：張添烜 博士

Advisor: Tian-Sheuan Chang



A Thesis
Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in
Electronics Engineering
September 2009
Hsinchu, Taiwan

中華民國 九十八年 九月

適用於助聽器之低功率噪音消除設計

研究生：蔡政君

指導教授：張添烜 博士

國立交通大學

電子工程學系電子研究所

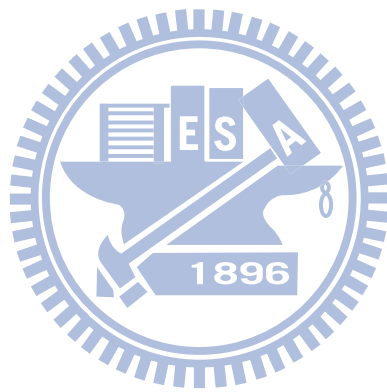
摘要

噪音消除是助聽器中的關鍵問題。為了補償患者的聽力損失，助聽器需要對輸入聲音加以放大，如此一來必需要以噪音消除設計來增進在噪音環境下的聲音品質和辨識度。在整合的助聽器系統中，為了延長電池使用壽命及最小化系統的體積，我們需要低功率的設計。

在此論文中，我們提出一套適用於助聽器的低功率噪音消除設計，其中包含了以熵值為基礎的語音偵測，及以濾波器組為基礎的頻域刪減。以熵值為基礎的語音偵測可在噪音環境下區分該時段是語音訊號或是沉默區間。以 filter bank 為基礎的頻域刪減估計噪音量值，並根據以熵值為基礎的語音偵測之結果做不同的頻域刪減。關閉機制在噪音量值低於一固定閾值時，停止頻域刪減之作動以節省耗電。透過降低運算複雜度，此演算法針對低功率的硬體設計作了最佳化設計。從實驗結果可以得知，平均區段噪訊比增進了 6.27dB。PESQ 分數則平均增進了 0.316 分。

最後此演算法在聯華電子 90 奈米 CMOS 製程下完成硬體實現。工作頻率為 6 百萬赫茲。為了節省面積及耗電，我們採用折疊硬體設計。基於資料存儲之需要，我們使用了 1.536 千位元組的靜態隨機存取記憶體。若包含靜態隨機存取記憶體，估計需要的邏輯閘約為 101,697 個。如不包含靜態隨機存取記憶體，則估計需要

的邏輯閘約為 80,628 個。耗電量則為 2.927×10^{-4} 瓦。



Low Power Noise Reduction Design for Hearing Aids Application

Student: Cheng-Chun Tsai

Advisor: Tian-Sheuan Chang

Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University

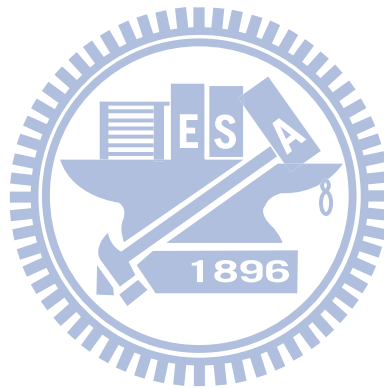
Abstract

For hearing aids application, the amplification of input sound is needed in order to compensate the hearing loss of the patient. Thus noise reduction is required to improve speech quality and intelligibility under noisy environments. For integrated hearing aids system, low-power design is necessary such that the battery life can be expended and the system volume can be minimized.

In this thesis, we propose a low power noise reduction design for hearing aids application with entropy-based voice activity detection and filter bank-based spectral subtraction. The entropy-based voice activity detection distinguishes the speech period from the silence period in noisy environment and makes the decision whether it is voice active or not. The filter bank-based spectral subtraction estimates noise level and performs different spectral subtraction schemes based on the result carried out by the entropy-based voice activity detection. Off mechanism turns off the spectral subtraction process if noise level lies below a fixed threshold in order to reduce power consumption. The proposed algorithm is optimized for low power hardware design by minimizing the calculation complexity. From simulation results,

the average segment SNR improvement is 6.27dB and the average PESQ score is elevated by 0.316.

The final design is implemented by UMC 90nm CMOS technology with high V_T cell library. The clock frequency is 6MHz. For the hardware architecture, folding technique is adopted to save area and to reduce power consumption. For data storage, 1.536K Bytes of SRAM is utilized. The total estimated gate count is 101,697 including SRAM and 80,628 excluding SRAM. The total power consumption is 292.7 μ W.



誌 謝

首先，要感謝我的指導教授—張添烜博士，在我就讀電子所這兩年來，給予的支持和鼓勵，並引領我以正確的態度來面對與解決問題。在研究方面讓我自由的發揮，並在我遭遇瓶頸與困難時給予建議與協助。感謝老師提供豐富的實驗室資源，使我不但能充分的利用新穎的軟硬體設備來做研究，亦提供了一個溫馨舒適的環境，在修課與努力用功時皆無後顧之憂。此外也感謝老師給我機會參與多個國內外學術研討會，大大增廣我的視野。因此，我對與老師的感激之情難以言述。也謝謝我的口試委員們，交大電機冀泰石老師與清大電機王小川老師。感謝你們百忙中抽空來指導我，因為你們寶貴的意見，讓我的論文更加完備。

感謝張老師共同指導的魏誠文學長，除了在研究上毫無保留地提供經驗分享外，也對我人生歷練及方向有著正面的幫助及影響。認識學長是我念電子所最幸運的收穫。感謝 VSP 實驗室的好伙伴們，孟維、博淵，和你們一起打球是我念交大以來最有趣的回憶，不論在研究或是生活上，我們都是最好的同學。還有筱珊同學、之悠同學，一起修課、一起做研究、一起出遊的革命情感，亦是人生難忘的一頁。感謝李國龍學長、曾宇晟學長、王國振學長，不論在研究或生活上都給我很多鼓勵與導引。還有已畢業的張彥中學長、林佑昆學長、詹景竹學長、蔡宗憲學長、張瑋城學長、戴瑋呈學長，在我做研究時的熱心幫忙，或找工作時的經驗分享，都讓我感到非常的溫馨，且受用無窮。另外也感謝實驗室的學弟妹們，廖元歆、許博雄、陳宥辰、陳奕均、洪瑩蓉，能和你們在同一間實驗室，真的相當開心。感謝郭羽庭學長、張國強學長、鍾譯賢同學，及參與助聽器計劃的老師與同學們，沒有你們，就沒有我的研究成果。

最後要感謝默默支持我的家人們，我的爸爸媽媽、弟弟，以及女友。你們真心的支持、包容與關懷，是讓我能堅持到最後一刻的動力。

在此，把本論獻給所有愛我與所有我愛的人。

Contents

Chapter 1.	Introduction.....	1
1.1.	Background.....	1
1.2.	Motivation and Contribution.....	2
1.3.	Thesis Organization.....	2
Chapter 2.	Related Work.....	3
2.1.	Overview.....	3
2.2.	Spectral-Subtractive Method.....	4
2.2.1.	Introduction.....	4
2.2.2.	Voice Activity Detection.....	5
2.2.3.	Spectral Subtraction.....	5
2.3.	Summary.....	6
Chapter 3.	Noise Reduction Algorithm with Entropy-Based Voice Activity Detection and Filter Bank-Based Spectral Subtraction.....	7
3.1.	Overview.....	7
3.2.	Introduction to the Hearing Aid System.....	8
3.3.	ANSI S1.11 Filter Bank for Digital Hearing Aids[26].....	9
3.4.	Introduction to the Proposed Noise Reduction Algorithm.....	11
3.5.	Entropy-Based Voice Activity Detection.....	13
3.5.1.	Entropy in Speech Processing.....	13
3.5.2.	Entropy Calculation.....	15
3.5.3.	Low-Power Hardware Optimizations for Entropy Calculation...	18
3.5.4.	Adaptive Thresholding.....	24

3.6.	Filter Bank-Based Spectral Subtraction.....	27
3.6.1.	Introduction.....	27
3.6.2.	Noise Estimation.....	28
3.6.3.	Spectral Attenuation for Noise.....	29
3.6.4.	Spectral Subtraction for Speech.....	30
3.6.5.	Low-Power Hardware Optimizations for Filter Bank-Based Spectral Subtraction.....	32
3.6.6.	Off Mechanism.....	33
3.6.7.	Data Output.....	33
3.7.	Summary.....	33
Chapter 4.	Simulation and Analysis.....	35
4.1.	Simulation Settings.....	36
4.2.	Experimental Result and Analysis.....	38
4.2.1.	Influence of Different Original Segment SNR.....	74
4.2.2.	Influence of Different Noise Type.....	74
4.2.3.	Influence of Different Silence Period Length before Speech.....	87
4.3.	Comparison with Different Algorithms.....	96
4.4.	Average Segment SNR Improvement and Average PESQ Score Improvement.....	97
4.5.	Summary.....	97
Chapter 5.	Hardware Implementation.....	99
5.1.	Architecture Design.....	99
5.2.	Implementation Result.....	103
Chapter 6.	Conclusion and Future Work.....	106
6.1.	Conclusion.....	106

6.2.	Future Work	106
Reference	108
作者簡歷	111

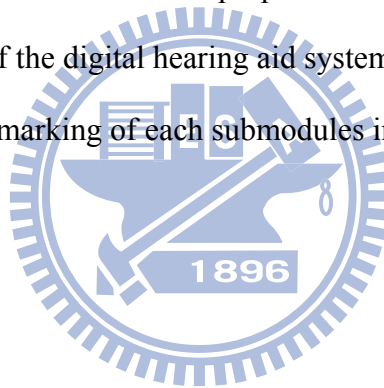


List of Figures

Fig. 3-1 Functional block diagram of the hearing aid system.....	8
Fig. 3-2 Block diagram of the digital system.....	8
Fig. 3-3 Flow of the proposed algorithm	11
Fig. 3-4 Sine wave with period T	14
Fig. 3-5 Gaussian noise.....	14
Fig. 3-6 Scheme for input sequence window gathering.....	17
Fig. 3-7 Scheme for window averaging.....	17
Fig. 3-8 Scheme for window averaging by register refreshing.....	19
Fig. 3-9 Example of adaptive thresholding.....	27
Fig. 3-10 State decision after VAD.....	28
Fig. 4-1 VAD result for white noise.....	39
Fig. 4-2 VAD result for babble noise.....	39
Fig. 4-3 VAD result for factory noise	40
Fig. 4-4 VAD result for car noise.....	40
Fig. 4-5 Segment SNR improvement and PESQ scores	42
Fig. 4-6 Segment SNR improvement and PESQ scores	43
Fig. 4-7 Segment SNR improvement and PESQ scores	44
Fig. 4-8 Segment SNR improvement and PESQ scores	45
Fig. 4-9 Segment SNR improvement and PESQ scores	46
Fig. 4-10 Segment SNR improvement and PESQ scores	47
Fig. 4-11 Segment SNR improvement and PESQ scores	48
Fig. 4-12 Segment SNR improvement and PESQ scores	49
Fig. 4-13 Segment SNR improvement and PESQ scores	50

Fig. 4-14 Segment SNR improvement and PESQ scores	51
Fig. 4-15 Segment SNR improvement and PESQ scores	52
Fig. 4-16 Segment SNR improvement and PESQ scores	53
Fig. 4-17 Segment SNR improvement and PESQ scores	54
Fig. 4-18 Segment SNR improvement and PESQ scores	55
Fig. 4-19 Segment SNR improvement and PESQ scores	56
Fig. 4-20 Segment SNR improvement and PESQ scores	57
Fig. 4-21 Segment SNR comparison and PESQ score comparison.....	58
Fig. 4-22 Segment SNR comparison and PESQ score comparison.....	59
Fig. 4-23 Segment SNR comparison and PESQ score comparison.....	60
Fig. 4-24 Segment SNR comparison and PESQ score comparison.....	61
Fig. 4-25 Segment SNR comparison and PESQ score comparison.....	62
Fig. 4-26 Segment SNR comparison and PESQ score comparison.....	63
Fig. 4-27 Segment SNR comparison and PESQ score comparison.....	64
Fig. 4-28 Segment SNR comparison and PESQ score comparison.....	65
Fig. 4-29 Segment SNR comparison and PESQ score comparison.....	66
Fig. 4-30 Segment SNR comparison and PESQ score comparison.....	67
Fig. 4-31 Segment SNR comparison and PESQ score comparison.....	68
Fig. 4-32 Segment SNR comparison and PESQ score comparison.....	69
Fig. 4-33 Segment SNR comparison and PESQ score comparison.....	70
Fig. 4-34 Segment SNR comparison and PESQ score comparison.....	71
Fig. 4-35 Segment SNR comparison and PESQ score comparison.....	72
Fig. 4-36 Segment SNR comparison and PESQ score comparison.....	73
Fig. 4-37 Average segment SNR improvement for ideal/non-ideal VAD....	85
Fig. 4-38 Average segment SNR difference for ideal/non-ideal VAD.....	85

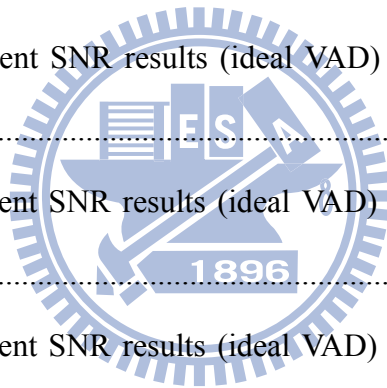
Fig. 4-39 Average PESQ improvement for different noise type	86
Fig. 4-40 Average PESQ difference for ideal/non-ideal VAD	86
Fig. 4-41 Segment SNR for different silence length before speech.....	94
Fig. 4-42 PESQ for different silence length before speech	94
Fig. 4-43 Comparison of segment SNR for ideal/non-ideal VAD with 16 sec silence	95
Fig. 4-44 Comparison of PESQ scores for ideal/non-ideal VAD with 16 sec silence	95
Fig. 5-1 Hardware architecture of the proposed design.....	101
Fig. 5-2 Hardware schedule of the proposed design.....	102
Fig. 5-3 Layout of the digital hearing aid system chip	104
Fig. 5-4 Position marking of each submodules in the chip layout.....	104



List of Tables

Table 3-1 Frequency response of each subband of the filter bank.....	10
Table 3-2 Energy and Entropy for Gaussian noise and sine wave.....	15
Table 3-3 Mitchell’s approximation example	22
Table 4-1 Segment SNR results for sequence 1 with white noise	42
Table 4-2 Segment SNR results for sequence 2 with white noise	43
Table 4-3 Segment SNR results for sequence 3 with white noise	44
Table 4-4 Segment SNR results for sequence 4 with white noise	45
Table 4-5 Segment SNR results for sequence 1 with babble noise.....	46
Table 4-6 Segment SNR results for sequence 2 with babble noise.....	47
Table 4-7 Segment SNR results for sequence 3 with babble noise.....	48
Table 4-8 Segment SNR results for sequence 4 with babble noise.....	49
Table 4-9 Segment SNR results for sequence 1 with factory noise.....	50
Table 4-10 Segment SNR results for sequence 2 with factory noise.....	51
Table 4-11 Segment SNR results for sequence 3 with factory noise	52
Table 4-12 Segment SNR results for sequence 4 with factory noise.....	53
Table 4-13 Segment SNR results for sequence 1 with car noise	54
Table 4-14 Segment SNR results for sequence 2 with car noise	55
Table 4-15 Segment SNR results for sequence 3 with car noise	56
Table 4-16 Segment SNR results for sequence 4 with car noise	57
Table 4-17 Segment SNR results (ideal VAD) for sequence 1 with white noise	58
Table 4-18 Segment SNR results (ideal VAD) for sequence 2 with white noise	59

Table 4-19 Segment SNR results (ideal VAD) for sequence 3 with white noise	60
Table 4-20 Segment SNR results (ideal VAD) for sequence 4 with white noise	61
Table 4-21 Segment SNR results (ideal VAD) for sequence 1 with babble noise	62
Table 4-22 Segment SNR results (ideal VAD) for sequence 2 with babble noise	63
Table 4-23 Segment SNR results (ideal VAD) for sequence 3 with babble noise	64
Table 4-24 Segment SNR results (ideal VAD) for sequence 4 with babble noise	65
Table 4-25 Segment SNR results (ideal VAD) for sequence 1 with factory noise	66
Table 4-26 Segment SNR results (ideal VAD) for sequence 2 with factory noise	67
Table 4-27 Segment SNR results (ideal VAD) for sequence 3 with factory noise	68
Table 4-28 Segment SNR results (ideal VAD) for sequence 4 with factory noise	69
Table 4-29 Segment SNR results (ideal VAD) for sequence 1 with car noise	70
Table 4-30 Segment SNR results (ideal VAD) for sequence 2 with car noise	71
Table 4-31 Segment SNR results (ideal VAD) for sequence 3 with car noise	



.....	72
Table 4-32 Segment SNR results (ideal VAD) for sequence 4 with car noise	73
Table 4-33 Average performance in white noise.....	77
Table 4-34 Average performance in babble noise.....	78
Table 4-35 Average performance in factory noise.....	79
Table 4-36 Average performance in car noise.....	80
Table 4-37 Average performance in white noise with ideal VAD.....	81
Table 4-38 Average performance in babble noise with ideal VAD.....	82
Table 4-39 Average performance in factory noise with ideal VAD.....	83
Table 4-40 Average performance in car noise with ideal VAD.....	84
Table 4-41 Average performance for 2 seconds silence before speech.....	89
Table 4-42 Average performance for 4 seconds silence before speech.....	90
Table 4-43 Average performance for 8 seconds silence before speech.....	91
Table 4-44 Average performance for 16 seconds silence before speech.....	92
Table 4-45 Average performance for 16 seconds silence before speech with ideal VAD.....	93
Table 4-46 Comparison of segment SNR over different algorithms.	96
Table 4-47 Average segment SNR improvement and average PESQ score improvement.....	97
Table 5-1 Hardware specifications of the proposed design.....	103
Table 5-2 Power report of the proposed noise reduction design.....	105

Chapter 1. Introduction

1.1. Background

Environment noise degrades speech quality. For hearing aids application, since the amplification is required in order to compensate the patients' hearing loss, louder noise not only causes the reduction of speech intelligibility but also results in uncomfortable experience for the patients. Thus, noise reduction is an important issue in hearing aid application.

For noise reduction in digital hearing aids systems, the key point is that the design must be real-time and follows the specification of the system. Modern hearing aids applications emphasize on mobility and long battery life, which is particularly preferred by the patients who suffered from inconvenience experience caused by some hearing aids systems. The more electric power is required, the heavier and the larger the battery will be to extend the battery changing period. Thus the noise reduction also needs to be low-power designed in order to save the system volume and the computation power dissipation.

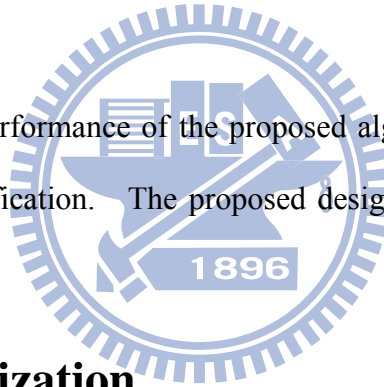
To sum up, noise reduction design has to be integrated into the digital hearing aid system chip and enhance the speech quality. Low power design is also required such that the volume and the weight of the system can be minimized, which is a challenging task.

1.2. Motivation and Contribution

The issues mentioned above motivate us to propose a low power noise reduction design for hearing aids application. The goal is to propose an algorithm which can be implemented by real-time hardware design with low power consumption.

The contribution of the thesis includes:

1. We formulated the VAD algorithm which is low power optimized and provides reference for the decision in spectral subtraction.
2. We formulated the spectral subtraction algorithm which is low power optimized, enhancing the speech quality and intelligibility under noisy environments.
3. We analyzed the performance of the proposed algorithm and implemented in hardware with verification. The proposed design is integrated into a digital hearing aid system.



1.3. Thesis Organization

In Chapter 2, we will briefly introduce different types of the noise reduction algorithms. In Chapter 3, the proposed noise reduction algorithm with entropy-based voice activity detection and filter bank-based spectral subtraction with hardware optimization techniques will be discussed. Chapter 4 will present and analyze the simulation results with different types of background noise and speech test sequence. Chapter 5 will discuss the hardware implementation of the proposed design and show the circuit area along with power reports. The conclusion will be given in Chapter 6.

Chapter 2. Related Work

2.1. Overview

Many noise reduction algorithms for hearing aid were proposed. According to [1], the noise reduction method can be categorized into 4 types: spectral-subtractive algorithms, wiener filtering, statistical-model-based methods, and subspace algorithms. Subspace algorithms such as [2] and [3] utilize the characteristics of the vector space of the noisy signal which can be decomposed into “signal” and “noise” subspaces. By keeping the components falling in the “signal” subspace and nulling the components that are in the “noise” subspace, noise can be suppressed. However, this type of algorithm needs further processing on the signal by forming the gain matrix, resulting in high computation complexity, by operations such as matrix arithmetic and matrix inversion. Wiener filtering algorithms that mentioned in [4] and implemented in [5], [6], and [7] use linear prediction methods under minimum-mean-square error criterion, establishing an optimum filter model by minimizing the speech distortion subject to the noise distortion lying under a given threshold. Nevertheless, those algorithms require multi-microphone architecture, which needs high computation power and are not suited to the proposed design. Statistical-model-based methods like [8], [9] and [10] often utilize nonlinear estimators of the magnitude of DFT coefficients with different types of statistical models and optimization criteria. Estimated clean speech signal are acquired by the information gathered with SNR estimation and noise signal variance. The drawback of the statistical-model-based methods is that they still suffer from high computation complexity and the frequency decomposition methods don't meet the requirements.

Spectral-subtractive algorithms manipulate the data in frequency domain and subtract unwanted noise spectral energy from the noisy speech energy. The unsophisticated algorithm procedures fulfill the need for low power requisition with comparatively lower computing complexity and more friendly for hardware implementation. The spectral-subtractive method is based on frequency dividing and spectral subtraction. The noisy speech signal is first converted into spectral domain and the spectral energy is then subtracted by the amount of the estimated background noise in order to restore the original clean speech. The spectral-subtractive methods satisfy the need for low computational complexity and low power hardware implementation according to its simple and inherent hardware-friendly characteristic. The spectral-subtractive methods will be introduced in the following section.

2.2. Spectral-Subtractive Method

2.2.1. Introduction

The spectral-subtractive method, first introduced by Boll[11] consists of two basic steps: spectral decomposition, i.e. frequency dividing, and de-noise process. The spectral decomposition is often implemented by methods such as fast Fourier transform (FFT), discrete cosine transform (DCT), discrete wavelet transform (DWT), or filter bank. After spectral decomposition is the de-noise process. In order to enhance the performance of the de-noise process, voice-activity detection (VAD) operation is widely adopted. VAD makes the judgment whether a specific period is voice active or not, giving decision for the action of taking noise estimation or performing the spectral subtraction. The spectral subtraction process includes spectral attenuation within silence period, and spectral subtraction, i.e. speech

enhancement for voiced period. The following section discusses the related works of VAD and spectral subtraction scheme respectively.

2.2.2. Voice Activity Detection

Noise reduction algorithm in speech processing requires high performance and needs to preserve good intelligibility and quality of the original speech. In that case, we need not only good noise reduction mechanisms but also accurate voice-activity detection algorithms in order to differentiate voice-active region from silence region, namely “noise-dominating region” in noisy environments. Various voice-activity detection methods have been proposed. Directly judging by signal energy or magnitude such as [12] suffers from lower accuracy under low SNR environments. Statistical-based or model-based VAD algorithms such as [13] and [14] achieve good performance in such condition, while need higher computation complexity and hardware resources. Autocorrelation function or teager energy operator (TEO) based VAD algorithms such as [15], [16], and [17] well balance complexity and accuracy under noisy environments, while the flexibility under different noisy environments does not meet the requirement of the hearing aid system. Entropy based method as proposed by [18] utilize the spectral energy and calculate the entropy value and have better performance compared to TEO based methods and magnitude judging methods. However, the computation complexity is higher than the TEO or judging by signal energy scheme and needs to be modified in order to meet the low-power requirement.

2.2.3. Spectral Subtraction

Many works on spectral subtraction have been proposed. From the studies on spectral subtraction, the main issues are that the presence of musical noise[19] and the

damage to the original speech. Many algorithms use hard thresholding or soft thresholding such as [20]. Over-subtraction methods such as [19] and [21] prevent the result of spectral subtraction from lying under some preset minimum threshold to conquer the musical noise problem. Nonlinear and multiband approaches such as [21-25] exploit different gain or subtraction factor for each frequency components in spectral domain, preserving the intelligibility of the speech spectral energy range and eliminating unwanted noise in other frequency components. The nonlinear and multiband approaches require frequency dividing process and the hardware complexity are comparatively higher thus need to be hardware-optimized when they are applied to the hearing aid system.

2.3. Summary

The noise reduction algorithms can be categorized into spectral-subtractive algorithms, wiener filtering, statistical-model-based methods, and subspace algorithms. For low power design, the spectral-subtractive algorithms are preferred. Most spectral subtractive algorithms take VAD and spectral subtraction as the basic steps in order to provide accurate noise estimation and better decision of different scheme for speech period and silence period. Entropy-based VAD and multiband spectral subtraction are well balanced between performance and computation complexity while they still need optimization for low-power hardware implementation.

Chapter 3. Noise Reduction Algorithm with Entropy-Based Voice Activity Detection and Filter Bank-Based Spectral Subtraction

3.1. Overview

In this chapter, we present a noise reduction algorithm with voice activity detection (VAD) and spectral subtraction. The proposed VAD algorithm utilizes the entropy of the speech signal energy with filter bank-based frequency dividing. The spectral subtraction process is done under frequency domain acquired by the analysis filter bank, which is inspired by previous spectral-subtractive methods. The VAD result controls the decision and offers the reference noise estimation of filter bank-based spectral subtraction within voiced or silence region, which further enhances the performance of the proposed algorithm. The filter bank-based spectral subtraction enhances the speech SNR by utilizing the information of noise magnitude estimated during voiced period and suppresses the noise signal during non-voiced period. The subband signals are then synthesized through synthesis filter bank after other processing block in the hearing aid system.

This chapter is organized as follows. First the hearing aid system and the filter bank will be briefly introduced. The proposed algorithm flow will be illustrated in the next section. Then each part of the algorithm will be discussed in detail in the rest of the chapter.

