Robust Speech Recognition under Noisy Environments

Lee Siu Wa

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Electronic Engineering

July 2004

To my parents.

Acknowledgements

There are lots of people that I must recognize for their help towards completing this thesis. First, I would to thank Professor P. C. Ching, my thesis advisor, for his guidance and support. He has taught me how to find potential research areas to work on. I would also like to express my sincere appreciation to Professor Y. T. Chan for his helpful suggestions. He is knowledgable and enthusiastic about research and teaching. His lessons are always enjoyable. I must mention other academic staffs at the Department of Electronic Engineering, they are Professor W. K. Cham and Professor Tan Lee, who have taught me and inspired my interests in signal processing.

During these two years, I have met several friends and colleagues that have helped me in several ways. Wing Kin Ma, Lai Yin Ngan and Chi Hang Yau are always generous and serious about research. We have countless discussion after office hours with critical suggestions at the end. They have also carefully reviewed my research papers, where they teach me how to deliver complicated concepts with concise explanation. Wai Kit Lo has helped me to solve a programme bug during my final year at undergraduate study. He is always helpful and capable. I am also grateful to Chen Yang, Kin On Luk, Yao Qian, Meng Yuan and Lai Tsang. They have made a warm and comfortable working environment for me and have always been there when needed. I have frequent discussions on speech recognition with Chen Yang. Thanks also go to Wai Zhang and Nengheng Zheng, who have helped me to prepare my first public presentation in Canada. The technical support from Kin On Luk is highly appreciated.

Finally, I want to dedicate this thesis to my parents and Siu Kei Tang, for

their love, support and understanding. They have always encouraged me at most of the time and always been there when I needed them the most. Siu Kei Tang has endured my bad temper over the years and always cheered me up when I feel depressed.

Abstract of thesis entitled:

Robust Speech Recognition under Noisy Environments

submitted by **Lee Siu Wa**

for the degree of **Master of Philosophy**

in **Electronic Engineering**

at **The Chinese University of Hong Kong** in

July 2004.

Automatic speech recognition (ASR) has achieved satisfactory performance in controlled environments, where the average accuracy of a digit string recognition task is about 98%. A controlled environment refers to one that is without additive noise or channel distortion. However, background noise influence and channel distortions often exist in daily applications and most current ASR systems are easily affected with significant degradation in performances. Take an example, when the signal-to-noise ratio (SNR) of input speech is 5 dB, the accuracy decreases to about 40%. This thesis mainly focuses on the additive noise problem towards ASR. Conventional approaches can be classified into three groups, namely speech enhancement, feature compensation and model-based adaptation.

In most standard ASR systems, acoustic models are trained with clean data. The analytical expression for noisy speech features is first derived and it is found that the recognition degradation may due to the mismatch between the training and testing conditions. This implies different acoustical models are necessary for various testing conditions. A simple noise-robust speech recognition system based on noise spectral estimation is proposed. A number of acoustical models are built for distinct SNR conditions. With the SNR estimated, the most relevant acoustical model is selected. This multi-modal approach improves the degree of matching. A modified statistical noise spectral estimation is further proposed for noise spectral estimation, which concentrates on the estimation

accuracy of harmonic frequencies. Experimental results show that the average recognition accuracy of the proposed system is higher than the baseline by 23%.

For fast changing testing conditions (rapidly changing noise characteristics), this multi-modal approach may not be sufficient. Likewise, the recognition accuracy of noisy speech with matched model is still lower than those from clean speech. This is due to the reduced discriminability at low SNR conditions. Hence, there is a need to compensate the noise influence on feature vectors. The noisy speech feature is a non-linear function of the clean speech feature and the complex noise spectrum. By looking at the phase relationship between the speech and noise signal, the noisy speech spectrum can be accurately expressed in terms of the power spectra of the speech and the noise signal. The resultant compensated spectrum is compared with the one from other methods. It is further evaluated on the recognition accuracy. Experimental results indicate that the compensation method is extremely effective under noisy environments. Compared with the widely-used Spectral Subtraction, the proposed method shows superior performance in both known and estimated noise power spectrum conditions. In particular, all sources of recognition error - substitution, deletion and insertion are substantially reduced.