3.2. Introduction to the Hearing Aid System

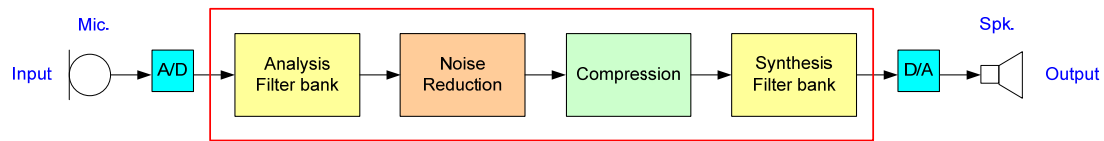


Fig. 3-1 Functional block diagram of the hearing aid system

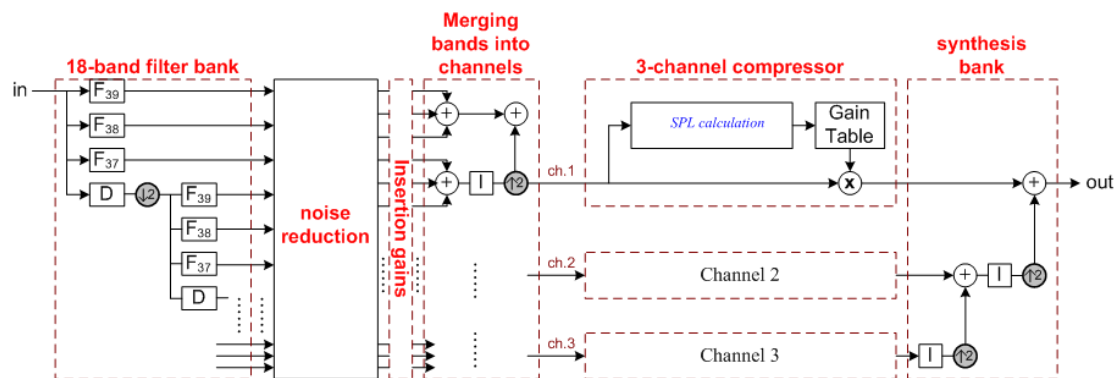


Fig. 3-2 Block diagram of the digital system

Fig. 3-1 shows the functional block diagram of the overall hearing aid system. The external sound is inputted to the system by microphone. The analog to digital converter then converts the analog signal to digital data samples with 24-kHz sampling rate. Then the digital system (red rectangle part) starts the digital signal processing.


Fig. 3-2 illustrates the block diagram of the digital system. After filtering by the “18-Band Analysis Filter bank” block, the samples are frequency-divided into 18 subbands. The “Noise Reduction” block reduces the unwanted background noise, enhancing the quality of speech. Then the “Insertion Gain” block applies different gains on each subband of the output of the noise reduction block in order to compensate the hearing loss of the patient. “Merging Bands into Channels” block,

also known as “B2C” block, merges 18 subbands into 3 channels for the following compressor. “Compressor” block compresses the dynamic range of the magnitude of the samples, preventing from over-amplifying the magnitude of the data samples, which may cause damage to the patient’s ear. Then the synthesis filter bank reconstructs the data samples from subband samples.

Finally, the digital to analog converter converts the digital data sample to analog signal, and the sound is generated by the speaker.

3.3.ANSI S1.11 Filter Bank for Digital Hearing

Aids[26]



In hearing aid systems, most of the parts require frequency dividing in order to apply different gains or compensations to specific bands. In the proposed noise reduction algorithm, filter-bank based spectral subtraction is utilized. In the filter-bank design, human hearing characteristics are well simulated since the frequency dividing mechanism is based on the ANSI S1.11[27] standard. Most of the filters that base on straightforward FIR design encounter the problem of high complexity and require large amount of hardware resource. The filter bank that adopted in the proposed noise reduction algorithm provides an energy-efficient solution, implementing the 22nd to the 39th 1/3-octave bands in the ANSI S1.11 standard.

The multi-rate algorithm is composed of three filters and a low-pass decimation filter. The inputs samples are recursively obtained for smaller octaves by band-limiting and downsampling the input signals.

The number of output data samples from each analysis filter bank octave descends by a factor of 1/2 from each input frame length, which is 32, resulting in 32, 32, 32, 16, 16, 16, 8, 8, 8, 4, 4, 4, 2, 2, 2, 1, 1, 1 samples for band F39 to F22 respectively. The bitwidth of the data sample is 16-bit, with 24kHz sampling rate. Table 3-1 shows the specifications of frequency response for each subband of the filter bank.

band	Center frequency. (Hz)	Upper bound (Hz)	Lower bound (Hz)
F39	8000	10365	6174
F38	6300	8163	4862
F37	5000	6478	3859
F36	4000	5183	3087
F35	3150	4081	2431
F34	2500	3239	1930
F33	2000	2591	1544
F32	1600	2073	1235
F31	1250	1620	965
F30	1000	1296	772
F29	800	1037	617
F28	630	816	486
F27	500	648	386
F26	400	518	309
F25	315	408	243
F24	250	324	193
F23	200	259	154
F22	160	207	123

Table 3-1 Frequency response of each subband of the filter bank.

3.4. Introduction to the Proposed Noise Reduction

Algorithm

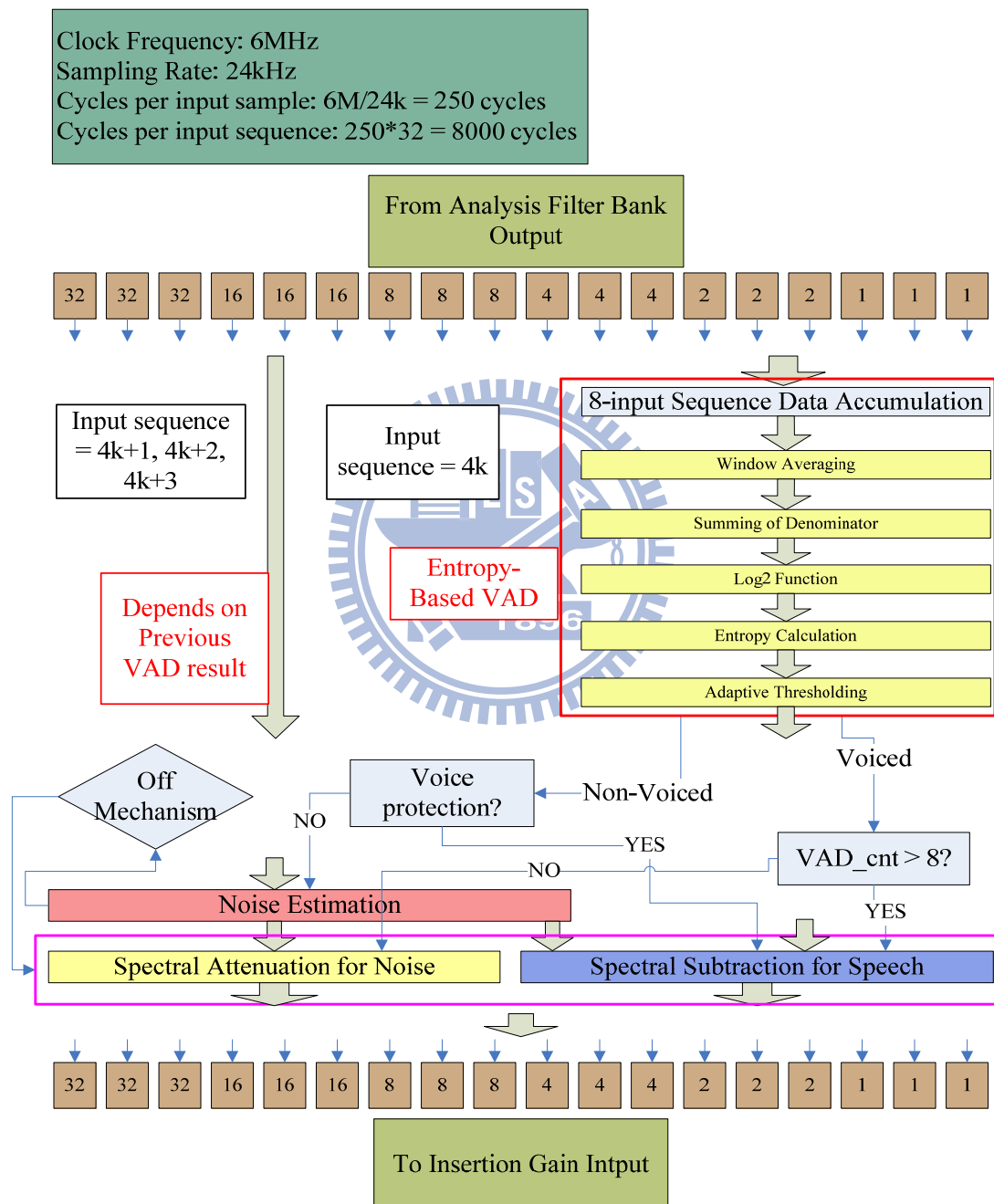


Fig. 3-3 Flow of the proposed algorithm

Fig. 3-3 illustrates the flow of the noise reduction algorithm with entropy-based voice activity detection and filter-bank based spectral subtraction. Initially the digital noise-corrupted speech signal is acquired through D/A converter with 24kHz sampling rate. Then the data passes the ANSI S1.11 analysis filter bank, resulting in 18 subband signals. The length of each filter-bank input sequence is 32, i.e., 32 samples of speech data will be fed into the filter bank, generating different length of output data sequence in each subband respectively as shown in the upper part of the figure. If the input sequence number is the multiple of 4, for instance, 4k, the entropy-based VAD operation will be activated. Otherwise the data will be directly processed depending on previous VAD result. The entropy-based VAD calculates the entropy of the input signal from the data in each subband. The magnitude of entropy indicates the possibility whether the corresponding region is voice-active or not. In order to improve the distinguishing ability of the entropy in different types of speech or different noisy environments, the adaptive thresholding technique is proposed. With the utilization of adaptive thresholding technique, VAD result will be generated and be applied to the judgment of voiced or silence region signal processing.

Different signal processing method will be performed depending on the type of the signal (voiced or silence). If a signal sequence is judged as voice-active and the VAD_cnt exceeds 8, namely, the VAD has made the judgment of voice-active for more than 8 times consecutively, the signal will be processed through “Spectral subtraction for Speech” block, or it will be processed through “Spectral Attenuation for Noise” block instead. If the signal is judged as silence but lies in the “Voice protection zone”, the signal will also be processed through “Spectral Subtraction for Speech” block, or the signal will be processed through “Spectral Attenuation for

Noise” block. “Noise Estimation” block estimates the magnitude of the environment noise, which is then become the reference noise data in “Spectral Subtraction for Speech”. “Off Mechanism” disables the calculation if the estimated noise is less than a fixed threshold. Finally the processed data is fed to “Insertion Gain” block which is previously described.

3.5. Entropy-Based Voice Activity Detection

3.5.1. Entropy in Speech Processing

Entropy[28], in information theory, stands for the amount of uncertainty measured with some set of specific variable, for example, X . It is usually denoted by $H(X)$, where

$$H(X) = -\sum_{x_i \in X} p(x_i) \log p(x_i) \quad (3-1)$$

Where $p(x)$ stands for the probability density function of x .

As we can see, the entropy equation indicates that the entropy function maximizes if all the variables x_i in set X are equiprobable, i.e. $p(x_i) = 1/n$, where n = number of x_i 's in set X . In other words, we could state that X is most “unpredictable” under such condition. Fig. 3-4 is a sinusoidal wave with period T . Fig. 3-5 is a Gaussian noise generated by MATLAB function *awgn*. The magnitudes of the samples of the two signals are normalized to 1 respectively. The part $p(x_i)$ is the magnitude acquired by taking 256-point Fourier transform and normalized over the energy of all the transform coefficients, namely:

$$p(x_i) = \frac{\text{abs}(FFT(x_i))}{\sum_1^{256} \text{abs}(FFT(x_i))} \quad , \quad i = 1 \dots 256 \quad (3-2)$$

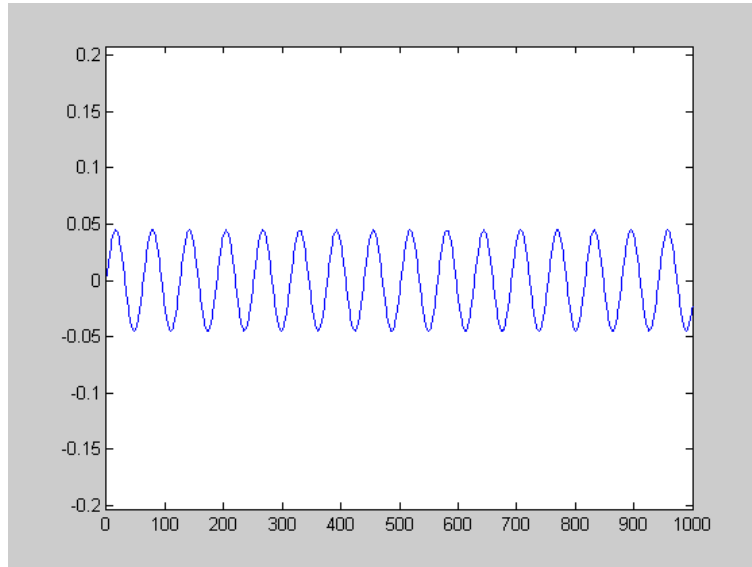


Fig. 3-4 Sine wave with period T

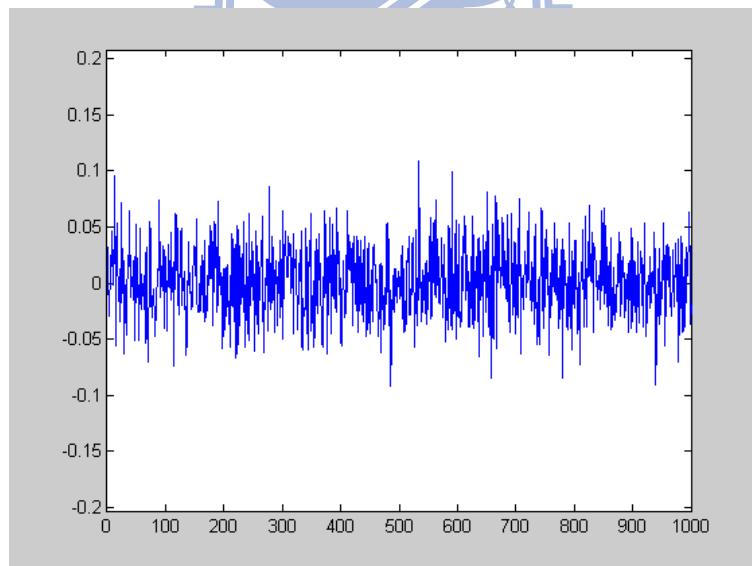


Fig. 3-5 Gaussian noise

Table 3-2 lists the energy and entropy calculated by (3-1) of the two signals. We can find that although the two signals have almost same energy, the entropy differs a lot since the dominating frequency components in sine wave and Gaussian noise are not the same. The dominant frequency component in sine wave is the inherent frequency of itself while there's no dominating frequency component in Gaussian noise according to its definition. Thus we can learn that for signals with some dominating frequency component, for example, speech signals, the entropy will behave dissimilarly from that of noise signals, which usually have no specific dominant frequency components. This condition is closer to “equiprobable” that mentioned earlier in this chapter.

	Energy	Entropy (H)
Gaussian noise	0.249	2.36
sine wave	0.251	1.36

Table 3-2 Energy and Entropy for Gaussian noise and sine wave

3.5.2. Entropy Calculation

According to the definition of entropy and the specification of the filter bank, the entropy is calculated by the following equation:

$$H(Y) = -\sum_{y_i \in Y} p(y_i) \log p(y_i) \quad (3-3)$$

Where

$$p(y_i) = \frac{abs(y_i)}{\sum_{F22}^{F39} abs(y_i)} \quad , \quad i = F22 \sim F39 \quad (3-4)$$

Where y_i is the average spectral energy of the filter bank output in subband i which is acquired by taking the absolute value for simplicity.

For entropy calculated in (3.3), the noise-dominant region and the speech-dominant region can be differentiated[18]. In order to make the entropy difference more recognizable and let the VAD be more robust, a constant K is introduced[29]. Thus we get the modified entropy equation:

$$H'(Y) = -\sum_{y_i \in Y} p'(y_i) \log p'(y_i) \quad (3-5)$$

Where

$$p'(y_i) = \frac{abs(y_i)+K}{\sum_{F22}^{F39}(abs(y_i)+K)}, \quad i = F22 \sim F39 \quad (3-6)$$

Now, we let

$$\Delta p(y_i) = p'(y_i) - p(y_i) = \frac{1}{\frac{\sum_{F22}^{F39} abs(y_i)}{K} + N} \cdot (1 - N) \cdot p(y_i) \quad (3-7)$$

Where $N = 39-22+1 = 18$

By observing (3-7), we can analyze the influence of K . If $p(y_i) > 1/N$, $\Delta p(y_i) < 0$, which means $p'(y_i) < p(y_i)$. In contrast, if $p(y_i) < 1/N$, $\Delta p(y_i) > 0$, which means $p'(y_i) > p(y_i)$. The introduction of K makes $p'(y_i)$ s tend to be equal in one frame, thus the entropy of each frame increases. The key point that makes the difference of $H'(Y)$ between noise-dominant frames and speech-dominant frames become larger is that the energy of speech-dominant frames (speech + noise) are commonly greater than that of noise-dominant ones (noise only), which results in larger advance in entropy for noise-dominant frames than that of speech dominant frames. Consequently, it is easier to do thresholding for the entropy, resulting in

better VAD accuracy.

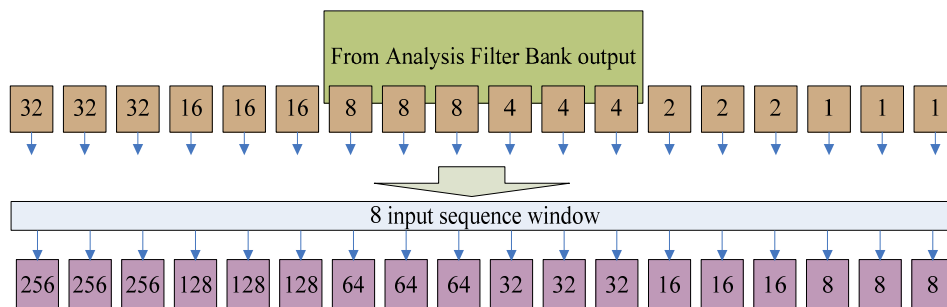


Fig. 3-6 Scheme for input sequence window gathering

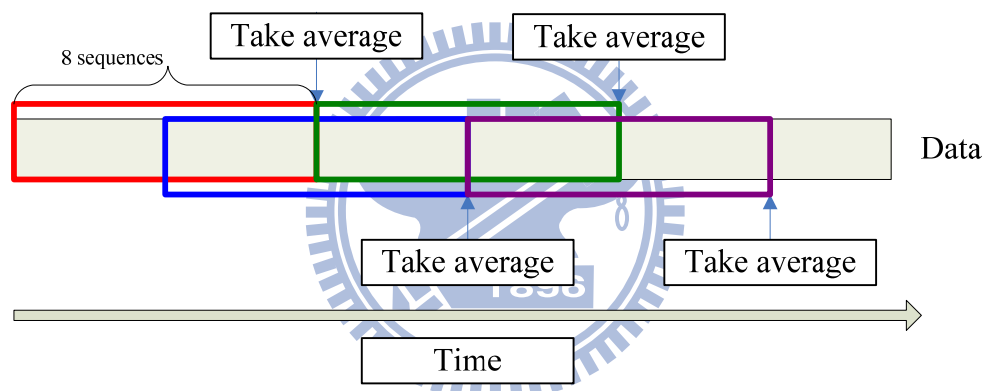


Fig. 3-7 Scheme for window averaging

The calculation of entropy is performed by the following step:

- The output of the ANSI S1.11 analysis filter bank is first gathered for 8 sequences, i.e. 256 samples for band F39~F37, as shown in Fig. 3-6.
- Then the window is averaged every 4 sequences, i.e. every 128 samples for band F39~F37, as shown in Fig. 3-7.
- The averaged data sample in each subband is set to be y_i . After taking absolute value of y_i , we sum up $abs(y_i)$'s from band F22 to band F39 and add K to obtain the denominator of (3-6).

- d. After calculating $p'(y_i)$ by (3-6) for each subband, the log value of $p'(y_i)$ is then acquired with log base 10.
- e. Then the entropy value $H'(Y)$ is derived by (3-5). By (3-8), the VAD result is determined.

$$VAD = \begin{cases} 1, & \text{if } (-H'(Y)) > \text{threshold} \\ 0, & \text{if } (-H'(Y)) \leq \text{threshold} \end{cases} \quad (3-8)$$

The entropy calculation steps are listed above. However, some of them are not suitable for low-power hardware implementation. The detailed optimization process for low-power hardware design will be discussed in the following section.

3.5.3. Low-Power Hardware Optimizations for Entropy

Calculation

Our goal for the noise reduction algorithm is not only suitable for digital hearing aid system but also operates with low-power dissipation. For hardware optimization, the algorithm flow is modified for each step as follows:

- a. The input sequence costs lots of storage space and may leads to high power consumption and large die-area for 3.5.2a. Also, the real-time constraint limits the data to be read and stored simultaneously when the calculation is in progress. If the data samples are directly written into the register, high area and gate count will be required, resulting in large power consumption.

To solve the problem, we have the input data samples from analysis filter bank be written into 2 SRAM sets alternately for further data processing. SRAM

has smaller area occupation than the register and only needs extra 2 cycles for data read/write. For real-time processing, 2 SRAM sets are used as a ping-pong buffer such that current input sequences can be processed when the next input sequence is being gathered in the same time. In that way, the real-time processing with small hardware usage can be carried out.

- b. For window averaging in 3.5.2b, the data samples of each subband are summed and written into the register for every input sequences and are denoted as $r00, r01, \dots, r17$. Then $r00, r01, \dots, r17$ in each sequence are summed together respectively for every 4 input sequences.

The registers used for summing $r00, r01, \dots, r17$ in sequence $8k \dots 8k+3$ are called $r01_Reg0to3, r02_Reg0to3, \dots, r17_Reg0to3$. And the registers used for summing $r00, r01, \dots, r17$ in sequence $8k+4$ to $8k+7$ are called $r01_Reg4to7, r02_Reg4to7, \dots, r17_Reg4to7$.

We can use two sets of registers to accumulate and take average of the sequence that is more efficient and saves the circuit area. The scheme is illustrated in Fig. 3-8.

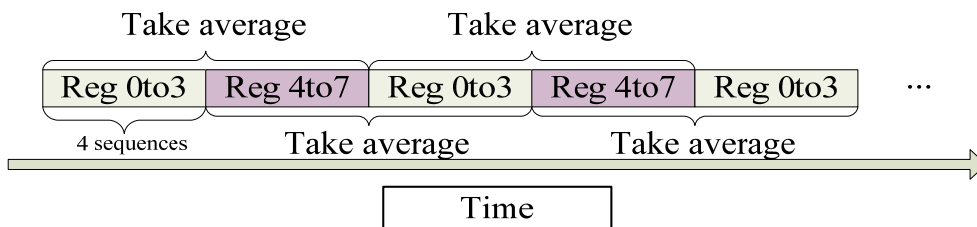


Fig. 3-8 Scheme for window averaging by register refreshing

- c. The constant K in (3-6) is set as $(\frac{1}{2})^5$, which is 0x0200 corresponding to 16-bit data format of input sequence.
- d. The log function of the algorithm in 3.5.2d has base 10, which is hard to implement in hardware design.

To solve the problem, we do the modification as follows:

- i. Change the base to 2 by base-changing properties:

$$\log_{10} p'(y_i) = \log_2 p'(y_i) \times \log_{10} 2 \quad (3-9)$$

- ii. By Mitchell's algorithm[30], we can approximate the log values with base 2 by interpolation method. If we want to take binary logarithms of a number N (i.e. $\log_2(N)$), first we can express N as:

$$N = 2^k(1 + m) \quad (3-10)$$

Then we take the logarithms on both sides:

$$\log_2 N = k + \log_2(1 + m) \quad (3-11)$$

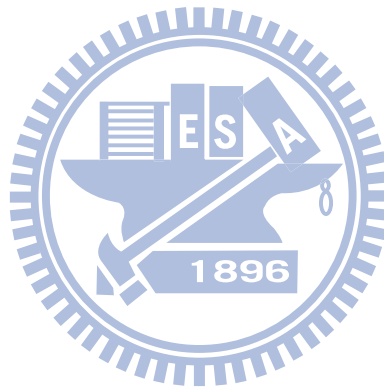
The approximation based on Mitchell's algorithm is:

$$(\log_2 N)' = k + m \quad (3-12)$$

So the approximation error will be:

$$\log_2(1 + m) - m \quad (3-13)$$

The example table of the above scheme is listed in Table 3-3. The example has 3 integer bits and 5 fractional bits while for proposed hardware design, the log function has 6 integer bits and 10 fractional bits



N Decimal	N Binary	$\log_2 N$ Exact	$\log_2 N$ Approx.	$\log_2 N$ Binary Approx.
$1=2^0$	00001	0.00000	0.00000	000.00000
$2=2^1$	00010	1.00000	1.00000	001.00000
3	00011	2.58496	1.50000	001.10000
$4=2^2$	00100	2.00000	2.00000	010.00000
5	00101	2.32193	2.25000	010.01000
6	00110	2.58496	2.50000	010.10000
7	00111	2.80735	2.75000	010.11000
$8=2^3$	01000	3.00000	3.00000	011.00000
9	01001	3.16993	3.12500	011.00100
10	01010	3.32193	3.25000	011.01000
11	01011	3.45943	3.37500	011.01100
12	01100	3.58496	3.50000	011.10000
13	01101	3.70044	3.65200	011.10100
14	01110	3.80735	3.75000	011.11000
15	01111	3.90689	3.87500	011.11100
$16=2^4$	10000	4.00000	4.00000	100.00000
17	10001	4.08746	4.06250	100.00010
18	10010	4.16993	4.12500	100.00100
19	10011	4.24793	4.18750	100.00110

Table 3-3 Mitchell's approximation example

- e. The equation (3-6) in 3.5.2 has division, which is not preferred in hardware design and requires additional hardware resource.

To solve the problem, we let $p'(y_i) = \frac{num_p(y_i)}{den_p(y_i)}$ first. Then the division operation of (3-6) is reduced by rewriting (3-5) as:

$$\begin{aligned}
 den_p(y_i) \times H'(Y) = & \\
 - \sum_{y_i \in Y} (num_p(y_i)) \times [\log(num_p(y_i)) - \log(den_p(y_i))] & \\
 & \qquad \qquad \qquad (3-14)
 \end{aligned}$$

Thus (3-8) must also be rewritten as:

$$VAD = \begin{cases} 1, & \text{if } den_p(y_i) \times (-H'(Y)) > den_p(y_i) \times threshold \\ 0, & \text{if } den_p(y_i) \times (-H'(Y)) \leq den_p(y_i) \times threshold \end{cases} \quad (3-15)$$

After removing the division, the remaining multiplication is then replaced by shift-add techniques and will be discussed in 5.1. Thus no hardware multipliers are needed, resulting in an one-adder arithmetic unit architecture in the circuit design.

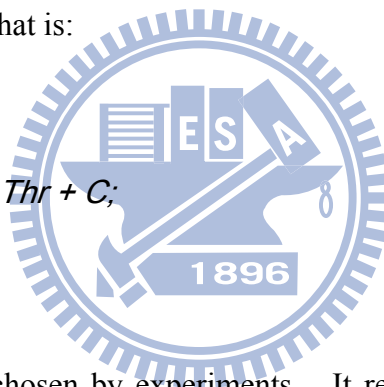
By the modification mentioned above, complexity of the algorithm is reduced effectively for hardware implementation with simpler calculations.

3.5.4. Adaptive Thresholding

The entropy value may be the reference for determining whether the frame tends to be speech-dominant or not. To present a definite VAD result, a threshold is required. A fixed threshold may be simple but offers poor accuracy when different type or SNR of environment noise is dominant. Thus the adaptive thresholding technique may be a good solution to the condition above. The adaptive thresholding process will be shown below.

First we assign current $threshold = Thr$ as the summation of a constant C and the parameter $Adaptive_Thr$, that is:

$$Thr = Adaptive_Thr + C;$$



Where C is a constant chosen by experiments. It represents the initial value of Thr .

Then we calculate Ent_static_cnt with the following:

```
If (abs(prev_Thr - Thr) < Q && abs(prev_Ent - Ent) < R)
```

```
Ent_static_cnt = Ent_static_cnt + 1;
```

```
else
```

```
Ent_static_cnt = 0;
```

Where $prev_Thr$ is previous *threshold*, Ent is $-H'(Y)$, $prev_Ent$ is previous $-H'(Y)$, Q and R are 0x002dc6c0, which is chosen by experiment. Then we assign

$$Ent_margin = abs(Ent - Thr)$$

Where Ent_margin is the distance between current *threshold* and $-H'(Y)$.

Finally we do *threshold* updating if $-H'(Y)$ and *threshold* has kept static - that is, varying within some pre-defined range - over the period that needed for invoking VAD calculation 24 times.

```

If (Ent_static_cnt > 24){
    if ((Thr - Ent) > S){
        Adaptive_Thr = Adaptive_Thr - L;

        Ent_static_cnt = 0;
    }

    Else {

        Adaptive_Thr = Adaptive_Thr + L;

        Ent_static_cnt = 0;
    }
}

```

Where S is $0x0200 \times den_p(y_i)$ due to the relationship mentioned in (3-15). L is the adaptive step for *Adaptive_Thr* and is $0x0100$.

To briefly sum up, the concept of this algorithm is that if *threshold* keeps far above $-H'(Y)$ statically for a fixed period, we decrease it to make the voiced period judgment more accurate. If *threshold* lies below $-H'(Y)$ statically for a fixed period, we increase it to prevent from misjudging the silence period as voiced ones. Finally the VAD result is acquired. If $-H'(Y)$ exceeds *threshold*, the VAD flag is set to 1, and vice versa.

Fig. 3-9 gives an example of adaptive thresholding. The upmost subgraph is clean speech signal plot with VAD result in red rectangle. The center subgraph is the noisy speech. The lowest subgraph is the entropy plot with adaptive threshold drawn in red line. As we can see, the threshold approaches the entropy when the entropy is at some steady level for a period, and lies slightly above the entropy to make sure the abrupt change in entropy may result in the judgment of voiced frame accurately.

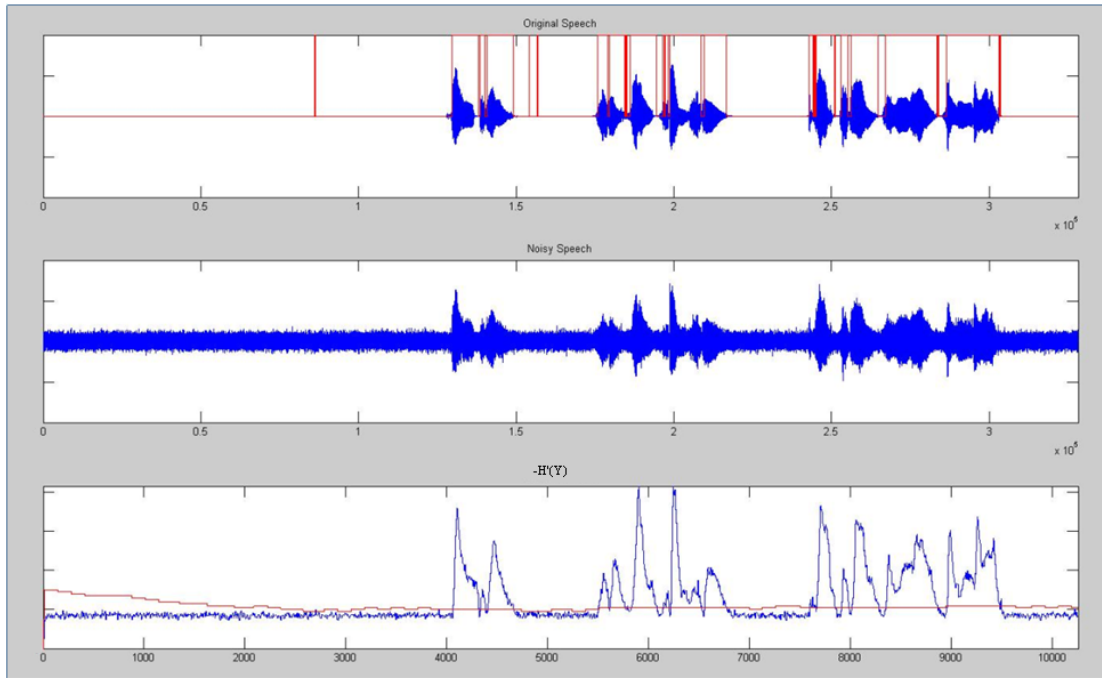


Fig. 3-9 Example of adaptive thresholding

3.6. Filter Bank-Based Spectral Subtraction

3.6.1. Introduction

After VAD process, we get the information of current frame based on which the decision of different noise reduction steps could be taken. There are four different states for noise reduction, which are *Voiced-Zone*, *Too-Short Voiced-Zone*, *Voice-Protection-Zone*, and *Silence-Zone*. The flow to decide which state to enter is shown in Fig. 3-10 and is described below,

- a. If a frame is judged as $VAD = 1$ and the previous 8 VAD results are also 1, that is $VAD_cnt > 8$, then the state will be *Voiced-Zone*. $VAD_cnt = VAD_cnt + 1$.
- b. If a frame is judged as $VAD = 1$ but is not the case of a, that is, $VAD_cnt \leq 8$ then the state will be *Too-Short Voiced-Zone*. $VAD_cnt = VAD_cnt + 1$.
- c. If a frame is judged as $VAD = 0$, but $VAD_cnt > 3$, which means it's just right

after some Voiced-Zone, then the state will be *Voice-Protection-Zone*. $VAD_cnt = VAD_cnt - 3$.

- d. If a frame is none of any of the above conditions, the state will be *Silence-Zone*. “Noise Estimation” will be performed.

The data sample will be processed under one of the states mentioned above. For case a and c, “Spectral Subtraction for Speech” will be performed. For case b and d, “Spectral Attenuation for Noise” will be performed. Finally, the output data sample will be generated and feeds to the Insertion Gain block.

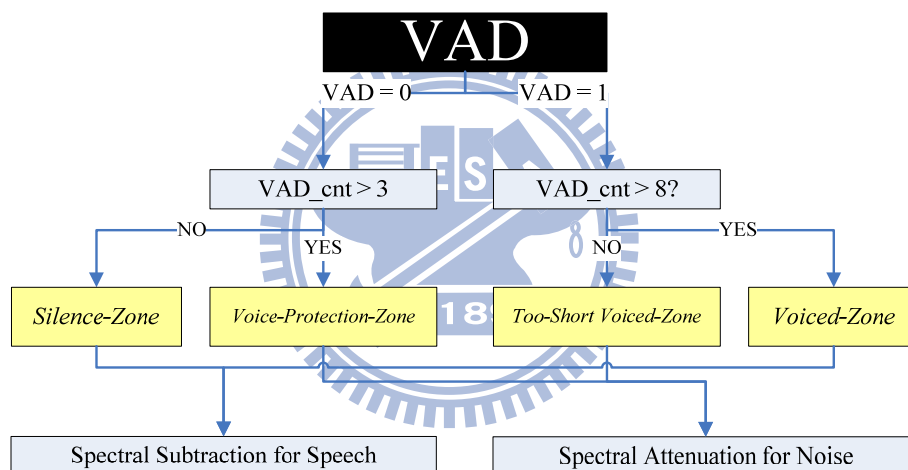


Fig. 3-10 State decision after VAD

3.6.2. Noise Estimation

If *Silence-Zone* (case d) state is entered, the “Noise Estimation” will be performed. Each time when “Noise Estimation” is activated, the data sample in each subband will be averaged respectively. Also, there will be an increment in a counter. When the counter reaches 128, that is, the noise estimation has been performed for 128 times, the noise estimation result value is updated by averaging over the past 128 noise data

that were averaged in each subband. Then the counter will be reset to zero. The process iterates again and again to ensure the noise estimation result value is up-to-date in order to provide appropriate information for spectral subtraction process. The result of noise estimation will be the reference noise signal d_i in (3-19), (3-20) and (3-21).

3.6.3. Spectral Attenuation for Noise

For case d, after noise estimation, the data samples that enter *Noised Zone* state will be attenuated. For case b, the data samples that enter *Too-Short Voiced-Zone* will also be attenuated because they are tend to be noise since their VAD period is too short.

The mechanism of *Silence-Zone* is simply attenuating each data sample in every subband by multiplying 0.125 to them, i.e.:

$$y'_{i,k} = y_{i,k} * 0.125 \quad (3-16)$$

Where $y'_{i,k}$ is the k th processed subband data sample in band i , $i = F22 \sim F39$. $y_{i,k}$ is the k th input subband data sample in band i , $i = F22 \sim F39$. Note that the range of k is dependent on which subband it is.

And the mechanism of *Too-Short Voiced-Zone* is also simply attenuating each data sample in every subband by multiplying 0.25 to them, i.e.:

$$y'_{i,k} = y_{i,k} * 0.25 \quad (3-17)$$

Then the output data samples are acquired.

3.6.4. Spectral Subtraction for Speech

From 3.6.1, if *Voiced-Zone* (case a) state or *Voice-Protection-Zone* (case c) state is entered, the spectral subtraction for speech will be performed in order to eliminate unwanted noise from speech signal. To do the spectral subtraction, we first express the speech signal in time domain that is corrupted by noise as:

$$y(n) = x(n) + d(n) \quad (3-18)$$

Where $y(n)$ is the noisy speech data sample, $x(n)$ is the original speech data sample. $d(n)$ is the noise data sample. After filtering by filter bank, the relationship may be expressed by the following equation:

$$|y_{i,k}|^2 \approx |x_{i,k}|^2 + |d_i|^2 \quad (3-19)$$

Where $y_{i,k}$ is the k th subband noisy speech data sample in band i , $i = F22 \sim F39$. $x_{i,k}$ is the k th subband original speech data sample in band i . d_i is the subband noise data sample in band i which is estimated in 3.6.2. Note that the range of k is dependent on which subband it is.

To simplify the calculation, we can estimate the clean speech by the following equation:

$$|y_{i,k}| \approx |x_{i,k}| + |d_i| \quad (3-20)$$

Now we want to estimate the speech data sample by the following equation:

$$|\hat{x}_{i,k}| = |y_{i,k}| - \mu_i |d_i| \quad (3-21)$$

μ_i is a constant and varies in different subband, and is defined as:

$$\mu_i = \begin{cases} 2.5, & \text{if } i = F22 \sim F30 \\ 1, & \text{if } i = F31 \sim F39 \end{cases} \quad (3-22)$$

The reason μ_i is different over subbands is that for band frequency lower than $F30$, the energy is closer to human voice, while for frequency higher than $F30$, it may be unwanted noise during speech. Thus we apply a higher subtraction factor to those subbands.

Next, from [25], to avoid negative values resulting from (3-21), $|\hat{x}_{i,k}|$ is floored as follows:

$$|\hat{x}_{i,k}| = \begin{cases} |\hat{x}_{i,k}|, & \text{if } |\hat{x}_{i,k}| > \beta |y_{i,k}| \\ \beta |y_{i,k}|, & \text{else} \end{cases} \quad (3-23)$$

Where β is set to 0.05.

Finally, to mask musical noise, a small amount of noisy data samples are added to the processed data samples, i.e.:

$$|\bar{\bar{x}}_{i,k}| = |\hat{x}_{i,k}| + \gamma|y_{i,k}| \quad (3-24)$$

So the processed data sample in speech period is:

$$y'_{i,k} = \text{sign}(y_{i,k}) \times |\bar{\bar{x}}_{i,k}| \quad (3-25)$$

Where the multiplication of $\text{sign}(y_{i,k})$ ensures $y'_{i,k}$ to have the same sign number as $y_{i,k}$.

From the above equation, the output of spectral subtraction is also acquired.

3.6.5. Low-Power Hardware Optimizations for Filter Bank-Based Spectral Subtraction

The spectral attenuation and spectral subtraction in the proposed algorithm requires constant number multiplication. In order to reduce the complexity and circuit area, the multiplication is approximated by linear combining the multiplicand by the factor of 1/2. The approximation is applied in (3-21), (3-23), and (3-24). For example, $y = x \times 0.1$ can be approximated by the following step:

$$\begin{aligned} y &= x \times 0.1 \\ &\cong x \times 0.09375 \\ &= (x \times 0.125) - (x \times 0.03125) \\ &= (x \gg 3) - (x \gg 5) \end{aligned} \quad (3-26)$$

As a result, the hardware architecture can be simplified.

3.6.6. Off Mechanism

For high SNR conditions, noise reduction process may not be necessary but will cause additional quality loss to the processed speech. Thus a simple off mechanism is applied. When the magnitude of noise estimation in each subband is lower than 0x0001 (in 16-bit format), the off counter will be added by 1. If the off counter exceeds 0x000f, the noise suppression and the spectral subtraction block will not be entered. It also benefits the reduction of calculation, decreasing the total power consumption.

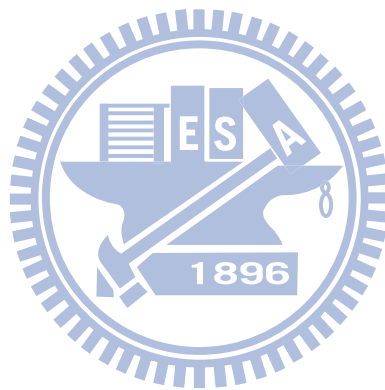
3.6.7. Data Output

Finally, the output of the proposed noise reduction algorithm is then sent to “Insertion Gain” block with the same 16-bit bitwidth format of the input data samples. The output data samples align in the manner that is same as the input data samples to the proposed noise reduction block fed by the analysis filter bank.

3.7. Summary

In this chapter, the proposed noise reduction is introduced. The entropy-based VAD step performs on every 4 input sequences by calculating the entropy of the input sequence. The entropy calculation is hardware-optimized by log base changing and linear interpolation. The filter bank-based spectral subtraction is performed based on frequency dividing by filter bank with spectral attenuation for noise during silence period and spectral subtraction in speech period. The spectral subtraction process is

also hardware optimized in order to achieve low-power consumption. In addition, off mechanism is applied to reduce computation power. The simulation and analysis will be discussed in the next chapter.



Chapter 4. Simulation and Analysis

In this chapter, the simulation settings will be described, and the results will be shown with the analysis. The simulation includes segment SNR tests and PESQ tests. Segment SNR is a measure based on Signal-to-Noise Ratio with the knowledge of the position of voice and silence regions and is defined as follows:

$$SNR_{seg} = 10 \times \log_{10}\left(\frac{Energy_{signal}}{Energy_{noise}}\right) \quad (4-1)$$

Where $Energy_{signal}$ is the energy of clean speech signal during voiced period, and $Energy_{noise}$ is the energy of the difference between clean speech and noisy/processed speech during voiced period.

Notice that the energies are calculated in speech period so that the noise in silence region is ignored. The reason to do so is that the performance of the noise reduction algorithm mostly depends on the performance during voiced region, thus we only do the calculation within pre-defined voiced region.

PESQ[31] is a objective measure that was originally based on ITU-T standard which evaluates the quality of speech. The PESQ score gives the information on the quality difference between the degraded/processed and the original speech, which varies from 0.5 to 4.5. The PESQ score is not influenced by loudness loss, sidetone, or talker echo but only reflects the one-way speech distortion perceived by the end user.

This chapter is organized as follows: section 4.1 will describe the simulation environment settings and the information of the database. Section 4.2 will be the

simulation results and the analysis.

4.1. Simulation Settings

The speech database has 150 Mandarin Chinese 2-words terms and is from Yang Ming University. The sampling frequency of the test sequences is 24kHz. The 4 test sequences are formed by concatenating 27 terms and insert silence period between each term as follows:

Seq.1: a-b-c-d-e-f

Seq.2: h-i-j-k-l-m-n

Seq.3: o-p-q-r-s-t-u

Seq.4: v-w-x-y-z-aa-bb

Where a, b, c ... bb represent the terms. The terms are:

a: ren-jian (人間)

b: shi-yong (使用)

c: shang-sin (傷心)

d: kou-ciang (口腔)

e: ming-yun (命運)

f: wun-lu (問路)

h: di-ming (地名)

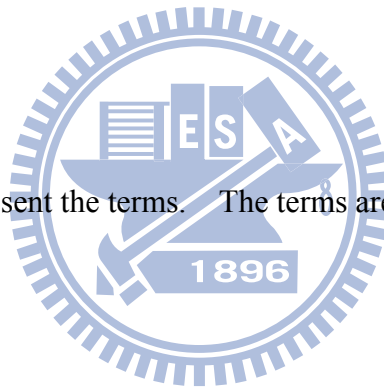
i: zeng-jin (增進)

j: da-ge (大哥)

k: nyu-hai (女孩)

l: si-si (嬉戲)

m: siao-cao (小草)



n: gong-yeh (工業)
o: nian-gao (年糕)
p: cih-hsang (慈祥)
q: cheng-jhang (成長)
r: jhan-chang (戰場)
s: tan-bing (探病)
t: lan-jhu (攔住)
u: jiao-ban (攪拌)
v: fu-yao (敷藥)
w: jhih-hui (智慧)
x: rong-shu (榕樹)
y: yue-chi (樂器)
z: ji-che (機車)
aa: lie-huo (烈火)
bb: jia-chong(甲蟲)



The first simulation is VAD accuracy test which is performed by comparing the proposed Entropy-Based VAD result with the pre-defined speech region. The second simulation is the segment SNR calculation by (4-1) along with the PESQ score comparison. The third simulation is the same as the second one but with ideal VAD (i.e. pre-defined speech region) thus the influence of the proposed VAD may be excluded, giving standalone performance report on proposed filter bank-based spectral subtraction design. The last simulation is for testing the influence of different silence length before the speech. Silence periods with 4 different lengths are added before the speech sequence 1. They are 16,8,4,2 seconds respectively.

The simulation result will also show the segment SNR and PESQ scores separately from the simulation 2 and 3.

The 4 noise sound files for the above tests are from Noisex-92 database, which are white, babble, factory, and car noise. The noisy test sequence is constructed by adding the noise sound to the clean sequence directly with the same 24kHz sampling frequency.

4.2. Experimental Result and Analysis

Fig. 4-1 to Fig. 4-4 show the graph of accuracy of proposed entropy-based voice activity detection. The x-axis is the original segment SNR and the y-axis is the accuracy in %. It can be observed that VAD accuracy has close relationship to the original segment SNR but saturates at about 90%. For original segment SNR over 6dB, the accuracy reaches 80%. Different types of noise also affect the VAD result. For white noise under low segment SNR condition, the VAD accuracy falls below 70% since the entropy level pulsates a lot so that the adaptive threshold doesn't perform well. For babble and factory noise, the VAD accuracy meets the average performance. Finally for car noise, the VAD accuracy performs outstanding since the main energy of car noise concentrates in very low frequency and doesn't affect other subbands in the filter bank, causing the speech energy component highly distinguishable.

VAD accuracy result: white noise (%)

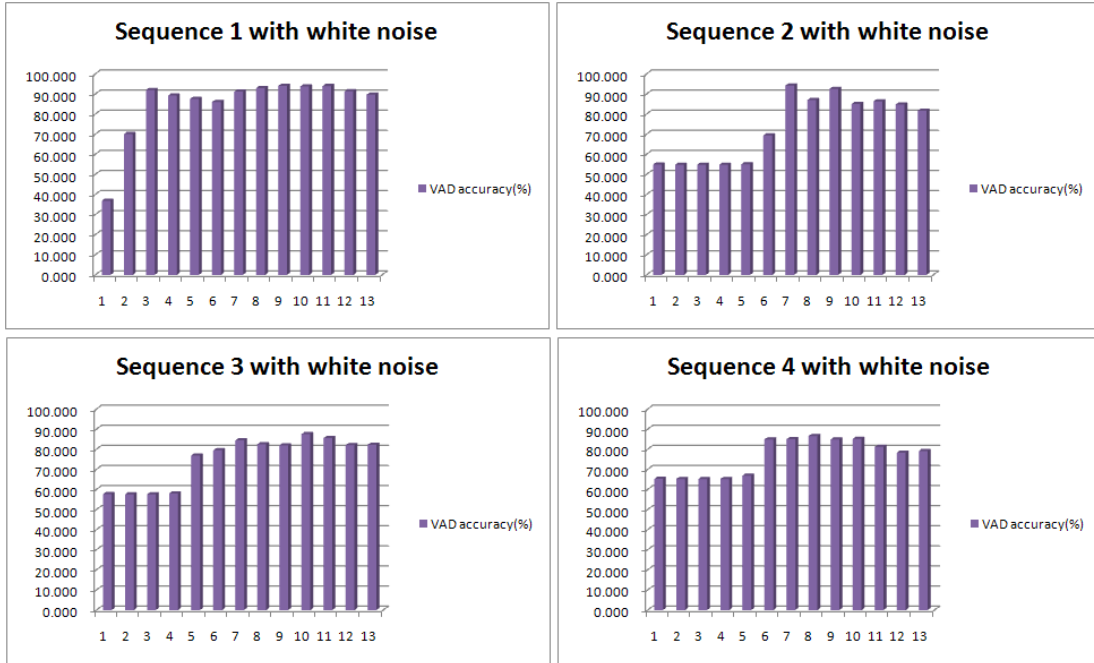


Fig. 4-1 VAD result for white noise

VAD accuracy result: babble noise (%)

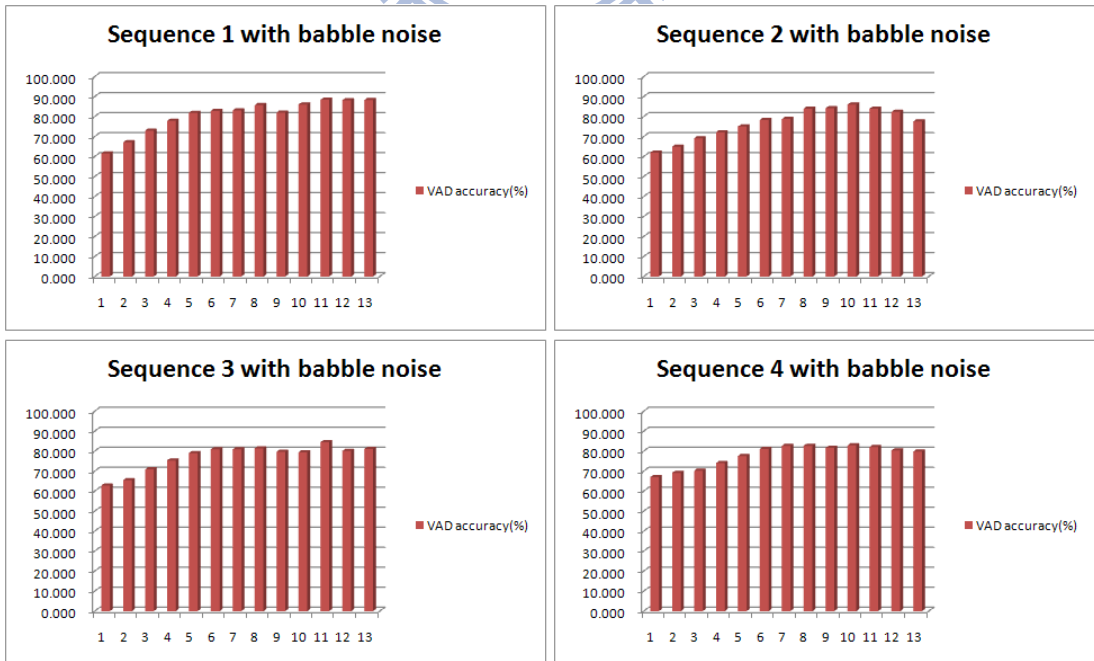


Fig. 4-2 VAD result for babble noise

VAD accuracy result: factory noise (%)

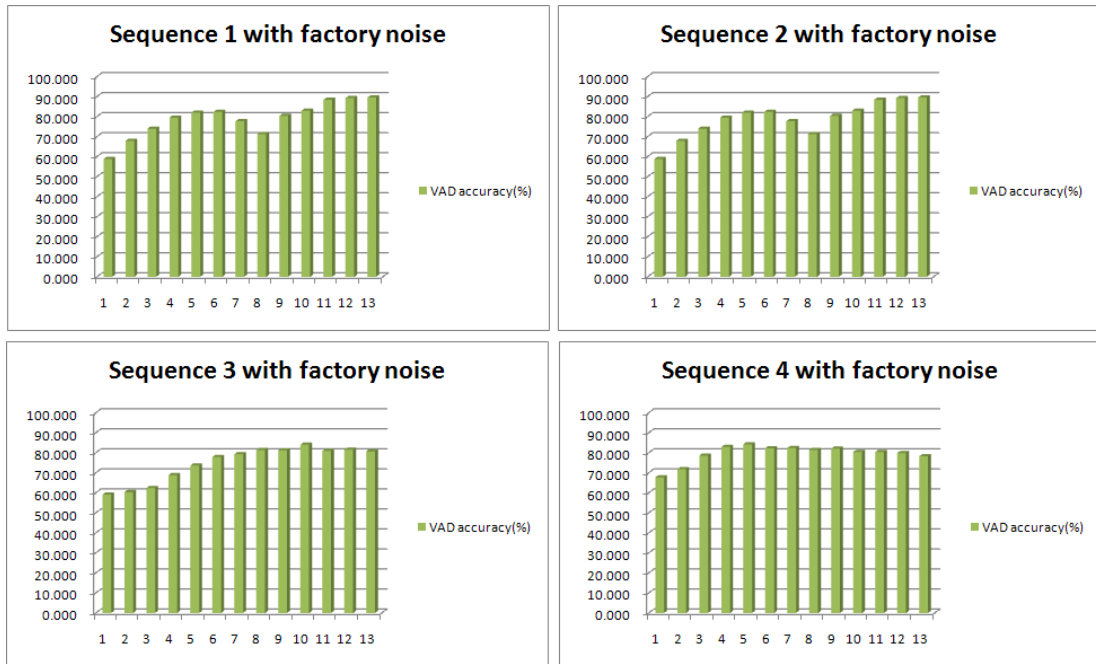


Fig. 4-3 VAD result for factory noise

VAD accuracy result: car noise (%)

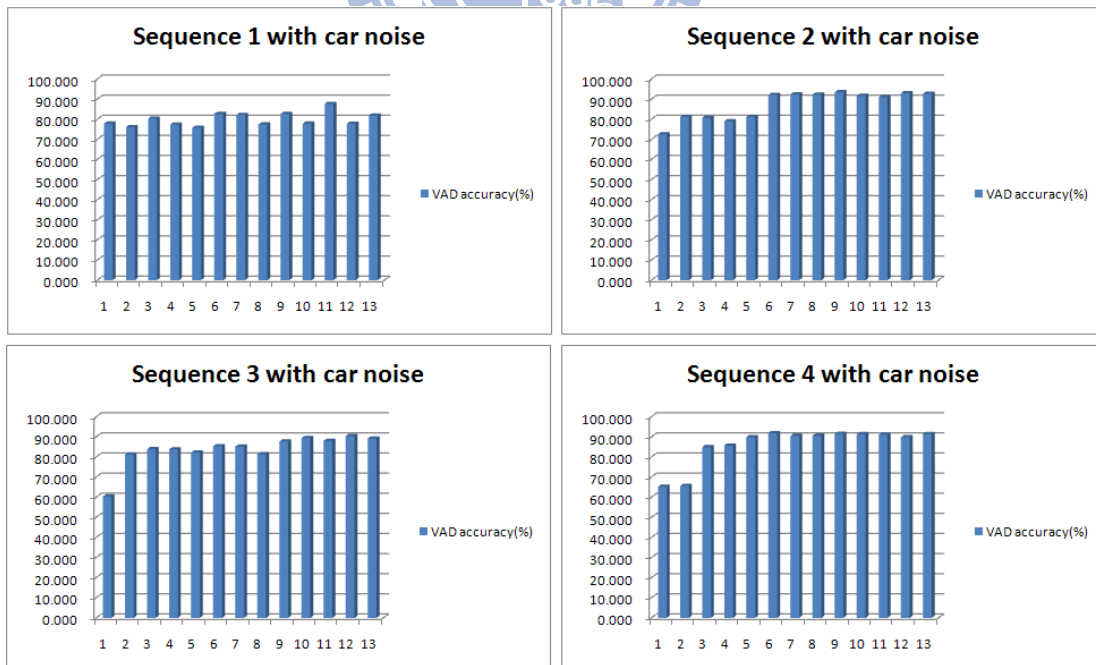
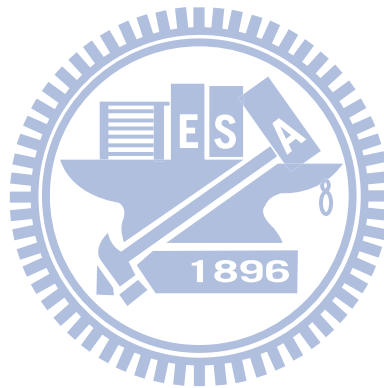


Fig. 4-4 VAD result for car noise

Table 4-1 to Table 4-16 show the segment SNR results and PESQ scores for the proposed algorithm. Fig. 4-5 to Fig. 4-20 illustrate the improvement in segment SNR for proposed algorithm on the left and depicts the PESQ scores for original noisy speech sequence, processed speech sequence and the improvement on the right respectively. Table 4-17 to Table 4-32 show the result of segment SNR and PESQ scores for proposed spectral subtraction with ideal VAD and Fig. 4-21 to Fig. 4-36 plot the corresponding comparison between the result of spectral subtraction between the experiments with proposed VAD and the experiments with ideal VAD. The simulation results will be discussed and analyzed in the aspect of original SNR and noise types respectively.



Sequence 1 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	8.395	8.395	1.003	1.499	0.496
1	9.171	8.171	1.041	1.539	0.498
2	9.626	7.626	1.065	1.610	0.546
3	9.943	6.943	1.098	1.670	0.572
4	10.068	6.068	1.148	1.723	0.575
5	10.046	5.046	1.209	1.778	0.570
6	11.727	5.727	1.267	1.902	0.635
7	12.640	5.640	1.322	2.054	0.732
8	13.283	5.283	1.374	2.111	0.737
9	13.964	4.964	1.451	2.194	0.743
10	14.232	4.232	1.526	2.260	0.734
11	14.169	3.169	1.606	2.281	0.675
12	14.139	2.139	1.688	2.323	0.635

Table 4-1 Segment SNR results for sequence 1 with white noise

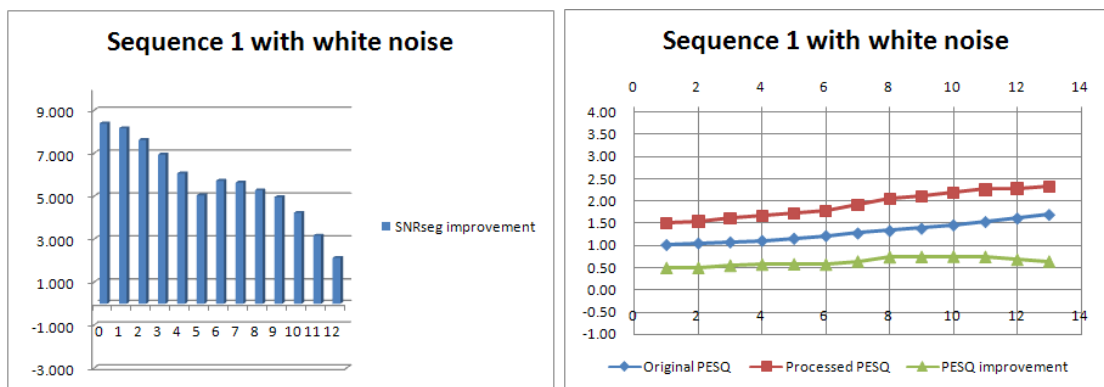


Fig. 4-5 Segment SNR improvement and PESQ scores

Sequence 2 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	3.077	3.077	0.748	0.794	0.046
1	4.364	3.364	0.794	0.884	0.091
2	5.635	3.635	0.843	0.980	0.136
3	6.869	3.869	0.900	1.088	0.188
4	8.065	4.065	0.967	1.224	0.257
5	9.178	4.178	1.039	1.366	0.327
6	11.370	5.370	1.103	1.854	0.751
7	11.487	4.487	1.177	1.901	0.723
8	12.025	4.025	1.265	1.992	0.726
9	11.431	2.431	1.363	2.022	0.660
10	12.391	2.391	1.460	2.134	0.674
11	11.990	0.990	1.565	2.183	0.618
12	12.368	0.368	1.666	2.330	0.664

Table 4-2 Segment SNR results for sequence 2 with white noise

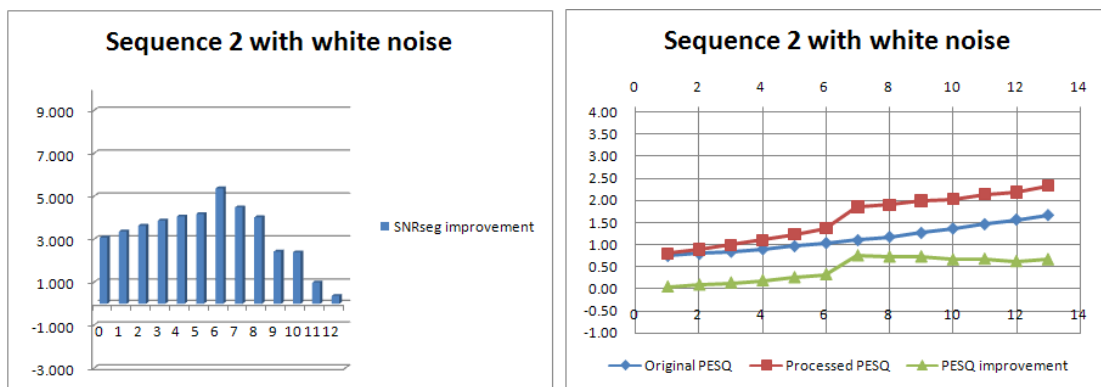


Fig. 4-6 Segment SNR improvement and PESQ scores

Sequence 3 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.839	2.839	1.070	1.124	0.054
1	4.128	3.128	1.102	1.191	0.089
2	5.378	3.378	1.138	1.272	0.134
3	6.736	3.736	1.182	1.366	0.184
4	9.050	5.050	1.230	1.711	0.481
5	9.674	4.674	1.286	1.951	0.665
6	10.755	4.755	1.342	2.076	0.734
7	11.164	4.164	1.415	2.133	0.718
8	11.274	3.274	1.486	2.191	0.706
9	12.829	3.829	1.561	2.344	0.783
10	13.168	3.168	1.646	2.385	0.739
11	12.553	1.553	1.740	2.392	0.652
12	12.873	0.873	1.838	2.455	0.617

Table 4-3 Segment SNR results for sequence 3 with white noise

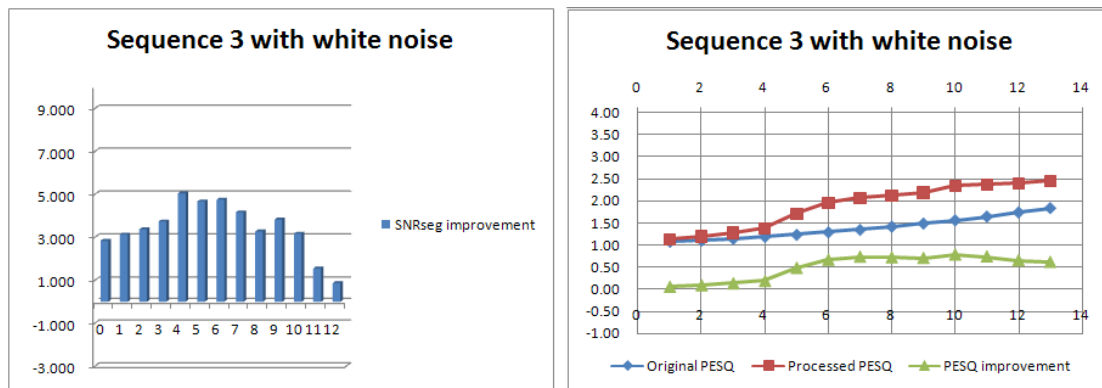


Fig. 4-7 Segment SNR improvement and PESQ scores

Sequence 4 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	3.042	3.042	0.994	1.124	0.130
1	4.396	3.396	1.055	1.216	0.160
2	5.731	3.731	1.120	1.323	0.203
3	7.101	4.101	1.189	1.443	0.254
4	8.423	4.423	1.259	1.556	0.297
5	9.904	4.904	1.337	1.750	0.413
6	11.914	5.914	1.420	2.170	0.750
7	12.580	5.580	1.492	2.249	0.757
8	13.025	5.025	1.573	2.331	0.758
9	13.574	4.574	1.659	2.395	0.735
10	13.131	3.131	1.755	2.432	0.677
11	12.500	1.500	1.852	2.455	0.603
12	12.737	0.737	1.931	2.520	0.589

Table 4-4 Segment SNR results for sequence 4 with white noise

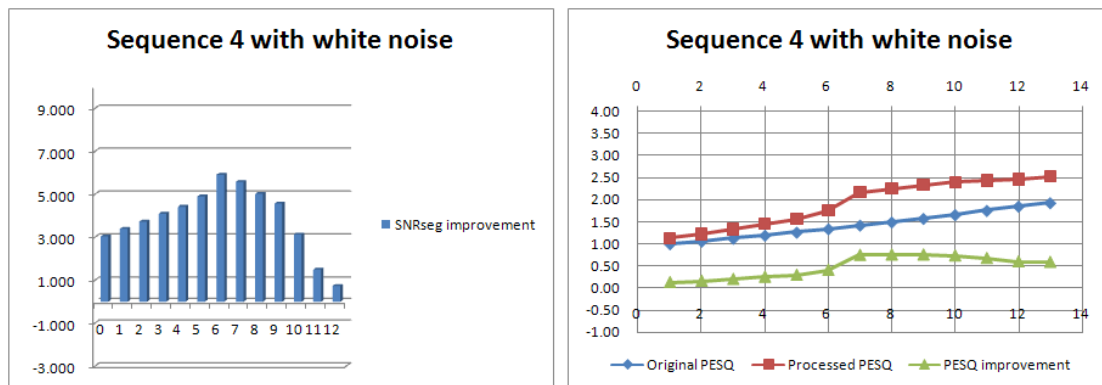


Fig. 4-8 Segment SNR improvement and PESQ scores

Sequence 1 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.919	2.919	1.283	1.498	0.214
1	4.294	3.294	1.326	1.546	0.220
2	5.331	3.331	1.369	1.629	0.260
3	6.267	3.267	1.422	1.711	0.289
4	7.019	3.019	1.475	1.778	0.303
5	7.507	2.507	1.531	1.851	0.320
6	8.177	2.177	1.573	1.907	0.333
7	9.208	2.208	1.639	1.982	0.343
8	10.708	2.708	1.712	2.052	0.340
9	11.262	2.262	1.775	2.103	0.327
10	11.891	1.891	1.851	2.184	0.333
11	12.752	1.752	1.940	2.234	0.293
12	13.013	1.013	2.006	2.301	0.294

Table 4-5 Segment SNR results for sequence 1 with babble noise

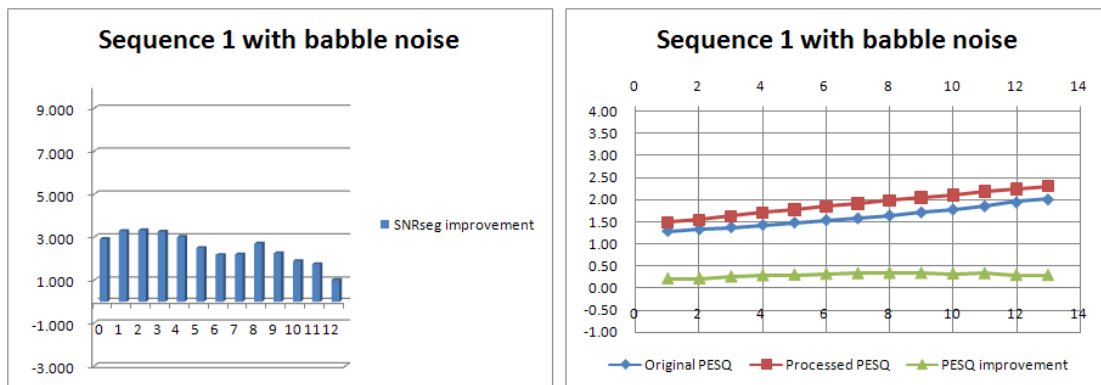


Fig. 4-9 Segment SNR improvement and PESQ scores

Sequence 2 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	0.858	0.858	1.283	1.498	0.214
1	2.760	1.760	1.326	1.546	0.220
2	4.435	2.435	1.369	1.629	0.260
3	5.661	2.661	1.422	1.711	0.289
4	6.610	2.610	1.475	1.778	0.303
5	7.792	2.792	1.531	1.851	0.320
6	8.656	2.656	1.573	1.907	0.333
7	9.564	2.564	1.639	1.982	0.343
8	10.194	2.194	1.712	2.052	0.340
9	11.204	2.204	1.775	2.103	0.327
10	11.846	1.846	1.851	2.184	0.333
11	12.231	1.231	1.940	2.234	0.293
12	10.219	-1.781	2.006	2.301	0.294

Table 4-6 Segment SNR results for sequence 2 with babble noise

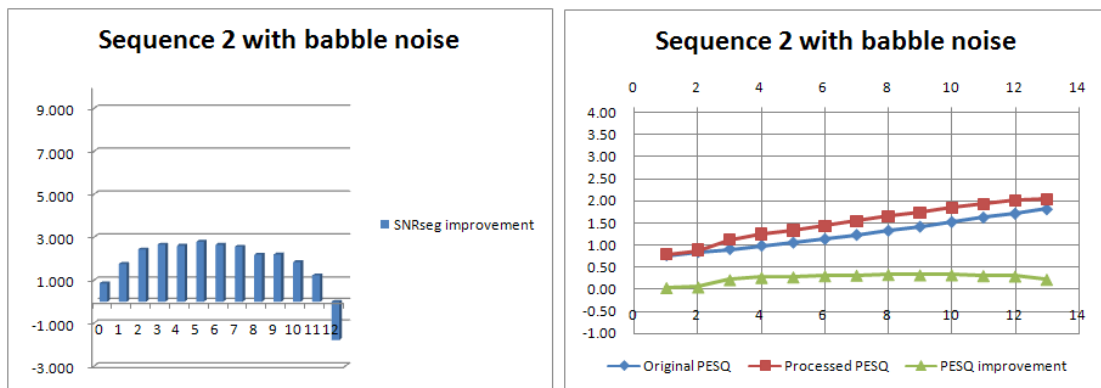


Fig. 4-10 Segment SNR improvement and PESQ scores

Sequence 3 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	0.498	0.498	1.234	1.265	0.031
1	1.577	0.577	1.291	1.329	0.038
2	4.820	2.820	1.349	1.570	0.221
3	5.634	2.634	1.415	1.633	0.217
4	6.519	2.519	1.470	1.729	0.260
5	7.235	2.235	1.521	1.821	0.299
6	7.930	1.930	1.583	1.920	0.337
7	8.649	1.649	1.653	2.025	0.373
8	8.942	0.942	1.733	2.048	0.315
9	9.571	0.571	1.800	2.187	0.388
10	11.040	1.040	1.874	2.243	0.368
11	10.942	-0.058	1.966	2.297	0.331
12	11.756	-0.244	2.055	2.381	0.326

Table 4-7 Segment SNR results for sequence 3 with babble noise

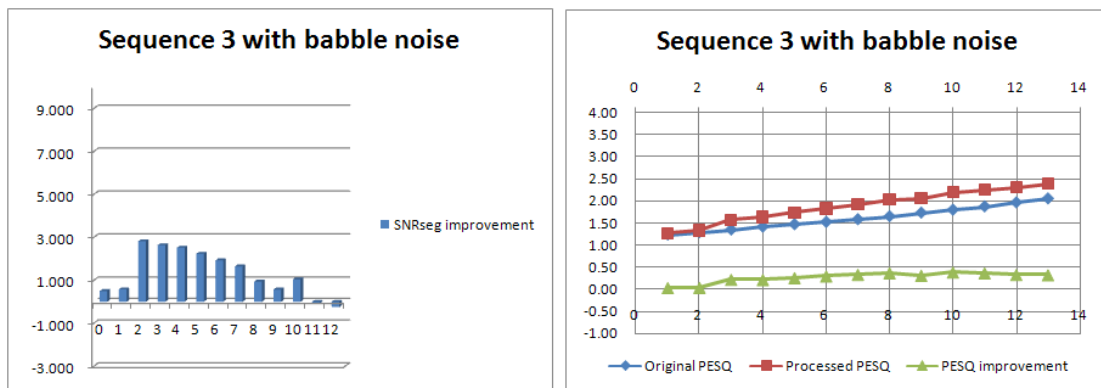


Fig. 4-11 Segment SNR improvement and PESQ scores

Sequence 4 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	1.091	1.091	1.139	1.199	0.060
1	2.228	1.228	1.172	1.256	0.085
2	3.330	1.330	1.283	1.371	0.088
3	5.411	2.411	1.354	1.545	0.192
4	6.469	2.469	1.400	1.620	0.220
5	7.746	2.746	1.482	1.792	0.309
6	8.497	2.497	1.557	1.920	0.363
7	9.436	2.436	1.639	2.052	0.412
8	10.105	2.105	1.708	2.113	0.405
9	11.027	2.027	1.810	2.211	0.401
10	11.479	1.479	1.888	2.285	0.398
11	12.116	1.116	1.980	2.362	0.382
12	12.396	0.396	2.085	2.395	0.310

Table 4-8 Segment SNR results for sequence 4 with babble noise

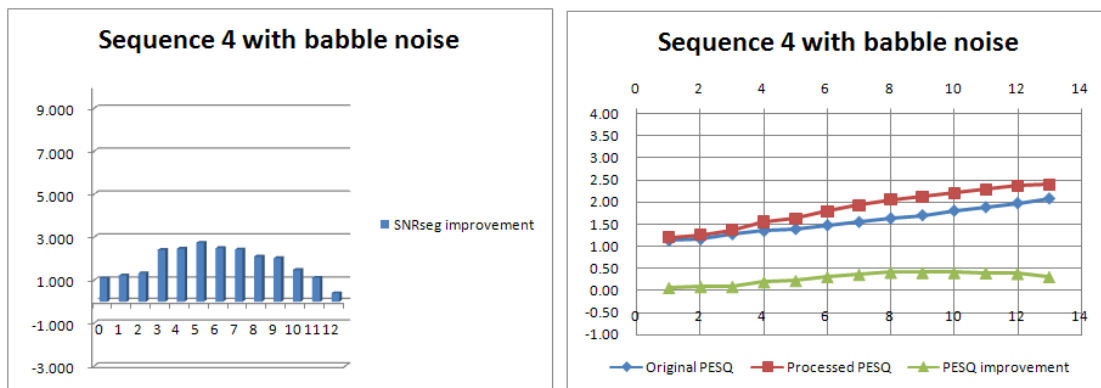


Fig. 4-12 Segment SNR improvement and PESQ scores

Sequence 1 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	4.912	4.912	1.074	1.406	0.332
1	6.001	5.001	1.112	1.490	0.377
2	6.749	4.749	1.150	1.523	0.372
3	7.359	4.359	1.207	1.610	0.403
4	7.851	3.851	1.259	1.646	0.387
5	8.248	3.248	1.295	1.710	0.415
6	10.268	4.268	1.357	1.854	0.496
7	11.187	4.187	1.421	1.956	0.535
8	11.934	3.934	1.489	2.022	0.533
9	12.561	3.561	1.537	2.102	0.565
10	13.133	3.133	1.610	2.166	0.556
11	13.096	2.096	1.703	2.253	0.550
12	13.685	1.685	1.787	2.304	0.517

Table 4-9 Segment SNR results for sequence 1 with factory noise

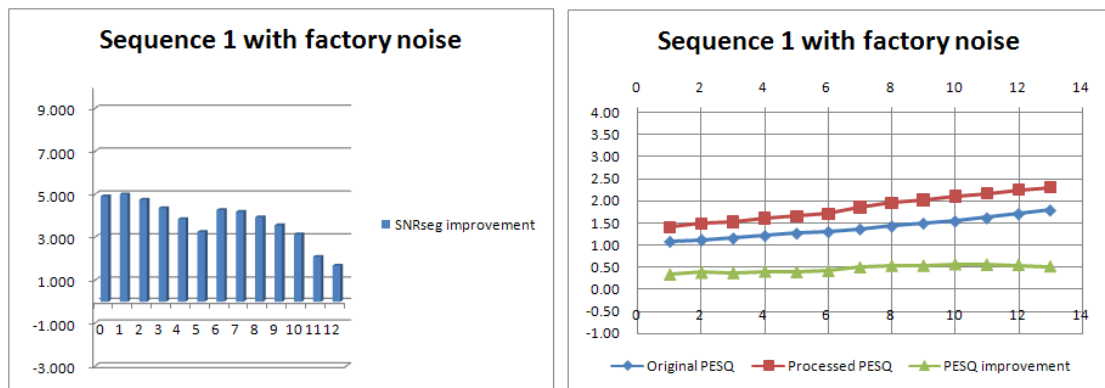


Fig. 4-13 Segment SNR improvement and PESQ scores

Sequence 2 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.646	2.646	0.720	0.762	0.042
1	3.791	2.791	0.776	0.842	0.066
2	4.865	2.865	0.832	0.932	0.100
3	6.458	3.458	0.898	1.097	0.199
4	8.394	4.394	0.965	1.333	0.368
5	9.035	4.035	1.034	1.433	0.398
6	9.861	3.861	1.102	1.585	0.483
7	10.627	3.627	1.180	1.684	0.504
8	11.109	3.109	1.250	1.666	0.416
9	11.856	2.856	1.342	1.807	0.466
10	12.150	2.150	1.434	1.898	0.464
11	12.193	1.193	1.540	1.922	0.382
12	11.259	-0.741	1.642	2.078	0.436

Table 4-10 Segment SNR results for sequence 2 with factory noise

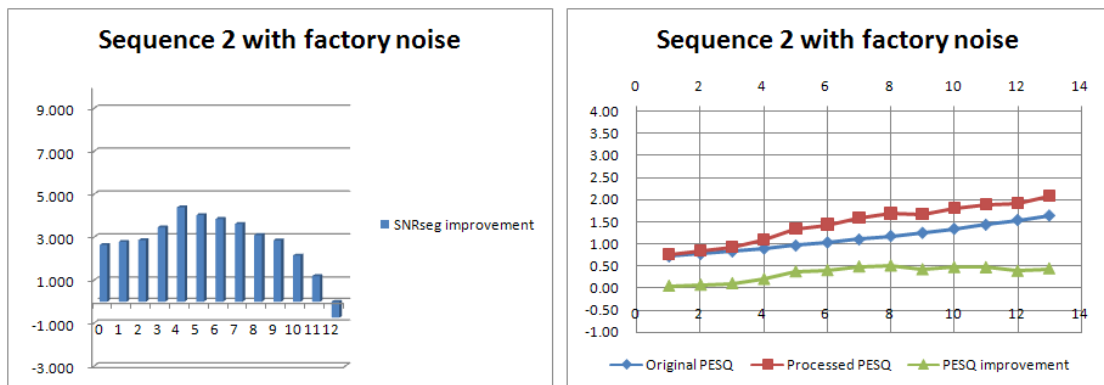


Fig. 4-14 Segment SNR improvement and PESQ scores

Sequence 3 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.434	2.434	1.143	1.241	0.098
1	3.488	2.488	1.187	1.303	0.116
2	4.586	2.586	1.233	1.373	0.140
3	5.698	2.698	1.287	1.458	0.171
4	7.903	3.903	1.338	1.710	0.372
5	8.555	3.555	1.399	1.794	0.395
6	9.320	3.320	1.463	1.887	0.424
7	9.665	2.665	1.514	1.937	0.423
8	9.944	1.944	1.564	2.015	0.450
9	11.056	2.056	1.656	2.135	0.478
10	11.110	1.110	1.744	2.189	0.445
11	11.277	0.277	1.835	2.249	0.414
12	12.175	0.175	1.942	2.332	0.391

Table 4-11 Segment SNR results for sequence 3 with factory noise

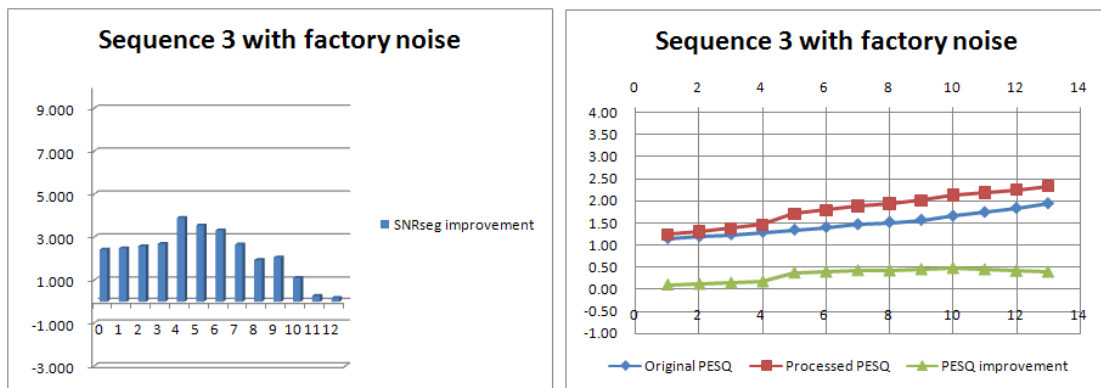


Fig. 4-15 Segment SNR improvement and PESQ scores

Sequence 4 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.993	2.993	1.035	1.171	0.135
1	4.103	3.103	1.096	1.259	0.163
2	5.453	3.453	1.169	1.418	0.249
3	7.442	4.442	1.243	1.686	0.444
4	8.358	4.358	1.323	1.809	0.485
5	9.177	4.177	1.404	1.892	0.488
6	9.953	3.953	1.489	1.993	0.504
7	10.339	3.339	1.548	2.048	0.500
8	11.336	3.336	1.636	2.128	0.492
9	11.418	2.418	1.727	2.226	0.498
10	12.157	2.157	1.827	2.268	0.441
11	12.588	1.588	1.926	2.361	0.435
12	12.473	0.473	2.017	2.424	0.407

Table 4-12 Segment SNR results for sequence 4 with factory noise

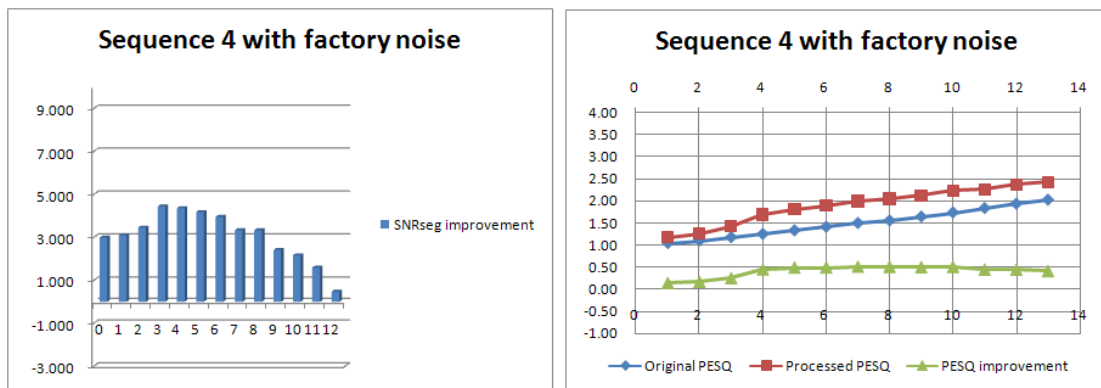


Fig. 4-16 Segment SNR improvement and PESQ scores

Sequence 1 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	12.331	12.331	2.701	2.728	0.028
1	13.995	12.995	2.759	2.844	0.085
2	15.752	13.752	2.818	2.890	0.072
3	19.788	16.788	2.872	3.133	0.260
4	20.964	16.964	2.928	3.069	0.142
5	21.672	16.672	2.966	3.061	0.094
6	23.633	17.633	3.028	3.047	0.020
7	24.591	17.591	3.079	3.100	0.021
8	25.668	17.668	3.141	3.160	0.019
9	26.529	17.529	3.189	3.214	0.025
10	27.688	17.688	3.254	3.274	0.020
11	28.561	17.561	3.298	3.318	0.020
12	29.785	17.785	3.357	3.373	0.015

Table 4-13 Segment SNR results for sequence 1 with car noise

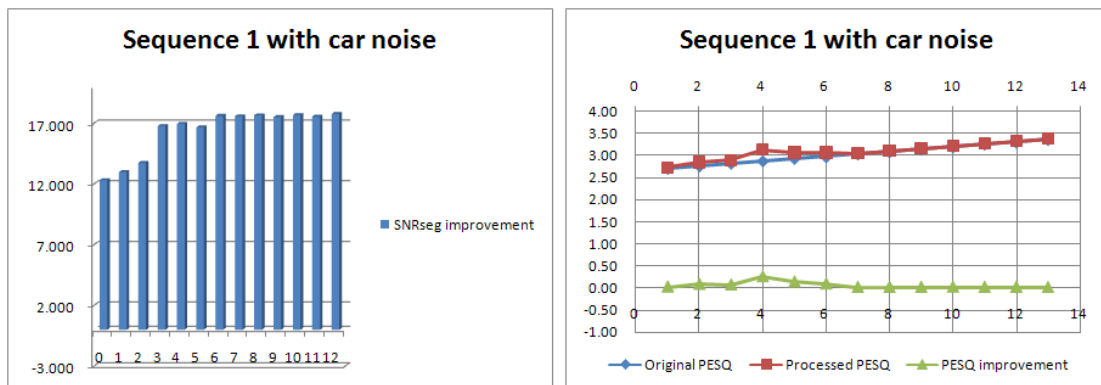


Fig. 4-17 Segment SNR improvement and PESQ scores

Sequence 2 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	14.839	14.839	2.625	2.701	0.076
1	15.069	14.069	2.712	2.775	0.063
2	15.279	13.279	2.791	2.838	0.047
3	21.763	18.763	2.873	2.936	0.063
4	15.648	11.648	2.939	2.808	-0.131
5	23.495	18.495	3.015	3.073	0.058
6	24.486	18.486	3.087	3.146	0.059
7	15.976	8.976	3.167	2.991	-0.176
8	15.934	7.934	3.244	3.064	-0.179
9	27.476	18.476	3.315	3.367	0.052
10	28.546	18.546	3.384	3.436	0.052
11	16.140	5.140	3.447	3.140	-0.307
12	30.601	18.601	3.519	3.565	0.046

Table 4-14 Segment SNR results for sequence 2 with car noise

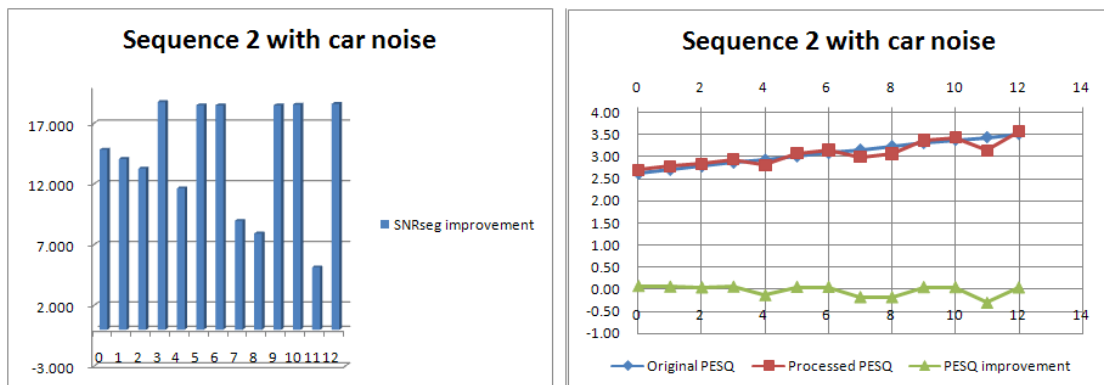


Fig. 4-18 Segment SNR improvement and PESQ scores

Sequence 3 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	15.983	15.983	2.860	2.836	-0.024
1	17.504	16.504	2.924	3.222	0.298
2	15.527	13.527	2.992	3.175	0.183
3	17.157	14.157	3.060	3.213	0.153
4	20.944	16.944	3.098	3.144	0.046
5	21.638	16.638	3.150	3.195	0.044
6	22.469	16.469	3.203	3.226	0.023
7	17.210	10.210	3.265	3.464	0.199
8	17.984	9.984	3.313	3.358	0.045
9	23.856	14.856	3.371	3.378	0.007
10	27.280	17.280	3.412	3.447	0.035
11	29.966	18.966	3.455	3.461	0.006
12	31.087	19.087	3.503	3.504	0.001

Table 4-15 Segment SNR results for sequence 3 with car noise

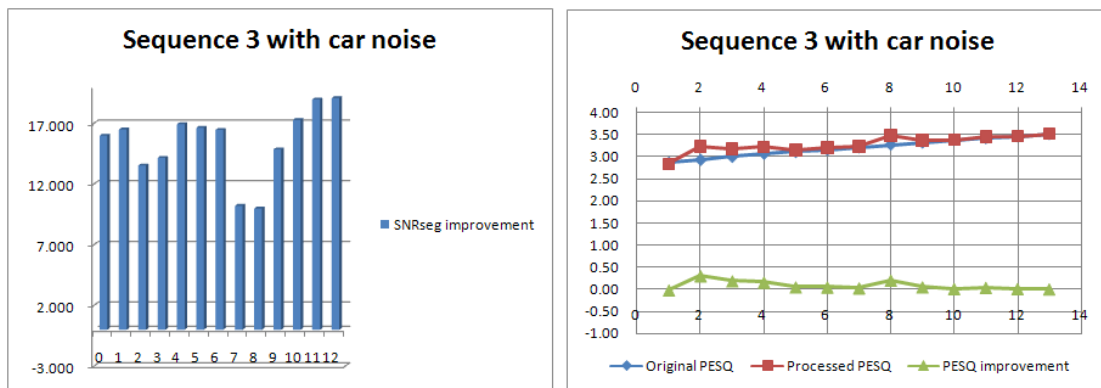


Fig. 4-19 Segment SNR improvement and PESQ scores

Sequence 4 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	19.925	19.925	2.877	2.925	0.049
1	20.979	19.979	2.947	2.991	0.045
2	21.968	19.968	3.008	3.050	0.043
3	20.526	17.526	3.049	3.324	0.275
4	22.253	18.253	3.114	3.289	0.175
5	22.059	17.059	3.169	3.368	0.200
6	22.203	16.203	3.227	3.406	0.179
7	24.822	17.822	3.286	3.360	0.073
8	27.678	19.678	3.346	3.370	0.024
9	28.750	19.750	3.406	3.423	0.017
10	26.028	16.028	3.453	3.529	0.076
11	30.821	19.821	3.500	3.532	0.032
12	31.717	19.717	3.546	3.569	0.023

Table 4-16 Segment SNR results for sequence 4 with car noise

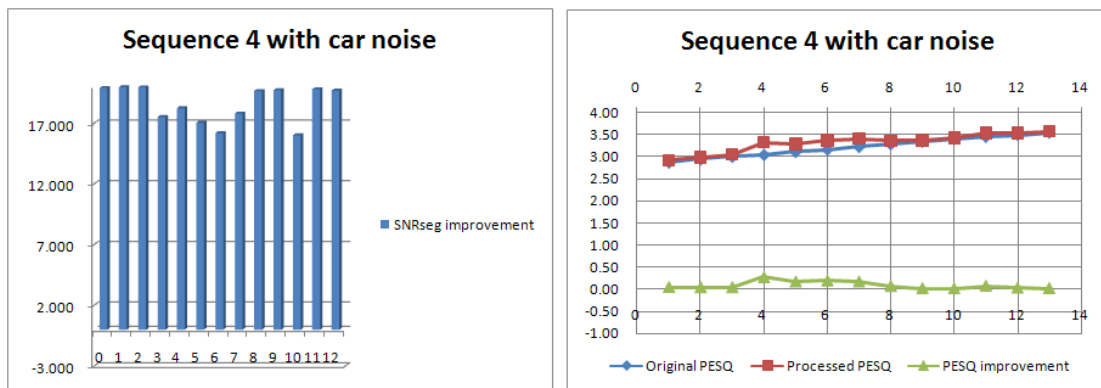


Fig. 4-20 Segment SNR improvement and PESQ scores

Sequence 1 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	8.533	8.533	1.003	1.494	0.491
1	9.187	8.187	1.041	1.556	0.515
2	9.829	7.829	1.065	1.656	0.592
3	10.459	7.459	1.098	1.753	0.654
4	11.067	7.067	1.148	1.849	0.701
5	11.698	6.698	1.209	1.941	0.732
6	12.316	6.316	1.267	2.015	0.748
7	12.946	5.946	1.322	2.084	0.762
8	13.569	5.569	1.374	2.160	0.786
9	14.201	5.201	1.451	2.238	0.787
10	14.775	4.775	1.526	2.308	0.782
11	15.380	4.380	1.606	2.379	0.773
12	15.943	3.943	1.688	2.439	0.751

Table 4-17 Segment SNR results (ideal VAD) for sequence 1 with white noise

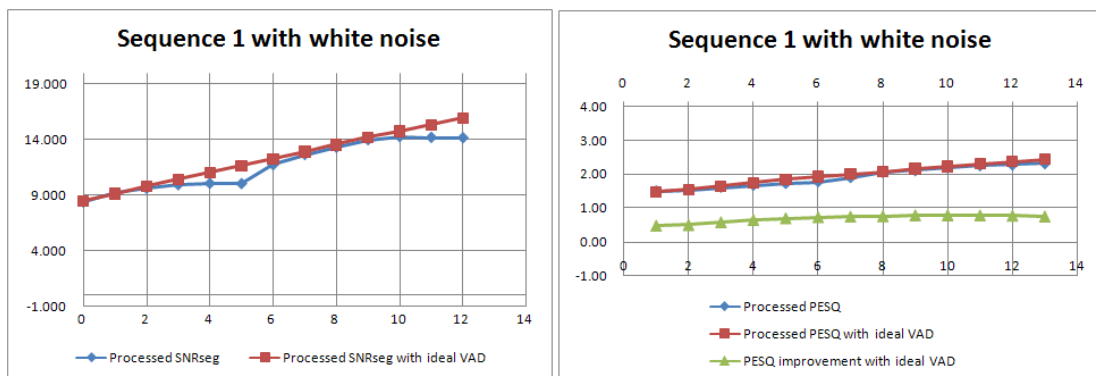


Fig. 4-21 Segment SNR comparison and PESQ score comparison.

Sequence 2 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	8.075	8.075	0.748	1.200	0.453
1	8.668	7.668	0.794	1.338	0.544
2	9.254	7.254	0.843	1.439	0.596
3	9.840	6.840	0.900	1.555	0.655
4	10.381	6.381	0.967	1.668	0.702
5	10.958	5.958	1.039	1.757	0.718
6	11.371	5.371	1.103	1.859	0.756
7	11.926	4.926	1.177	1.935	0.757
8	11.946	3.946	1.265	1.961	0.695
9	12.046	3.046	1.363	2.073	0.710
10	12.484	2.484	1.460	2.129	0.669
11	12.869	1.869	1.565	2.210	0.646
12	13.204	1.204	1.666	2.293	0.627

Table 4-18 Segment SNR results (ideal VAD) for sequence 2 with white noise

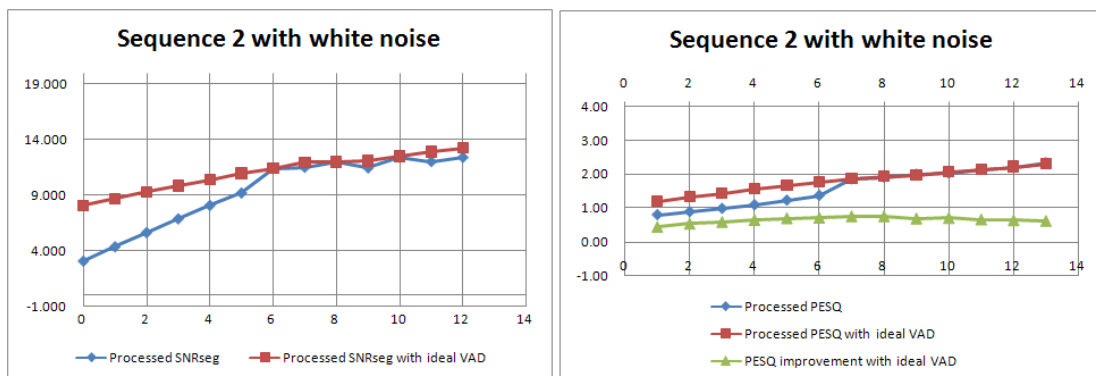


Fig. 4-22 Segment SNR comparison and PESQ score comparison.

Sequence 3 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	7.925	7.925	1.070	1.559	0.488
1	8.524	7.524	1.102	1.657	0.555
2	9.191	7.191	1.138	1.763	0.624
3	9.868	6.868	1.182	1.862	0.680
4	10.428	6.428	1.230	1.958	0.729
5	11.135	6.135	1.286	2.045	0.759
6	11.743	5.743	1.342	2.128	0.786
7	12.269	5.269	1.415	2.215	0.800
8	13.210	5.210	1.486	2.293	0.807
9	13.820	4.820	1.561	2.365	0.804
10	14.375	4.375	1.646	2.448	0.802
11	14.672	3.672	1.740	2.513	0.773
12	15.209	3.209	1.838	2.579	0.741

Table 4-19 Segment SNR results (ideal VAD) for sequence 3 with white noise

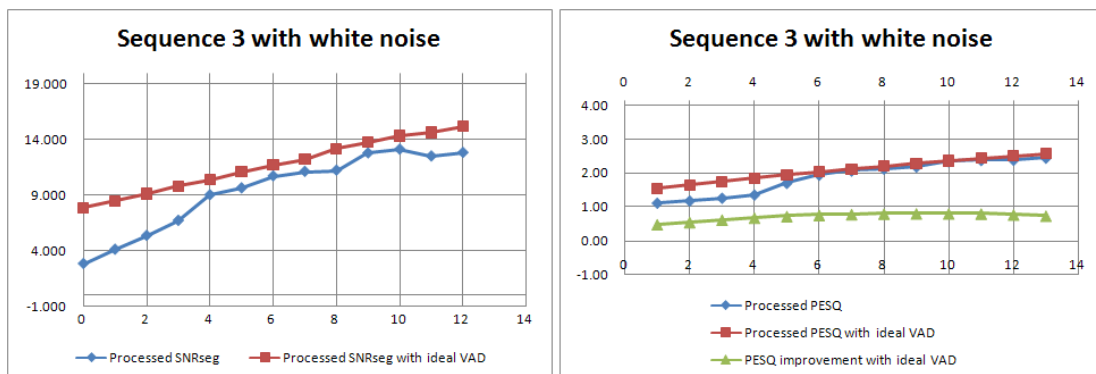


Fig. 4-23 Segment SNR comparison and PESQ score comparison.

Sequence 4 with white noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	6.210	6.210	0.994	1.434	0.440
1	8.951	7.951	1.055	1.739	0.684
2	9.578	7.578	1.120	1.837	0.717
3	10.207	7.207	1.189	1.932	0.743
4	10.823	6.823	1.259	2.034	0.775
5	11.432	6.432	1.337	2.097	0.760
6	12.027	6.027	1.420	2.187	0.767
7	12.640	5.640	1.492	2.268	0.776
8	13.229	5.229	1.573	2.360	0.787
9	13.872	4.872	1.659	2.420	0.761
10	14.431	4.431	1.755	2.492	0.737
11	15.044	4.044	1.852	2.576	0.724
12	15.612	3.612	1.931	2.635	0.704

Table 4-20 Segment SNR results (ideal VAD) for sequence 4 with white noise

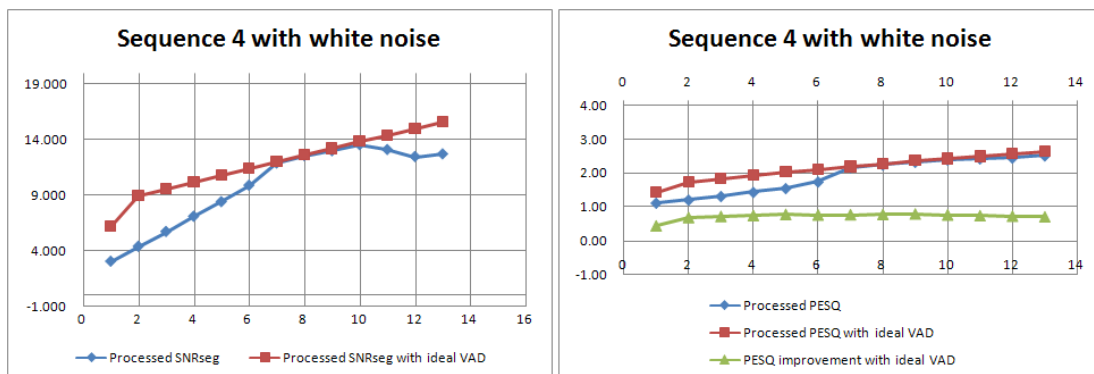


Fig. 4-24 Segment SNR comparison and PESQ score comparison.

Sequence 1 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	4.271	4.271	1.283	1.549	0.265
1	5.157	4.157	1.326	1.618	0.292
2	5.986	3.986	1.369	1.709	0.340
3	6.718	3.718	1.422	1.749	0.327
4	7.767	3.767	1.475	1.826	0.351
5	8.565	3.565	1.531	1.899	0.368
6	9.196	3.196	1.573	1.952	0.379
7	10.082	3.082	1.639	2.020	0.381
8	11.013	3.013	1.712	2.090	0.379
9	11.697	2.697	1.775	2.134	0.358
10	12.666	2.666	1.851	2.219	0.367
11	13.601	2.601	1.940	2.309	0.369
12	14.281	2.281	2.006	2.370	0.363

Table 4-21 Segment SNR results (ideal VAD) for sequence 1 with babble noise

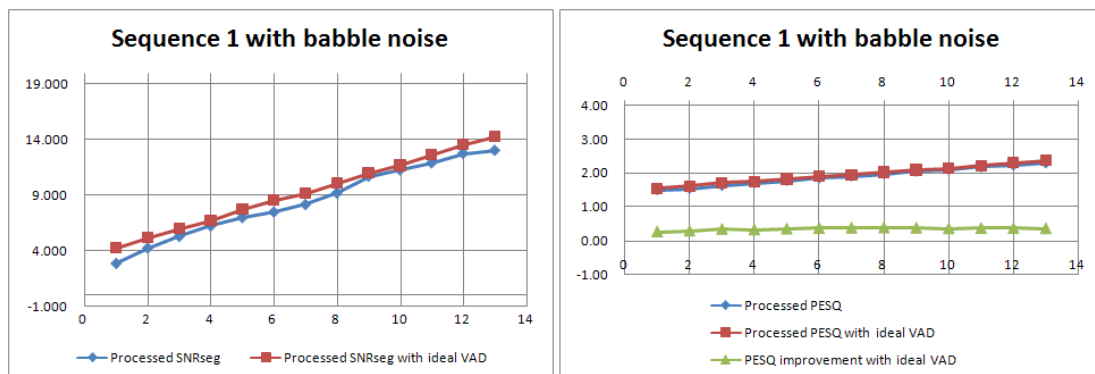


Fig. 4-25 Segment SNR comparison and PESQ score comparison.

Sequence 2 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	3.926	3.926	1.283	0.930	0.170
1	4.894	3.894	1.326	1.007	0.169
2	5.618	3.618	1.369	1.130	0.227
3	6.352	3.352	1.422	1.225	0.245
4	7.087	3.087	1.475	1.308	0.257
5	7.833	2.833	1.531	1.429	0.297
6	8.635	2.635	1.573	1.525	0.298
7	9.532	2.532	1.639	1.657	0.334
8	10.339	2.339	1.712	1.755	0.338
9	11.202	2.202	1.775	1.851	0.326
10	12.044	2.044	1.851	1.931	0.305
11	12.720	1.720	1.940	2.000	0.283
12	13.563	1.563	2.006	2.095	0.278

Table 4-22 Segment SNR results (ideal VAD) for sequence 2 with babble noise

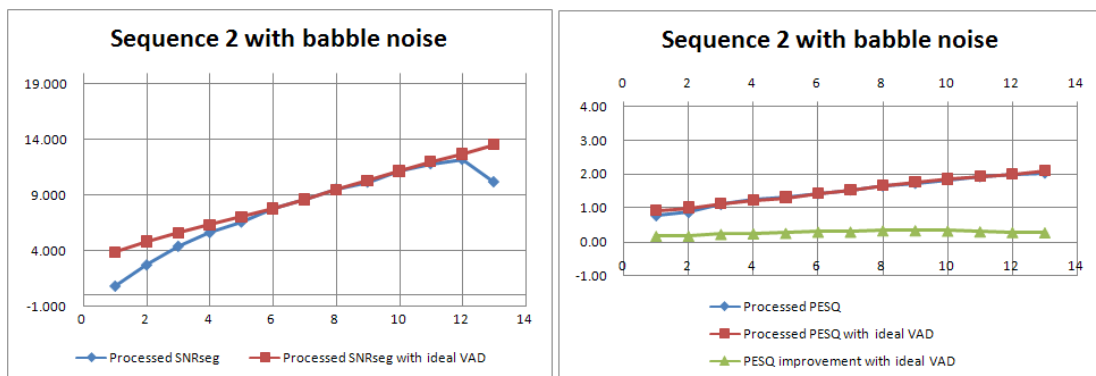


Fig. 4-26 Segment SNR comparison and PESQ score comparison.

Sequence 3 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	3.691	3.691	1.234	1.497	0.263
1	4.442	3.442	1.291	1.573	0.282
2	5.324	3.324	1.349	1.660	0.311
3	6.115	3.115	1.415	1.723	0.308
4	6.845	2.845	1.470	1.808	0.338
5	7.497	2.497	1.521	1.863	0.341
6	8.371	2.371	1.583	1.943	0.360
7	9.141	2.141	1.653	2.025	0.373
8	10.145	2.145	1.733	2.106	0.373
9	10.871	1.871	1.800	2.186	0.386
10	11.657	1.657	1.874	2.261	0.387
11	12.526	1.526	1.966	2.348	0.383
12	13.300	1.300	2.055	2.442	0.387

Table 4-23 Segment SNR results (ideal VAD) for sequence 3 with babble noise

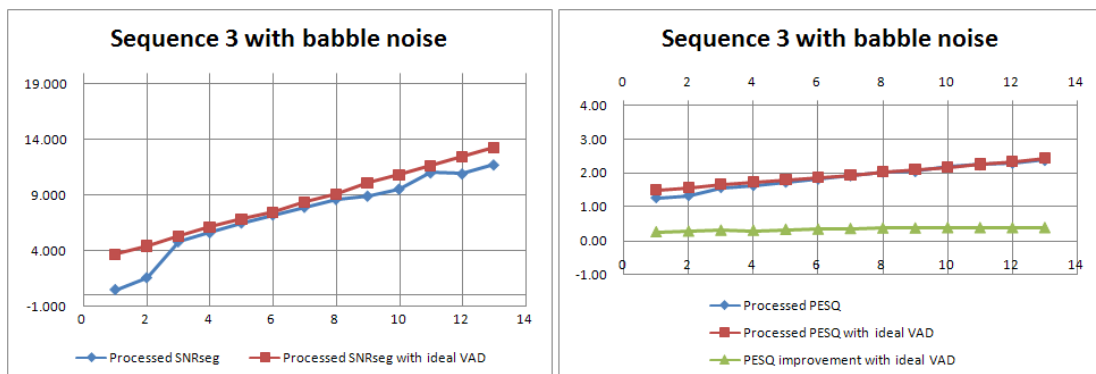


Fig. 4-27 Segment SNR comparison and PESQ score comparison.

Sequence 4 with babble noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	2.633	2.633	1.139	1.331	0.192
1	4.360	3.360	1.172	1.475	0.304
2	5.339	3.339	1.283	1.555	0.271
3	6.207	3.207	1.354	1.677	0.324
4	6.827	2.827	1.400	1.698	0.298
5	7.690	2.690	1.482	1.815	0.333
6	8.630	2.630	1.557	1.908	0.351
7	9.570	2.570	1.639	2.022	0.382
8	10.359	2.359	1.708	2.108	0.400
9	11.351	2.351	1.810	2.176	0.366
10	12.212	2.212	1.888	2.267	0.380
11	13.073	2.073	1.980	2.364	0.384
12	13.908	1.908	2.085	2.429	0.344

Table 4-24 Segment SNR results (ideal VAD) for sequence 4 with babble noise

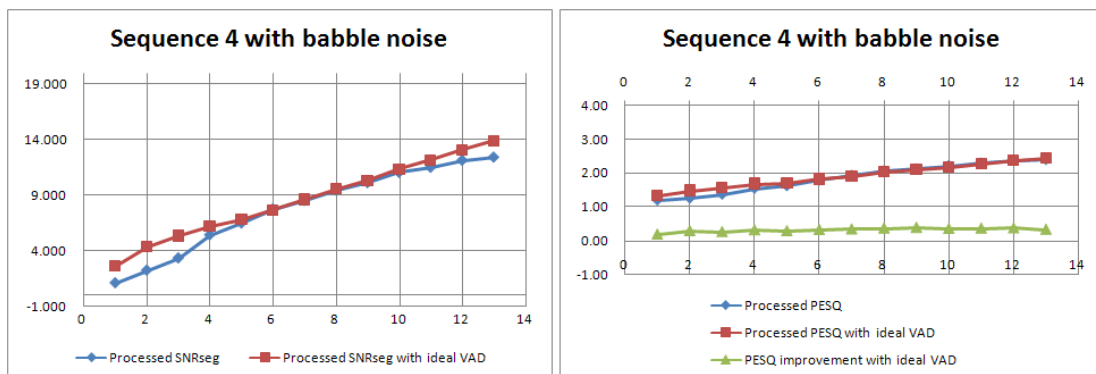


Fig. 4-28 Segment SNR comparison and PESQ score comparison.

Sequence 1 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	5.411	5.411	1.074	1.454	0.380
1	6.150	5.150	1.112	1.515	0.403
2	6.908	4.908	1.150	1.582	0.431
3	7.840	4.840	1.207	1.694	0.487
4	8.732	4.732	1.259	1.808	0.549
5	9.590	4.590	1.295	1.898	0.603
6	10.475	4.475	1.357	1.988	0.631
7	11.272	4.272	1.421	2.042	0.621
8	12.117	4.117	1.489	2.119	0.630
9	12.705	3.705	1.537	2.174	0.637
10	13.574	3.574	1.610	2.264	0.654
11	14.529	3.529	1.703	2.346	0.643
12	15.324	3.324	1.787	2.416	0.629

Table 4-25 Segment SNR results (ideal VAD) for sequence 1 with factory noise

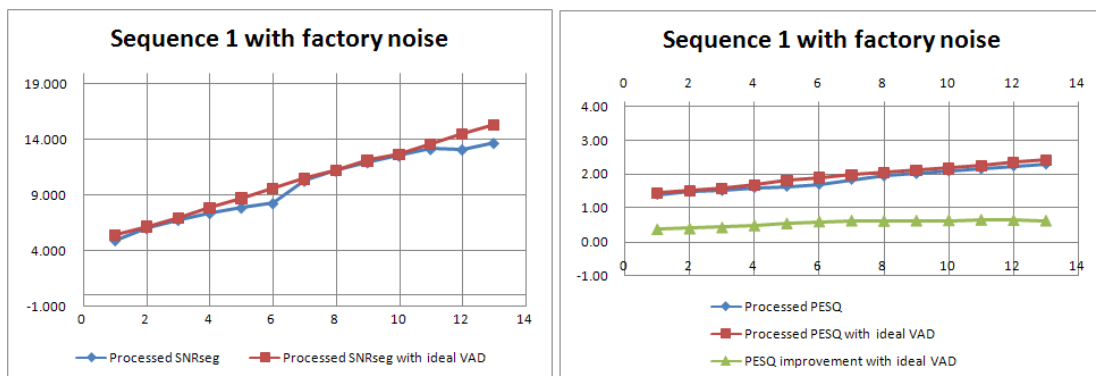


Fig. 4-29 Segment SNR comparison and PESQ score comparison.

Sequence 2 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	5.159	5.159	0.720	0.987	0.267
1	5.902	4.902	0.776	1.091	0.315
2	6.612	4.612	0.832	1.146	0.314
3	7.604	4.604	0.898	1.289	0.391
4	8.225	4.225	0.965	1.396	0.431
5	9.116	4.116	1.034	1.462	0.427
6	9.988	3.988	1.102	1.611	0.509
7	10.510	3.510	1.180	1.587	0.407
8	11.353	3.353	1.250	1.758	0.508
9	12.058	3.058	1.342	1.894	0.553
10	12.477	2.477	1.434	1.807	0.372
11	13.336	2.336	1.540	1.940	0.400
12	13.748	1.748	1.642	2.039	0.397

Table 4-26 Segment SNR results (ideal VAD) for sequence 2 with factory noise

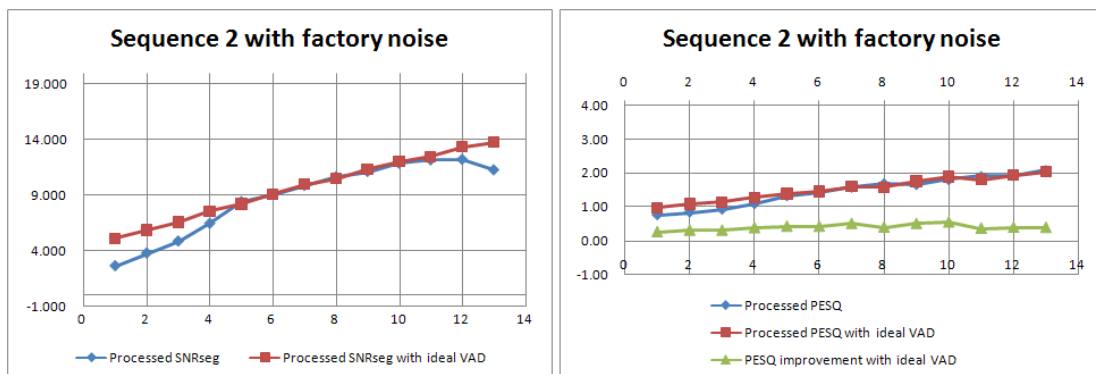


Fig. 4-30 Segment SNR comparison and PESQ score comparison.

Sequence 3 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	4.743	4.743	1.143	1.424	0.280
1	5.604	4.604	1.187	1.515	0.328
2	6.308	4.308	1.233	1.560	0.326
3	7.120	4.120	1.287	1.650	0.363
4	7.916	3.916	1.338	1.757	0.419
5	8.800	3.800	1.399	1.821	0.422
6	9.533	3.533	1.463	1.915	0.453
7	10.264	3.264	1.514	2.028	0.514
8	10.921	2.921	1.564	2.067	0.503
9	11.716	2.716	1.656	2.159	0.502
10	12.620	2.620	1.744	2.273	0.529
11	13.588	2.588	1.835	2.330	0.495
12	14.309	2.309	1.942	2.408	0.466

Table 4-27 Segment SNR results (ideal VAD) for sequence 3 with factory noise

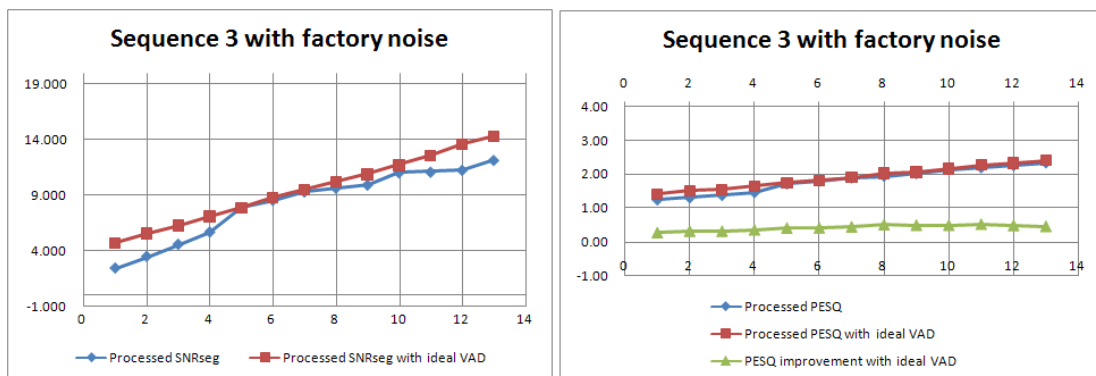


Fig. 4-31 Segment SNR comparison and PESQ score comparison.

Sequence 4 with factory noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	3.777	3.777	1.035	1.299	0.263
1	5.698	4.698	1.096	1.511	0.414
2	6.640	4.640	1.169	1.598	0.429
3	7.500	4.500	1.243	1.673	0.431
4	8.366	4.366	1.323	1.816	0.493
5	9.183	4.183	1.404	1.892	0.489
6	10.094	4.094	1.489	1.998	0.509
7	10.719	3.719	1.548	2.050	0.503
8	11.507	3.507	1.636	2.158	0.522
9	12.348	3.348	1.727	2.219	0.492
10	13.322	3.322	1.827	2.320	0.493
11	14.190	3.190	1.926	2.399	0.473
12	14.969	2.969	2.017	2.455	0.438

Table 4-28 Segment SNR results (ideal VAD) for sequence 4 with factory noise

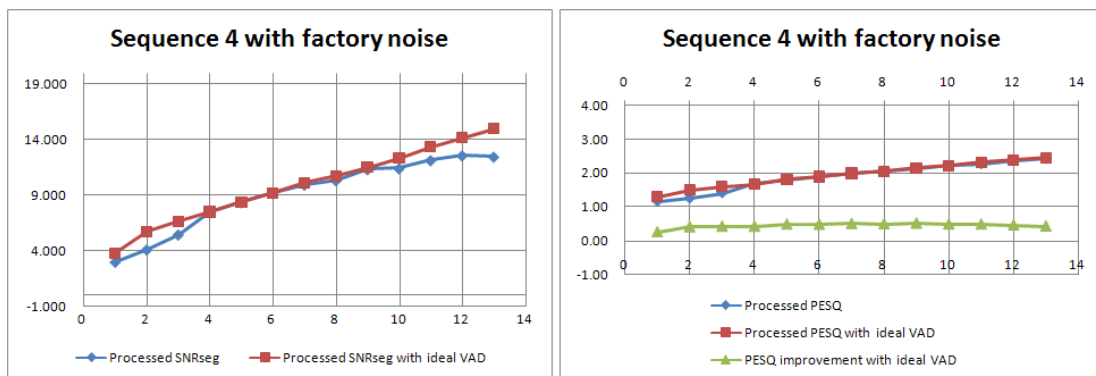


Fig. 4-32 Segment SNR comparison and PESQ score comparison.

Sequence 1 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	18.708	18.708	2.701	3.000	0.299
1	19.302	18.302	2.759	3.060	0.301
2	19.818	17.818	2.818	3.130	0.313
3	20.336	17.336	2.872	3.156	0.284
4	20.755	16.755	2.928	3.219	0.291
5	21.582	16.582	2.966	3.136	0.170
6	22.967	16.967	3.028	3.079	0.052
7	24.591	17.591	3.079	3.101	0.022
8	25.668	17.668	3.141	3.162	0.021
9	26.528	17.528	3.189	3.208	0.019
10	27.688	17.688	3.254	3.272	0.018
11	28.558	17.558	3.298	3.315	0.017
12	29.781	17.781	3.357	3.371	0.013

Table 4-29 Segment SNR results (ideal VAD) for sequence 1 with car noise

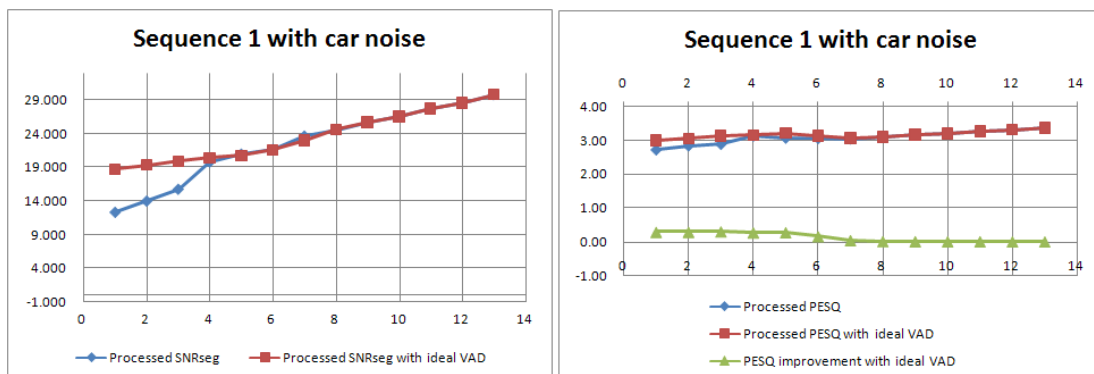


Fig. 4-33 Segment SNR comparison and PESQ score comparison.

Sequence 2 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	14.942	14.942	2.625	2.719	0.094
1	15.131	14.131	2.712	2.781	0.070
2	15.303	13.303	2.791	2.848	0.057
3	21.769	18.769	2.873	2.936	0.063
4	22.528	18.528	2.939	3.001	0.061
5	23.494	18.494	3.015	3.073	0.058
6	24.439	18.439	3.087	3.138	0.051
7	25.496	18.496	3.167	3.214	0.047
8	26.518	18.518	3.244	3.295	0.051
9	27.474	18.474	3.315	3.367	0.052
10	28.546	18.546	3.384	3.436	0.052
11	29.514	18.514	3.447	3.496	0.049
12	30.601	18.601	3.519	3.565	0.046

Table 4-30 Segment SNR results (ideal VAD) for sequence 2 with car noise

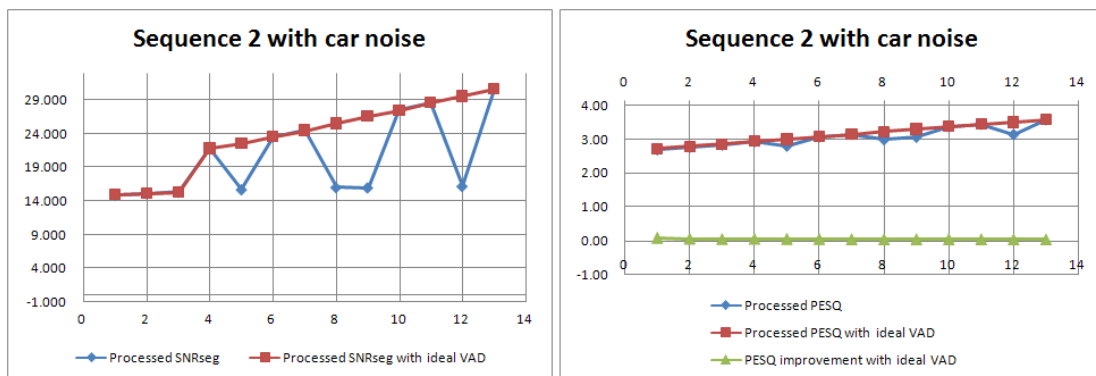


Fig. 4-34 Segment SNR comparison and PESQ score comparison.

Sequence 3 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	17.415	17.415	2.860	3.211	0.351
1	17.800	16.800	2.924	3.273	0.349
2	17.542	15.542	2.992	3.302	0.311
3	20.815	17.815	3.060	3.125	0.065
4	21.195	17.195	3.098	3.146	0.047
5	23.945	18.945	3.150	3.161	0.010
6	24.919	18.919	3.203	3.211	0.008
7	26.015	19.015	3.265	3.265	-0.001
8	26.894	18.894	3.313	3.306	-0.007
9	28.080	19.080	3.371	3.365	-0.006
10	28.971	18.971	3.412	3.408	-0.004
11	29.966	18.966	3.455	3.451	-0.004
12	31.087	19.087	3.503	3.498	-0.005

Table 4-31 Segment SNR results (ideal VAD) for sequence 3 with car noise

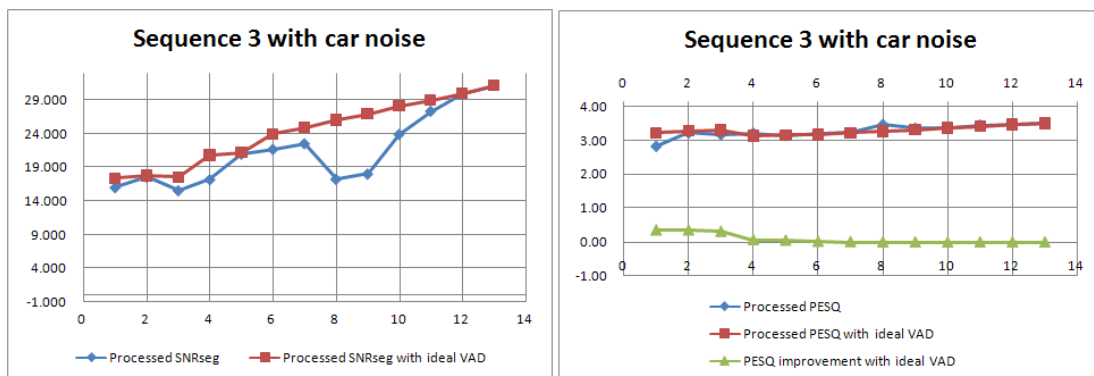


Fig. 4-35 Segment SNR comparison and PESQ score comparison.

Sequence 4 with car noise

(dB)

Original SNRseg	Processed SNRseg	SNRseg improvement	Original PESQ	Processed PESQ	PESQ improvement
0	19.296	19.296	2.877	3.162	0.286
1	20.498	19.498	2.947	3.300	0.353
2	20.968	18.968	3.008	3.295	0.288
3	22.095	19.095	3.049	3.221	0.172
4	22.711	18.711	3.114	3.272	0.158
5	23.172	18.172	3.169	3.320	0.151
6	25.700	19.700	3.227	3.259	0.032
7	26.722	19.722	3.286	3.314	0.028
8	27.678	19.678	3.346	3.369	0.023
9	28.750	19.750	3.406	3.424	0.018
10	29.716	19.716	3.453	3.470	0.017
11	30.800	19.800	3.500	3.523	0.023
12	31.717	19.717	3.546	3.569	0.023

Table 4-32 Segment SNR results (ideal VAD) for sequence 4 with car noise

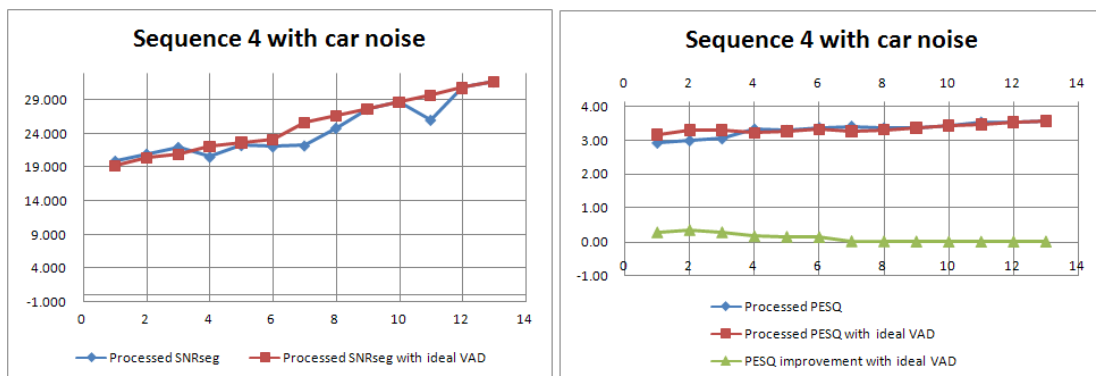


Fig. 4-36 Segment SNR comparison and PESQ score comparison.

4.2.1. Influence of Different Original Segment SNR

It can be seen that the improvement in segment SNR maximizes under medium original SNR, i.e. for segment SNR between 4dB and 8dB. The improvement mainly depends on the accuracy of VAD. Under medium original segment SNR, the VAD is more accurate and the processed SNR becomes much higher than that under low original segment SNR. For high original segment SNR, the improvement is limited since little noise energy variation may cause drastic influence in processed segment SNR.

4.2.2. Influence of Different Noise Type

Table 4-33 to Table 4-36 show the average segment SNR results for different types of noise respectively. Table 4-37 to Table 4-40 show the average segment SNR results for different types of noise with ideal VAD respectively. The segment SNR improvement for different types of noise is illustrated in Fig. 4-37. The graph of average segment SNR difference for ideal/non-ideal VAD is illustrated in Fig. 4-38. Fig. 4-39 illustrates average PESQ result for different types of noise. And Fig. 4-40 illustrates the average PESQ difference between the result using non-ideal (proposed) VAD and the ideal ones. Notice that the y-axis scale for car noise differs from that of the other noise. Different noise types affect the performance of the proposed algorithm and are discussed below:

- a. White noise: the spectral energy component of white noise is widely spread such that it is easier to differentiate the speech energy part from the background noise. The average segment SNR improvement and average PESQ for white noise is relative good as compared to babble noise and factory noise since the

noise estimation in silence region is more accurate than in babble noise and factory noise. The inherent characteristic of white noise is stationary if we expand the scope to a given bandwidth in a given period, for example, one subband frame of the filter bank since the energy is randomly distributed in the whole spectrum. And for the proposed design, the more stationary in one subband the noise is, the more accurate the noise estimation will be. Thus either for ideal or non-ideal VAD, the performance for the proposed algorithm under white noise ranked 2nd within the four noise types.

- b. Babble noise: the spectral energy component of babble noise highly overlapped with that of speech energy thus it becomes hard to distinguish between speech and background noise. Also, for spectral subtraction, the characteristic of the babble noise causes destruction for the original speech, resulting in lower quality improvement. Experimental results show that whether the VAD is ideal or not, the performance for the proposed algorithm under babble noise ranked last among the 4 noise types.
- c. Factory noise: the factory noise in the test database has energy that concentrated mostly in high frequency and low frequency. The characteristic leads to better distinguishability between the speech and background noise in spectral domain. The simulation results show that the performance of the proposed algorithm under factory noise is better than the babble noise due to the inherent property of this type of noise. However the non-stationary characteristic of the factory noise is responsible for the lower performance compared to that of white noise which is discussed before.
- d. Car noise: the car noise stands for the noise recorded inside a riding car which has nearly all the energy lying in the low frequency bands. The energy falls

mostly into the lowest subband (i.e. F22) in the filter bank, leaving other subband free of noise corruption, including the speech energy dominant bands. Experimental results show that the proposed algorithm performs best under car noise environment since the spectral subtraction effectively wipes out unwanted noise in the lowest subbands. The PESQ score improvement is low since the original PESQ of the car noise corrupted speech is high enough, which means the quality of the speech under car noise is close to the clean speech.



Average performance in white noise

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	4.338	4.338	0.954	1.135	0.299
1	5.515	4.515	0.998	1.207	0.301
2	6.592	4.592	1.041	1.296	0.313
3	7.662	4.662	1.092	1.392	0.284
4	8.901	4.901	1.151	1.553	0.291
5	9.701	4.701	1.218	1.711	0.170
6	11.441	5.441	1.283	2.000	0.052
7	11.968	4.968	1.351	2.084	0.022
8	12.401	4.401	1.424	2.156	0.021
9	12.949	3.949	1.508	2.239	0.019
10	13.230	3.230	1.597	2.303	0.018
11	12.803	1.803	1.691	2.328	0.017
12	13.029	1.029	1.781	2.407	0.013

Table 4-33 Average performance in white noise

Average performance in babble noise

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	1.342	1.342	0.941	1.024	0.083
1	2.715	1.715	0.993	1.090	0.097
2	4.479	2.479	1.063	1.258	0.195
3	5.743	2.743	1.129	1.370	0.241
4	6.654	2.654	1.188	1.453	0.265
5	7.570	2.570	1.262	1.569	0.307
6	8.315	2.315	1.337	1.674	0.337
7	9.214	2.214	1.414	1.779	0.365
8	9.987	1.987	1.485	1.830	0.345
9	10.766	1.766	1.583	1.944	0.361
10	11.564	1.564	1.673	2.024	0.351
11	12.010	1.010	1.768	2.093	0.326
12	11.846	-0.154	1.871	2.159	0.288

Table 4-34 Average performance in babble noise

Average performance in factory noise

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	3.246	3.246	0.993	1.145	0.152
1	4.346	3.346	1.043	1.223	0.181
2	5.413	3.413	1.096	1.311	0.215
3	6.739	3.739	1.158	1.463	0.304
4	8.126	4.126	1.221	1.624	0.403
5	8.754	3.754	1.283	1.707	0.424
6	9.851	3.851	1.353	1.829	0.477
7	10.455	3.455	1.416	1.906	0.491
8	11.081	3.081	1.485	1.958	0.473
9	11.722	2.722	1.566	2.067	0.502
10	12.138	2.138	1.654	2.130	0.477
11	12.289	1.289	1.751	2.196	0.445
12	12.398	0.398	1.847	2.285	0.438

Table 4-35 Average performance in factory noise

Average performance in car noise

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	15.769	15.769	2.766	2.798	0.032
1	16.887	15.887	2.835	2.958	0.123
2	17.131	15.131	2.902	2.988	0.086
3	19.808	16.808	2.964	3.151	0.188
4	19.952	15.952	3.020	3.078	0.058
5	22.216	17.216	3.075	3.174	0.099
6	23.198	17.198	3.136	3.207	0.070
7	20.649	13.649	3.199	3.228	0.029
8	21.816	13.816	3.261	3.238	-0.023
9	26.653	17.653	3.320	3.345	0.025
10	27.385	17.385	3.376	3.422	0.046
11	26.372	15.372	3.425	3.363	-0.062
12	30.797	18.797	3.481	3.503	0.021

Table 4-36 Average performance in car noise

Average performance in white noise with ideal VAD

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	7.685	7.685	0.954	1.422	0.468
1	8.833	7.833	0.998	1.572	0.575
2	9.463	7.463	1.041	1.674	0.632
3	10.093	7.093	1.092	1.775	0.683
4	10.675	6.675	1.151	1.877	0.727
5	11.306	6.306	1.218	1.960	0.742
6	11.864	5.864	1.283	2.047	0.764
7	12.445	5.445	1.351	2.125	0.774
8	12.988	4.988	1.424	2.193	0.769
9	13.485	4.485	1.508	2.274	0.766
10	14.016	4.016	1.597	2.344	0.748
11	14.491	3.491	1.691	2.420	0.729
12	14.992	2.992	1.781	2.487	0.706

Table 4-37 Average performance in white noise with ideal VAD

Average performance in babble noise with ideal VAD

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	3.630	3.630	0.941	1.163	0.223
1	4.713	3.713	0.993	1.255	0.262
2	5.567	3.567	1.063	1.350	0.287
3	6.348	3.348	1.129	1.430	0.301
4	7.131	3.131	1.188	1.499	0.311
5	7.896	2.896	1.262	1.597	0.335
6	8.708	2.708	1.337	1.684	0.347
7	9.581	2.581	1.414	1.782	0.368
8	10.464	2.464	1.485	1.857	0.373
9	11.280	2.280	1.583	1.942	0.359
10	12.145	2.145	1.673	2.033	0.360
11	12.980	1.980	1.768	2.123	0.355
12	13.763	1.763	1.871	2.214	0.343

Table 4-38 Average performance in babble noise with ideal VAD

Average performance in factory noise with ideal VAD

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	4.773	4.773	0.993	1.291	0.298
1	5.838	4.838	1.043	1.408	0.365
2	6.617	4.617	1.096	1.471	0.375
3	7.516	4.516	1.158	1.576	0.418
4	8.310	4.310	1.221	1.694	0.473
5	9.172	4.172	1.283	1.768	0.485
6	10.022	4.022	1.353	1.878	0.526
7	10.691	3.691	1.416	1.927	0.511
8	11.474	3.474	1.485	2.026	0.541
9	12.207	3.207	1.566	2.112	0.546
10	12.998	2.998	1.654	2.166	0.512
11	13.911	2.911	1.751	2.254	0.503
12	14.587	2.587	1.847	2.329	0.483

Table 4-39 Average performance in factory noise with ideal VAD

Average performance in car noise with ideal VAD

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	17.590	17.590	2.766	3.023	0.258
1	18.183	17.183	2.835	3.104	0.268
2	18.408	16.408	2.902	3.144	0.242
3	21.254	18.254	2.964	3.110	0.146
4	21.797	17.797	3.020	3.159	0.139
5	23.048	18.048	3.075	3.172	0.097
6	24.506	18.506	3.136	3.172	0.036
7	25.706	18.706	3.199	3.223	0.024
8	26.690	18.690	3.261	3.283	0.022
9	27.708	18.708	3.320	3.341	0.021
10	28.730	18.730	3.376	3.397	0.021
11	29.710	18.710	3.425	3.446	0.021
12	30.796	18.796	3.481	3.501	0.019

Table 4-40 Average performance in car noise with ideal VAD

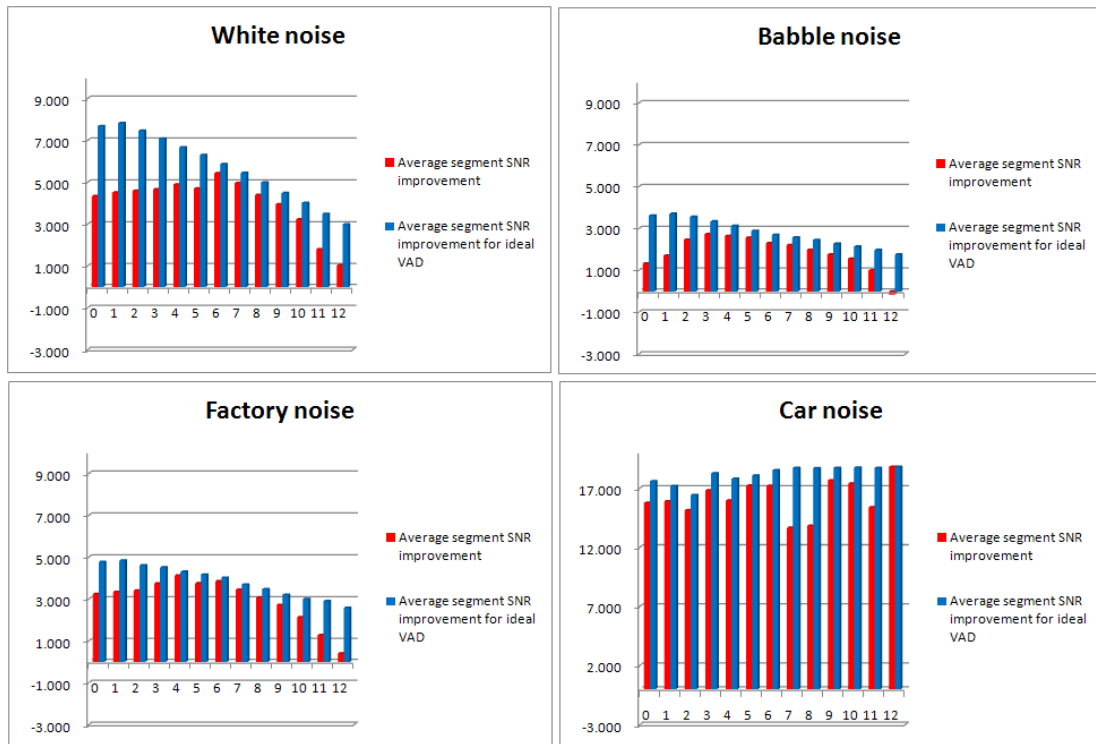


Fig. 4-37 Average segment SNR improvement for ideal/non-ideal VAD

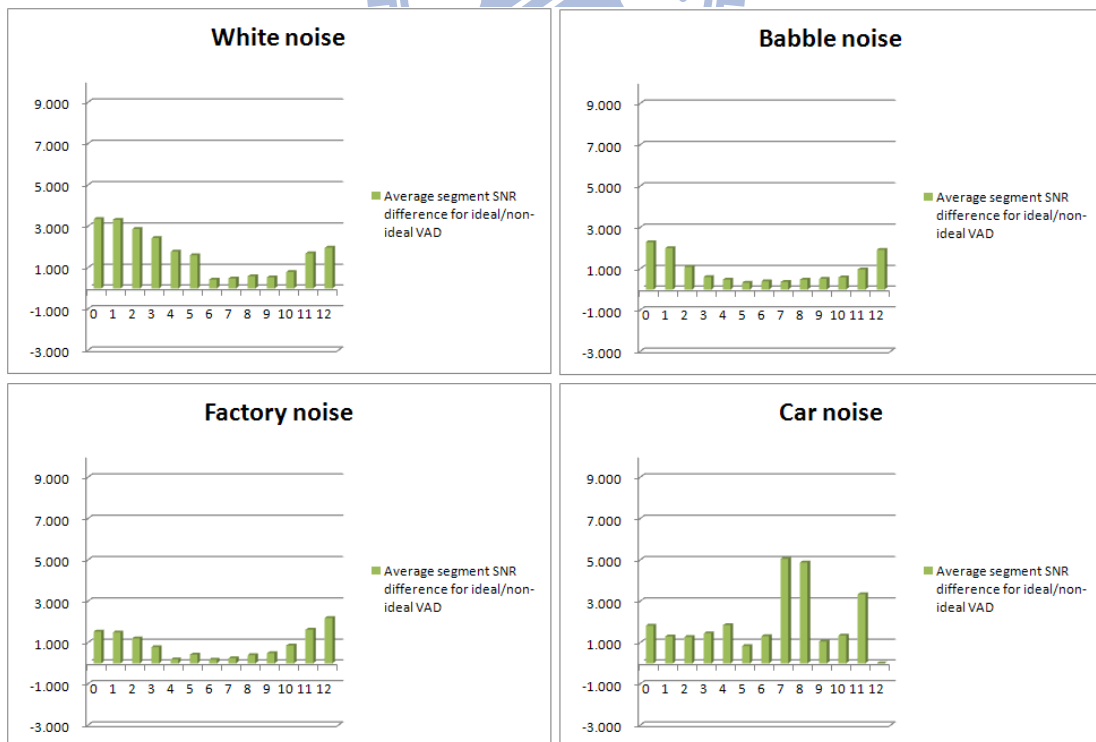


Fig. 4-38 Average segment SNR difference for ideal/non-ideal VAD

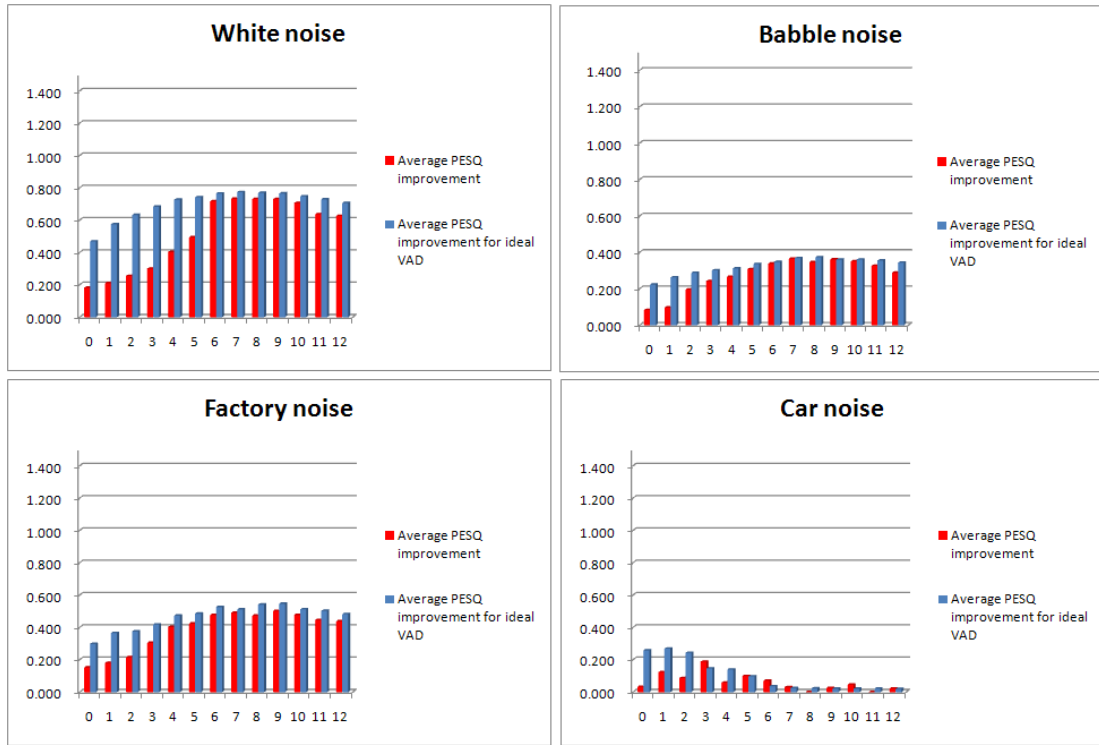


Fig. 4-39 Average PESQ improvement for different noise type

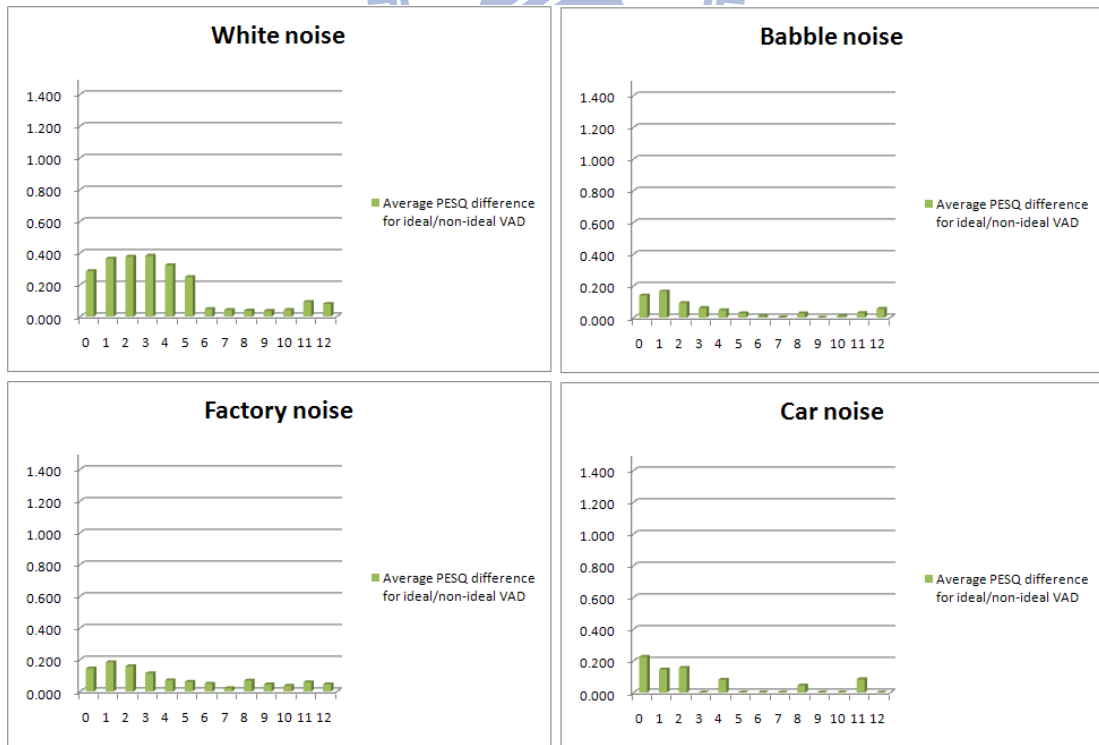


Fig. 4-40 Average PESQ difference for ideal/non-ideal VAD

4.2.3. Influence of Different Silence Period Length before Speech

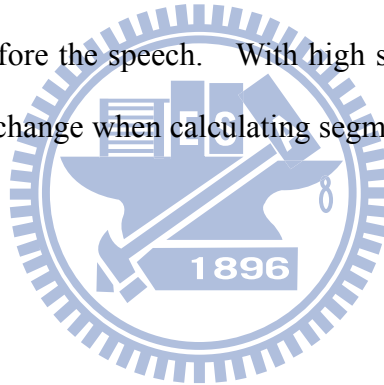
Table 4-41 to Table 4-44 show the average performance for different silence before speech, which are 2 seconds, 4 seconds, 8 seconds, and 16 seconds respectively. Table 4-45 shows the average performance for 16 seconds of silence before speech with ideal VAD.

Fig. 4-41 illustrates the segment SNR results for sequence 1 with different silence period added before the speech. Fig. 4-42 depicts the PESQ scores for sequence 1 with different length of silence period added before the speech. It can be observed that for sequences with silence period of 16 seconds and 8 seconds, the processed segment SNR performs better than the sequences with silence period of 4 seconds and 2 seconds for white noise and factory noise in low original segment SNR. Also, the PESQ score improvement for sequences with silence period of 16 seconds and 8 seconds performs better than the sequences with silence period of 4 seconds and 2 seconds for white noise, babble noise, and factory noise in low original segment SNR. The reason is that the longer the silence before the speech is, the more time allowed for adaptive threshold to move closer to the appropriate position. In that way, VAD accuracy and noise estimation will be enhanced and thus contributes to better performance for the proposed algorithm.

Fig. 4-43 shows the comparison of segment SNR results for sequence 1 with 16 seconds silence period added before the speech for ideal/non-ideal VAD. Fig. 4-44 shows the PESQ scores for sequence 1 with 16 seconds silence period added before the speech for ideal/non-ideal VAD. It can be observed that if we extend the silence

length before the speech, the performance will close to that using ideal VAD. Thus we can conclude that the adaptive threshold requires some “training” time, i.e. 16 seconds to achieve best performance which may contribute to better segment SNR improvement and PESQ score improvement.

The reason that the effect is not obvious for babble noise is that babble noise varies a lot from time to time, causing the adaptive threshold hard to make the appropriate tracking, reducing the performance even with long silence period before the speech. And for car noise, the processed segment SNR reaches over 10dB for low segment SNR even with short silence before the speech. The car noise is stationary such that the improvement is not obvious since adaptive threshold tracks effectively under 4 different silence lengths before the speech. With high segment SNR, a little energy glitch may result in severe change when calculating segment SNR and is shown in the figure.



Average performance for 2 seconds silence before speech

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	5.339	5.339	1.537	1.611	0.074
1	6.865	5.865	1.580	1.728	0.148
2	8.437	6.437	1.626	1.809	0.183
3	9.797	6.797	1.675	1.905	0.230
4	11.100	7.100	1.725	2.004	0.279
5	12.456	7.456	1.775	2.131	0.356
6	13.407	7.407	1.835	2.208	0.373
7	14.422	7.422	1.895	2.274	0.380
8	15.407	7.407	1.955	2.367	0.412
9	16.050	7.050	2.016	2.430	0.414
10	16.259	6.259	2.087	2.486	0.400
11	16.977	5.977	2.161	2.553	0.392
12	17.756	5.756	2.240	2.603	0.362

Table 4-41 Average performance for 2 seconds silence before speech

Average performance for 4 seconds silence before speech

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	5.058	5.058	1.515	1.586	0.070
1	6.083	5.083	1.560	1.661	0.102
2	7.742	5.742	1.601	1.753	0.152
3	9.695	6.695	1.650	1.849	0.199
4	10.912	6.912	1.702	1.950	0.248
5	12.746	7.746	1.750	2.138	0.388
6	13.247	7.247	1.806	2.185	0.379
7	14.492	7.492	1.865	2.249	0.383
8	15.414	7.414	1.929	2.345	0.417
9	15.991	6.991	1.988	2.394	0.406
10	15.658	5.658	2.060	2.484	0.424
11	16.793	5.793	2.137	2.524	0.387
12	17.538	5.538	2.210	2.575	0.365

Table 4-42 Average performance for 4 seconds silence before speech

Average performance for 8 seconds silence before speech

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	6.287	6.287	1.505	1.669	0.164
1	6.773	5.773	1.552	1.731	0.180
2	9.092	7.092	1.585	1.850	0.265
3	10.085	7.085	1.626	1.938	0.313
4	10.575	6.575	1.673	2.016	0.342
5	11.783	6.783	1.724	2.108	0.384
6	13.137	7.137	1.778	2.168	0.390
7	14.248	7.248	1.833	2.217	0.384
8	14.740	6.740	1.894	2.294	0.401
9	15.871	6.871	1.946	2.358	0.412
10	15.820	5.820	2.008	2.412	0.404
11	16.453	5.453	2.082	2.481	0.399
12	16.959	4.959	2.156	2.516	0.360

Table 4-43 Average performance for 8 seconds silence before speech

Average performance for 16 seconds silence before speech

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	6.772	6.772	1.547	1.773	0.226
1	7.473	6.473	1.573	1.802	0.229
2	8.151	6.151	1.608	1.844	0.236
3	9.750	6.750	1.644	1.953	0.309
4	10.857	6.857	1.816	2.039	0.223
5	11.662	6.662	1.705	2.077	0.372
6	13.282	7.282	1.774	2.130	0.356
7	14.140	7.140	1.828	2.205	0.377
8	14.946	6.946	1.888	2.283	0.395
9	15.916	6.916	1.941	2.349	0.408
10	16.245	6.245	1.999	2.397	0.398
11	16.342	5.342	2.063	2.612	0.548
12	16.691	4.691	2.124	2.494	0.371

Table 4-44 Average performance for 16 seconds silence before speech

Average performance for 16 seconds silence before speech with ideal VAD

Original SNRseg	Processed SNRseg (average)	SNRseg improvement (average)	Original PESQ (average)	Processed PESQ (average)	PESQ improvement (average)
0	9.003	9.003	1.547	1.843	0.296
1	9.737	8.737	1.573	1.904	0.331
2	10.451	8.451	1.608	1.969	0.361
3	11.147	8.147	1.644	2.036	0.392
4	11.818	7.818	1.816	2.115	0.299
5	12.735	7.735	1.705	2.147	0.442
6	13.354	7.354	1.774	2.218	0.445
7	14.479	7.479	1.828	2.255	0.427
8	15.337	7.337	1.888	2.330	0.442
9	16.118	7.118	1.941	2.389	0.448
10	16.917	6.917	1.999	2.453	0.454
11	17.798	6.798	2.063	2.526	0.463
12	18.607	6.607	2.124	2.585	0.461

Table 4-45 Average performance for 16 seconds silence before speech with ideal VAD

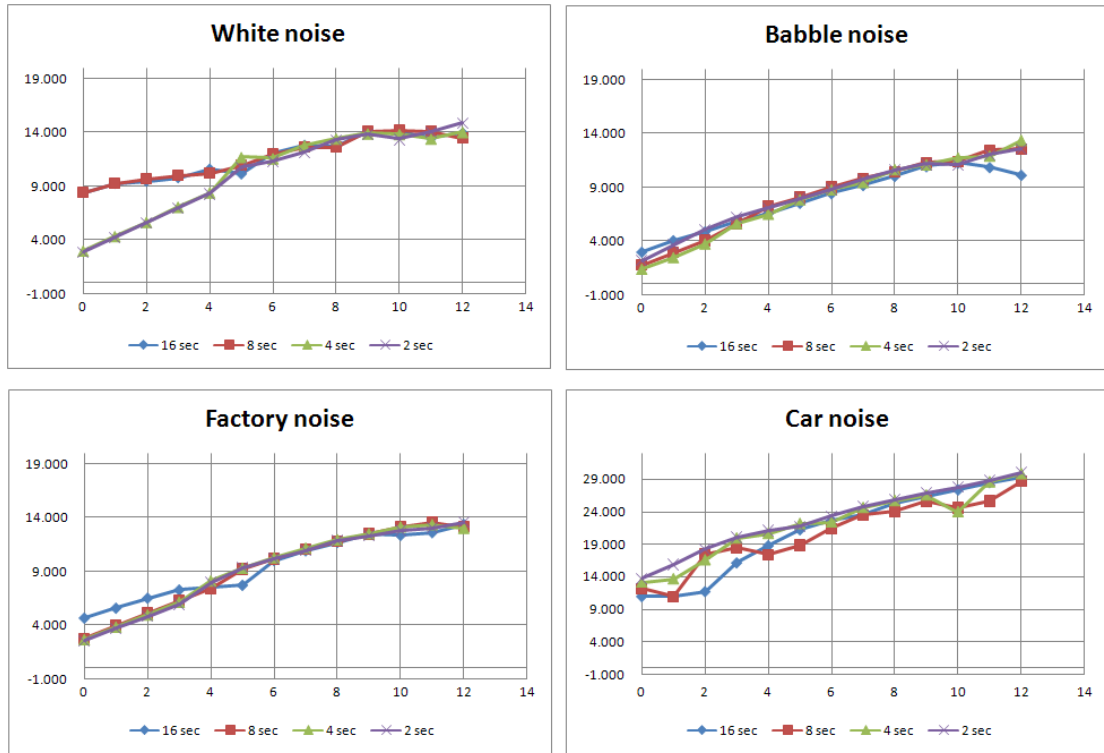


Fig. 4-41 Segment SNR for different silence length before speech

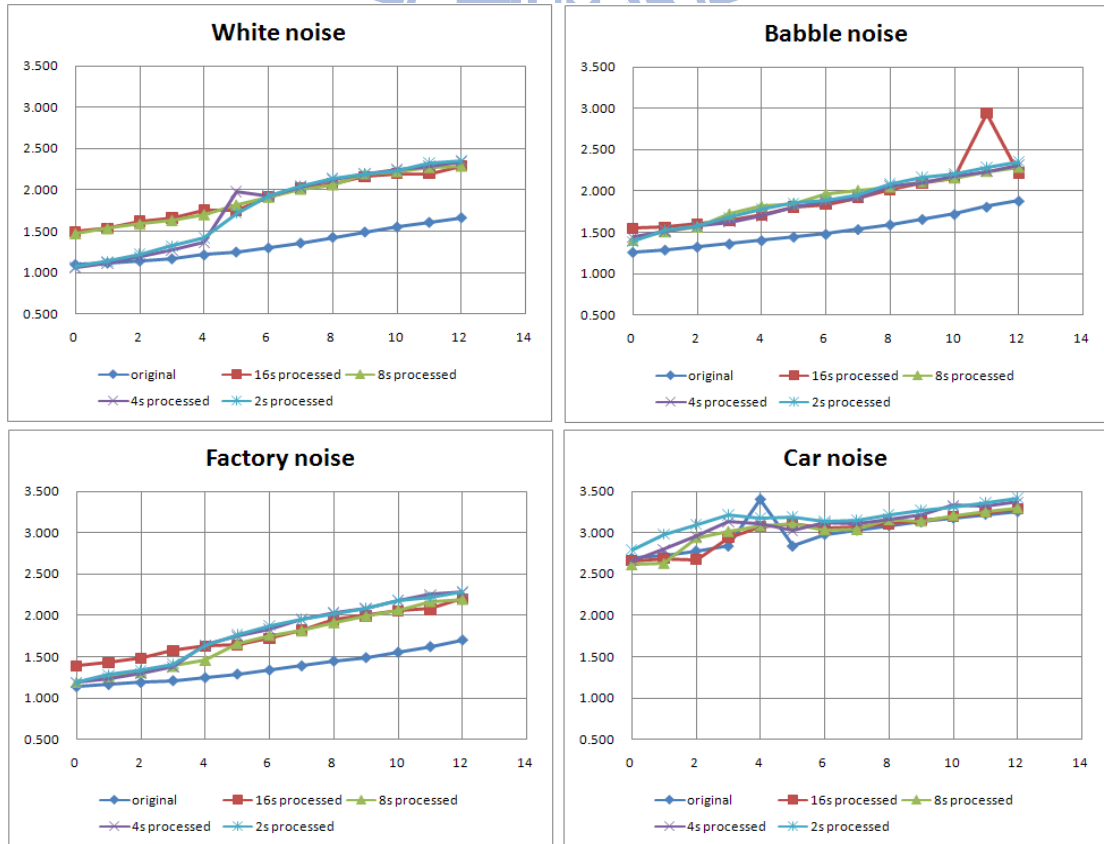


Fig. 4-42 PESQ for different silence length before speech

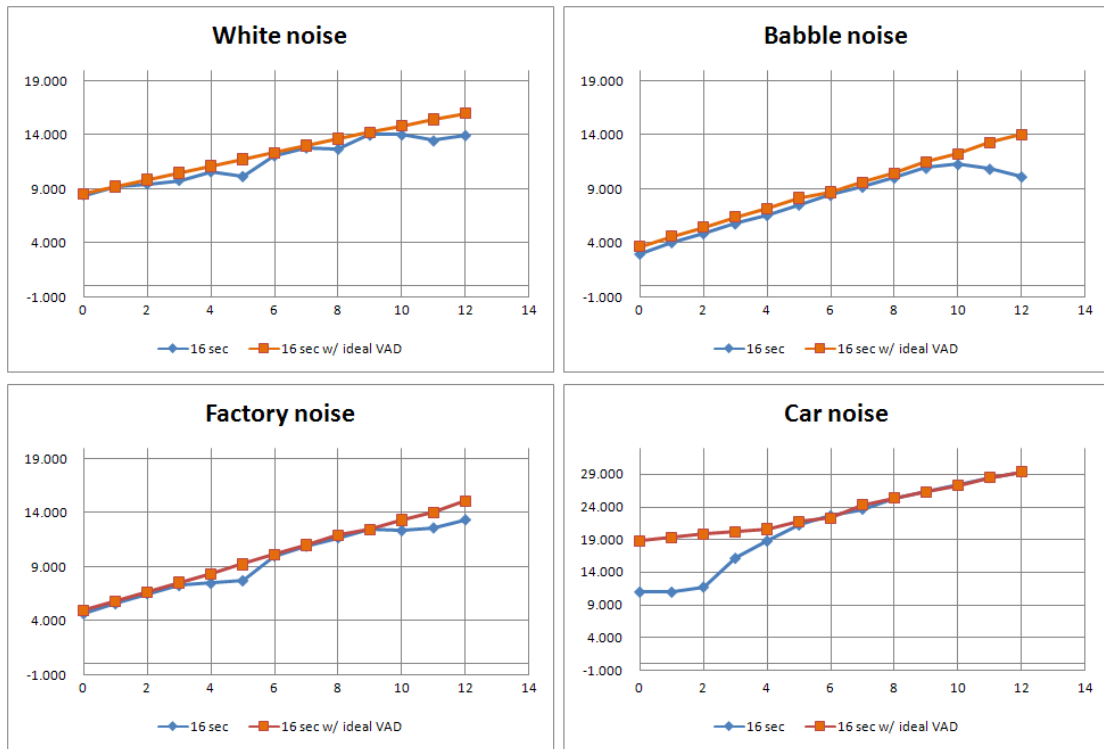


Fig. 4-43 Comparison of segment SNR for ideal/non-ideal VAD with 16 sec silence

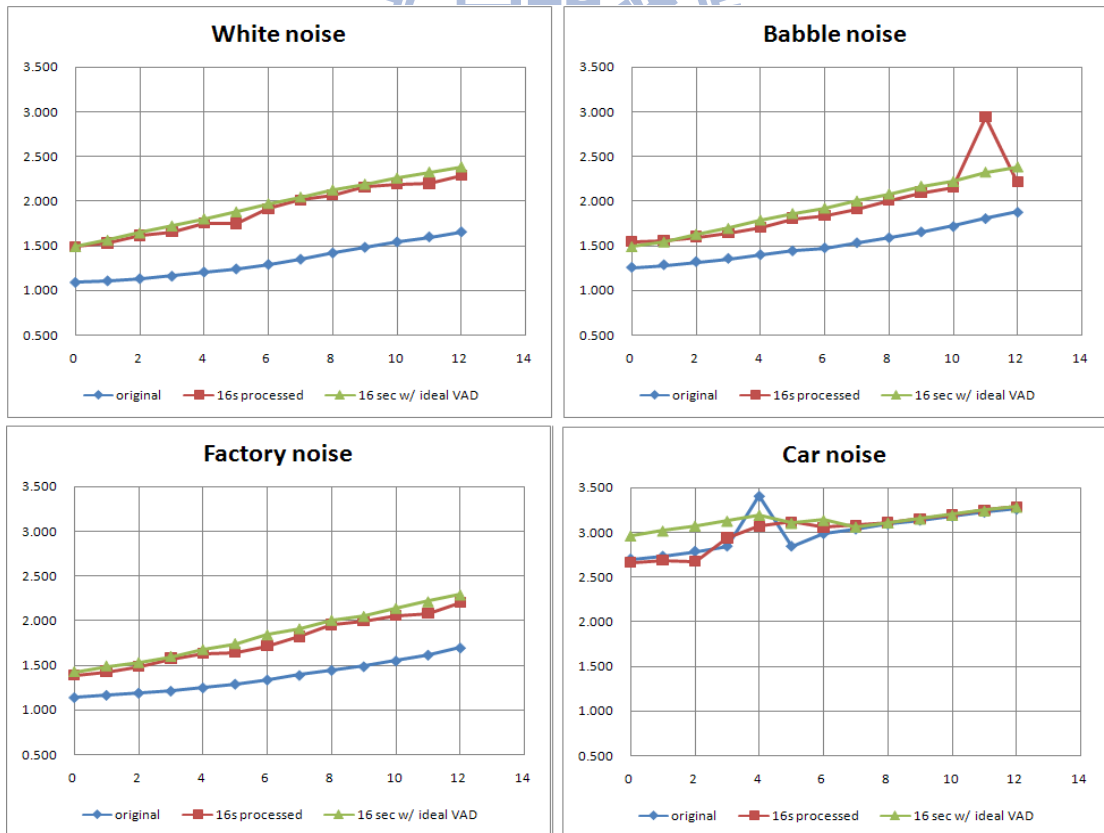
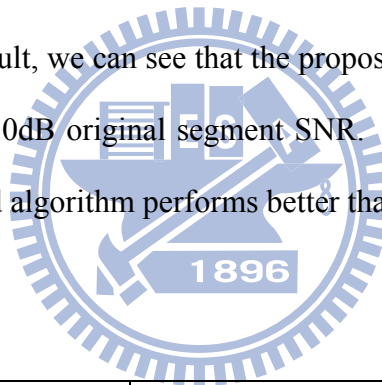


Fig. 4-44 Comparison of PESQ scores for ideal/non-ideal VAD with 16 sec silence

4.3. Comparison with Different Algorithms

Table 4-46 Comparison of segment SNR over different algorithms shows the comparison between the proposed algorithm and other noise reduction algorithms with original segment SNR of 0dB, 5dB, and 10dB. The test sequences of the other algorithms are based on their testing environment which is different from the proposed algorithm. The noise database of the other algorithms is also from Noisex-92. The noise types used for other algorithms are white, babble, factory, and F-16 cockpit. The noise types used for the proposed algorithm are white, babble, factory, and car.

From the comparison result, we can see that the proposed algorithm performs better than A and B and F under 0dB original segment SNR. For 5dB and 10dB original segment SNR, the proposed algorithm performs better than the algorithms A to F.



Original segment SNR	0dB	5dB	10dB
A[20]	1.79dB	7.22dB	10.96dB
B[20]	1.93dB	8.76dB	11.21dB
C[7]	6.60dB	9.23dB	12.65dB
D[16]	6.34dB	9.54dB	11.89dB
E[17]	6.53dB	9.62dB	12.85dB
F[22]	4.80dB	8.63dB	12.66dB
Proposed	5.81dB	11.64dB	15.72dB

Table 4-46 Comparison of segment SNR over different algorithms.

A: Hard thresholding.

B: Soft thresholding.

C: MMSE short time spectral amplitude estimator.

D: Wavelet speech enhancement based on the teager Energy Operator.

E: Speech enhancement using perceptual wavelet packet decomposition and teager energy operator.

F: Explicit-Form Gain Factor for Speech Enhancement Using Spectral-Domain Constrained Approach

4.4. Average Segment SNR Improvement and Average

PESQ Score Improvement

Table 4-47 shows the average segment SNR improvement and average PESQ score improvement for the proposed algorithm. The average segment SNR improvement for 4 sequences and 4 types of noise under 0dB to 12dB original segment SNR is 6.27dB. The average PESQ score improvement is 0.32.

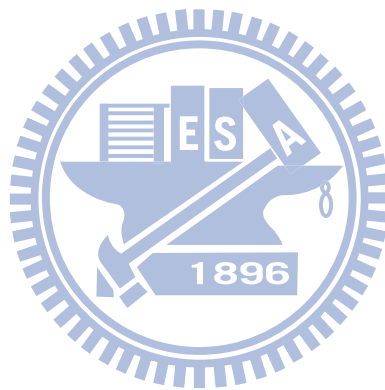
Average segment SNR improvement	Average PESQ score improvement
6.27dB	0.32

Table 4-47 Average segment SNR improvement and average PESQ score improvement

4.5. Summary

The experimental results show the VAD accuracy and the performance of the proposed algorithm for 4 input sequences and under 4 different type of noise which

are white, babble, factory, and car noise. Different noise type results in dissimilar performance of the proposed algorithm due to the inherent characteristic of the energy distribution in the noise. The input sequences that differ in the length of silence period before the speech also cause various performance results.



Chapter 5. Hardware Implementation

5.1. Architecture Design

The proposed algorithm is implemented by ASIC design flow with the hardware architecture depicted in Fig. 5-1. The hardware processing is based on the schedule as depicted in Fig. 5-2. The clock frequency is 6MHz such that the cycle period is 166ns. According to the system specification, the cycle count allowed for calculation is 8,000 while the cycle count needed to complete the processes is less than 8,000 such that no pipelining or parallel processing is required. From the schedule we know that the processes in the proposed design are performed sequentially which means only one stage processes the data at one time. Thus the hardware resource can be reused. The hardware architecture utilizes the “folding” technique which means that few hardware resources are reused again and again to process data that are fed in sequentially.

The conversion of the algorithm to the hardware architecture is described below:

- a. The input sequence data samples are stored into SRAM set 0 and SRAM set 1 with 256x16bit capacity alternately as described in 3.5.3. Thus one sequence can be processed when another sequence is being input simultaneously as depicted in Fig. 5-2. This is also called ping-pong buffer.

For the I/O timing synchronization of SRAM, two sets of registers are added at the input and output ports of the SRAM sets respectively.

- b. The *ABSVAL* block takes the absolute values of input sequence data samples. The data samples are then summed up together by a 32-bit adder *ADD*. The

- window averaging process in Fig. 5-2 is performed by the *shifter*. The averaged data samples are added by the constant K as depicted in 3.5.3.
- c. For logarithm calculation (i.e. Log2 function in Fig. 5-2), the multiplications in (3-9) and (3-14) are replaced by shift-add process with *shifter* and *ADD*. The subtractions are implemented by taking 2's complement number and then perform additions by *ADD*. The table-lookup process as shown in 3.5.3 is carried out by *Combinational logic*.
 - d. The entropy result calculation is done by *ADD*.
 - e. The adaptive thresholding process as shown in Fig. 5-2 is implemented by *Combinational logic* and the adder *ADD* as described in 3.5.4.
 - f. The VAD state decision in 3.6.1. is performed by *Combinational logic*. The noise estimation for *Silence Zone* as illustrated in Fig. 5-2 is implemented by *ADD*, *shifter*, and *Combinational logic* as shown in 3.6.2.
 - g. For spectral attenuation (*Silence Zone and Too-short Voiced Zone*) and spectral subtraction (*Voiced Zone and Voice Protection Zone*) which are depicted in 3.6.3, 3.6.4., and 3.6.5., *ADD* and *shifter* are utilized. The *Combinational logic* is also involved in making decision of the spectral subtraction.
 - h. The off mechanism is implemented by *Combination logic*. When it's activated, the processes after it (spectral attenuation or spectral subtraction) will not be performed.
 - i. For the increment in every counter for the proposed algorithm, the incrementor *INCR* is utilized.
 - j. The output data samples are stored in SRAM set 2 for buffering. The output stage then feed the data to the Insertion Gain in the digital hearing aid system.

From the mapping listed above, the algorithm is implemented by the architecture with simple hardware resource. For deep submicron fabrication, leakage power is an important issue. The proposed architecture is simple and utilizes one 32-bit *ADD* only, thus the circuit area is minimized, namely, reduces the static power consumption. The dynamic power consumption is also minimized since the computation complexity is reduced as discussed previously and all the calculations that need high hardware resource are taken apart or removed.

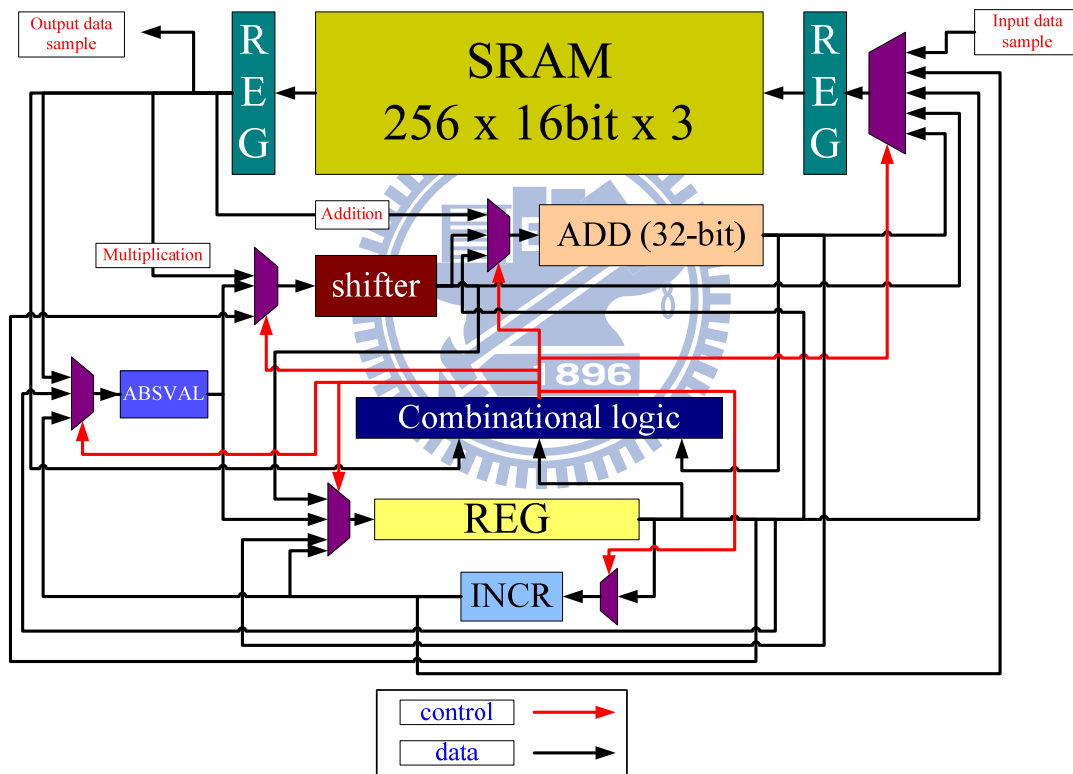


Fig. 5-1 Hardware architecture of the proposed design

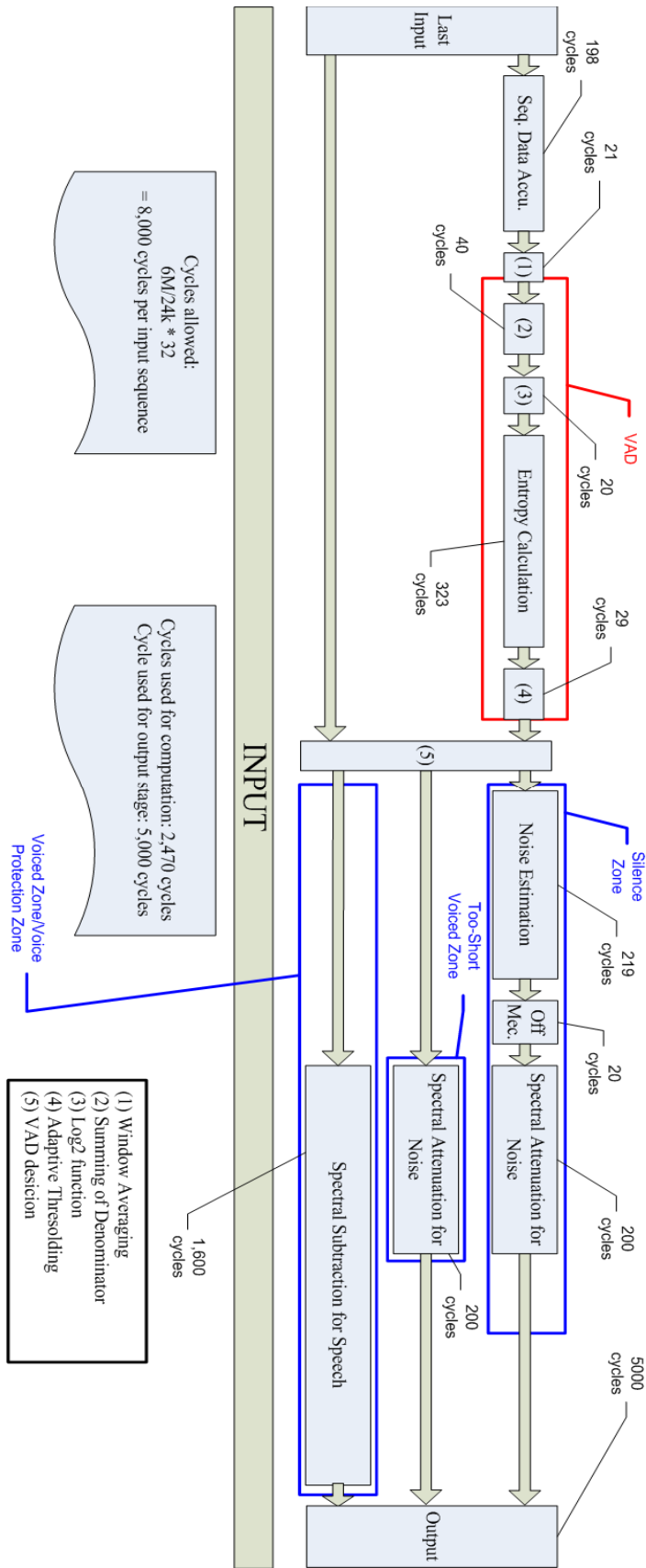


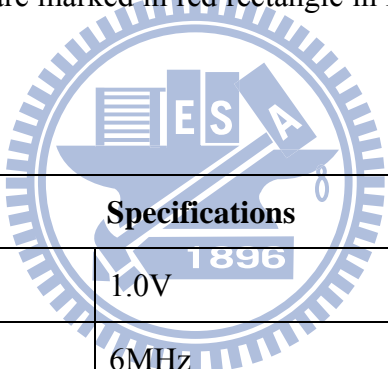
Fig. 5-2 Hardware schedule of the proposed design

5.2. Implementation Result

The hardware design is implemented with Verilog RTL coding. The design is synthesized by Synopsys DesignCompiler™ under UMC 90nm CMOS technology with high V_T cell library. 1.536K Bytes (256x16bit x3) of SRAM are utilized.

The estimated gate-count of the synthesized netlist is 101,697 (including SRAM). If the SRAM is excluded, the estimated gate-count is 80,628. The detailed hardware specifications are listed in Table 5-1.

The proposed design is integrated into the digital hearing aid system. Fig. 5-3 is the layout of the digital system chip. The position of the proposed design and the corresponding SRAM sets are marked in red rectangle in Fig. 5-4. The power report is shown in Table 5-2.



Specifications	
V_{DD} (supply voltage)	1.0V
Clock frequency	6MHz
Technology	UMC 90nm CMOS
Cell library	high V_T
SRAM usage	1.536K Bytes
Gate-count (including SRAM)	101,697 (estimated)
Gate-count (excluding SRAM)	80,628 (estimated)

Table 5-1 Hardware specifications of the proposed design

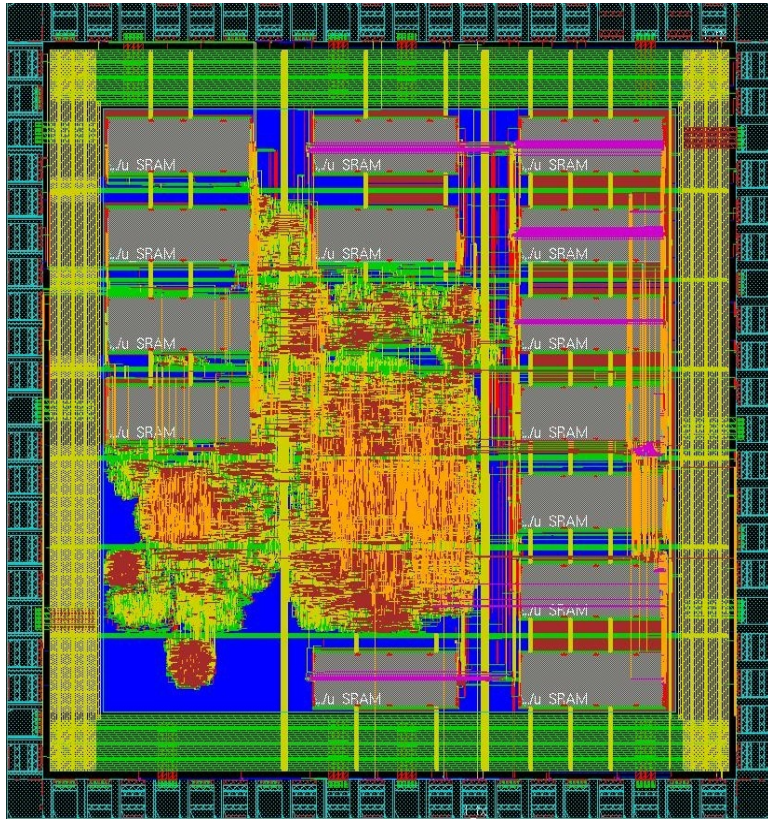


Fig. 5-3 Layout of the digital hearing aid system chip

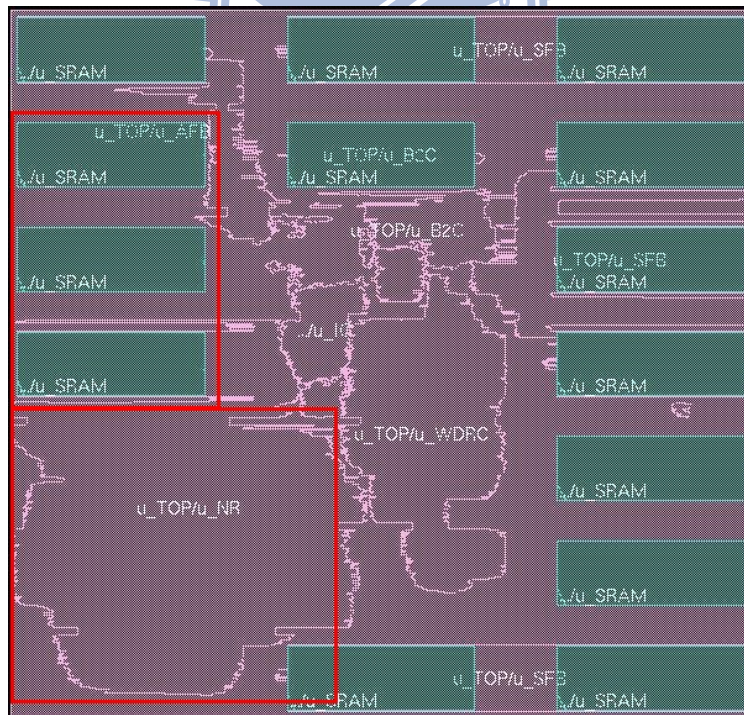


Fig. 5-4 Position marking of each submodules in the chip layout.

	Total Power	Dynamic Power (% of total power)	Leakage Power (% of total power)
Proposed noise reduction design	292.7μW	225.7μW (77.10 %)	67.0μW (22.90 %)

Table 5-2 Power report of the proposed noise reduction design



Chapter 6. Conclusion and Future Work

6.1. Conclusion

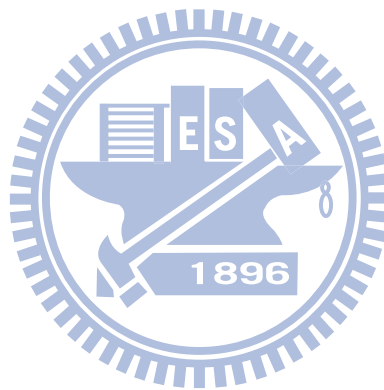
In this thesis, we propose a low power noise reduction design for hearing aids application. The algorithm is composed of entropy-based voice activity detection and filter bank-based spectral subtraction with low power hardware optimization respectively. Simulation results show that the average segment SNR improvement is 6.27dB and the average PESQ score is improved by 0.316. The comparison demonstrates that the processed segment SNR is better than other algorithms under 5dB and 10dB original segment SNR. The hardware design is implemented by UMC 90nm CMOS technology with high V_T cell library. For data storage, 1.536K Bytes of SRAM is utilized. The total estimated gate count is 101,697 including SRAM and 80,628 excluding SRAM. The total power consumption is 292.7 μ W.

In summary, our design can not only enhance the speech quality but also can be hardware implemented with low power consumption. In that way, the proposed design is suitable for low power hearing aid applications.

6.2. Future Work

We have proposed a low power hardware optimized noise reduction algorithm for hearing aids application, while there are still some issues that should be analyzed and improved in future modification. The VAD performance and segment SNR results for babble noise under low original segment SNR is comparatively lower than under other types of noise. A more accurate VAD decision mechanism might be developed

for non-stationary noise such as the babble noise. Subband VAD may also be considered since the precision and power consumption may be further enhanced. The performance of spectral subtraction might be further enhanced by utilizing the filter bank with better resolution and the refined subtraction factor based on the inherent characteristics of the Chinese language. For hardware implementation, CPU-like design may be adopted in order to further save area and power consumption since the ROM code has smaller area occupation than that of logic gates or registers.



Reference

- [1] P. C. Loizou, *Speech Enhancement: Theory and Practice*. Boca Raton, Florida: CRC Press, 2007.
- [2] H. Yi and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 334-341, 2003.
- [3] F. Asano, S. Hayamizu, T. Yamada, and S. Nakamura, "Speech enhancement based on the subspace method," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no.5, pp. 497-507, 2000.
- [4] J.-B. Maj, L. Royackers, J. Wouters, and M. Moonen, "Comparison of adaptive noise reduction algorithms in dual microphone hearing aids," *Speech Communication*, vol. 48, pp. 957-970, 2006.
- [5] N. Sasaoka, K. Shimada, Y. Itoh, and K. Fujii, "Wideband and sinusoidal noise reduction based on adaptive filter with variable step size," in *International Symposium on Communications and Information Technologies*, 2008, pp. 228-231.
- [6] J. V. Berghe and J. Wouters, "An adaptive noise canceller for hearing aids using two nearby microphones," *Journal. Acoust. Soc. Am.*, vol. 103, pp. 3621-3626, 1998.
- [7] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short time spectral amplitude estimator," *IEEE trans. Acoustics, Speech signal processing*, vol. 32, no. 6, pp. 1109-1121, 1984.
- [8] Y. ChangHuai, K. SooNgee, and S. Rahardja, "Adaptive β -order MMSE estimation for speech enhancement," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, pp. I-900-I-903.
- [9] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, pp. 2403-2418, 2001.
- [10] R. Martin, "Speech enhancement using a MMSE short time spectral estimation with Gamma distributed speech priors," in *Int. Conf. Speech Acoust. Signal Process.* vol. I, 2002, pp. 253-256.

- [11] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing.*, vol. 27, pp. 113-120, 1979.
- [12] J. Stegmann and G. Schroder, "Robust voice-activity detection based on the wavelet transform," in *Speech Coding For Telecommunications Proceeding, 1997, 1997 IEEE Workshop on*, 1997, pp. 99-100.
- [13] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, pp. 1-3, 1999.
- [14] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 217-231, 2001.
- [15] F. Jabloun, A. E. Cetin, and E. Erzin, "Teager energy based feature parameters for speech recognition in car noise," *IEEE Signal Processing Letters*, vol. 6, pp. 259-261, 1999.
- [16] M. Bahoura and J. Rount, "Wavelet speech enhancement based on the teager energy operator," *IEEE Signal Processing Letters*, vol. 8, no.1, pp. 10-12, 2001.
- [17] S. H. Chen and J. F. Wang, "Speech enhancement using perceptual wavelet packet decomposition and teager energy operator," *Journal of VLSI Signal Processing Systems*, vol. 2, no. 2-3, pp. 125-139, 2004.
- [18] J.-L. Shen, J.-W. Hung, and L.-S. Lee, "Robust entropy-based endpoint detection for speech recognition in noisy environments," in *International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998, p. 0232.
- [19] M. Berouti, M. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 1979, pp. 208-211.
- [20] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425-455, 1994.
- [21] P. Lockwood and J. Boudy, "Experiments with a non-linear spectral subtractor (NSS), hidden markov models and the projections, for robust recognition in cars," *Speech Communication*, vol. 11(2-3), pp. 215-228, 1992.

- [22] C.-T. Lu and H.-C. Wang, "An explicit-form gain factor for speech enhancement using spectral-domain-constrained approach," *IEICE Trans Inf Syst*, vol. E89-D, pp. 1195-1202, March 1, 2006 2006.
- [23] C.-T. Lu and H.-C. Wang, "Speech enhancement using hybrid gain factor in critical-band-wavelet-packet transform," *Digital Signal Processing*, vol. 17, pp. 172-188, 2007.
- [24] S. Kamath and P. C. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2002.
- [25] L. Singh and S. Sridharan, "Speech enhancement using critical band spectral subtraction," in *Proc. Intern. Conf. Spoken Lang. Process.*, 1998, pp. 2827-2830.
- [26] Y.-T. Kuo, T.-J. Lin, Y.-T. Li, C.-K. Lin, and C.-W. Liu, "Ultra low-power ANSI S1.11 filter bank for digital hearing aids," in *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 2009, pp. 115-116.
- [27] "Specification for octave-band and fractional-octave-band analog and digital filters." vol. ANSI S1.11-2004: Standards Secretariat Acoustical Society of America.
- [28] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.
- [29] C. JIA and B. XU, "An improved entropy-based endpoint detection algorithm," in *International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)*, Taipei, Taiwan, 2002, p. 96.
- [30] J. N. Mitchell Jr., "Computer multiplication and division using binary logarithms," *IRE Trans. Electron. Comput.*, vol. EC-11, pp. 512-517, 1962.
- [31] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ) — a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2001, pp. 749-752.

作者簡歷

姓名：蔡政君

籍貫：台北市

學歷：

國立交通大學電子所系統組 碩士 (民國 96 年 09 月 ~ 民國 98 年 09 月)

國立交通大學電機資訊學士班 學士 (民國 92 年 09 月 ~ 民國 96 年 06 月)

台北市立建國高級中學 (民國 89 年 09 月 ~ 民國 92 年 06 月)

著作：

- [1] Bo-Wen Shih, Cheng-Chun Tsai, and Tian-Sheuan Chang, “Low Power Acoustic Feedback Cancellation for Hearing Aids,” in *VLSI/CAD Symposium*, August 2008.