摘要

自動語音識別系統在實驗室環境（無加性噪聲和通道噪聲）下已取得了很大的成功。就數字串識別來講，平均識別率可以達到 98%。然而日常生活中背景噪音和通道噪聲必不可免。目前大部分的自動語音識別系統很容易受到噪聲的影響，造成識別率大幅度下降。例如在信噪比是 5dB 的情況下，識別率會下降到 40%。本文主要側重於研究加性噪聲對自動語音識別系統的影響。傳統的語音抗噪方法可以分爲三類，即語音增強，特徵參數補償和模型自適應。

在目前大部分的自動語音識別系統中，語音聲學模型由無噪語音訓練得到。在本文中，首先我們得到了帶噪語音的分析表達式，發現識別率下降的原因是訓練環境和測試環境的不匹配造成的。這意味著不同的測試環境，應該對應不同的語音聲學模型。因此本文提出了一種簡單的基於噪聲頻譜估計的抗噪語音識別系統，即爲不同的噪聲環境構建了相應的聲學模型，在語音測試時再根據信噪比的估計值，選擇最相近的聲學模型的多模型方法。這種多模型方法改善了訓練環境和測試環境的匹配度。另外，我們還提出了改進的統計噪聲譜估計方法，該方法致力提高於諧波頻率估計的準確性。實驗結果表明，用我們提出的方法識別率比基綫系統提高了 23% 。

但是，多模型的方法不適合應用於測試環境變化很快的情況（例如噪聲變化很快的情況），即使應用匹配模型，識別率仍然達不到無噪情況。其原因是低信噪比使語音的分辨性下降。因此，有必要在特徵參數上作補償。帶噪語音的特徵參數是無噪語音特徵參數和噪聲複頻譜的非綫性函數。通過觀察語音信號和噪聲信號的相位關係，我們發現帶噪語音頻譜可以由語音和噪音的功率譜準確表達。我們比較了用此種方法得到的補償頻譜與用其他方法得到的頻譜。除此之外，我們還用識別率對此方法作了進一步評估。實驗結果顯示此補償方法在噪音環境下非常有效。不論是已知噪聲頻譜和估計噪聲頻譜的情況下，我們提出的方法都遠遠超過被廣泛應用的譜減法。具體來講，誤識率的各種來源錯誤，即替代、刪除及插入錯誤都大大減少了。

Abbreviations

AMDF	average magnitude difference function
ANC	adaptive noise cancellation
AR	auto-regressive
ASR	automatic speech recognition
BSS	blind source separation
CMN	cepstral mean normalization
DCT	discrete cosine transform
DFT	discrete Fourier transform
HMM	hidden Markov model
ICA	independent component analysis
IDCT	inverse discrete cosine transform
IFI	in-phase feature induction
IFT	inverse Fourier transform
LPC	linear predictive coding
MFCC	mel-frequency cepstral coefficient
MSE	mean-square error
M-R T-F QBNE	mainlobe-resilient time-frequency quantile-based noise estimation
pdf	probability density function
PMC	parallel model combination
QBNE	quantile-based noise estimation
SNR	signal-to-noise ratio
SS	spectral subtraction
T-F QBNE	time-frequency quantile-based noise estimation
VAD	voice-activity detection

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As typical Automatic Speech Recognition (ASR) systems have achieved satisfactory performance in controlled environments and electronic devices become physically smaller and smaller, speech technologies have been deployed in domestic applications, such as the hand-free telecommunication. Hand-free telecommunication refers to a communication mode, in which the speakers interact with each other over a communication network, without wearing any tethered devices such as desktop microphones [1].

There exists a number of technical considerations in such a scenario. For example,

- background noise or speech from competitive speakers may be present when received, together with the desired speech signal and may affect the accuracy of ASR systems.
- The speech signal captured is a function of the acoustical conditions, which depends on the types of microphone being used, as well as the transmission channel where reflection and reverberation need to be considered.

This operating condition creates a difficult task for ASR. Recognition performance degrades drastically when speech is corrupted by additive noise and channel distortion caused during transmission. It would be desirable to have a system which is insensitive to these environmental influences. We use the word 'robust' to describe such a system.

In this thesis, we shall mainly focus on the recognition problem induced by additive noise. Figure 1.1 depicts the signal model used for representation of corrupted speech segments. $y(t)$ is the corrupted speech signal received. If no channel distortion is involved, the received speech $y(t)$ is only interfered by the noise $n(t)$ and $h(t)$ is the unit impulse function. The following section gives an overview of automatic speech recognition and talks about some classical methods to alleviate this problem.

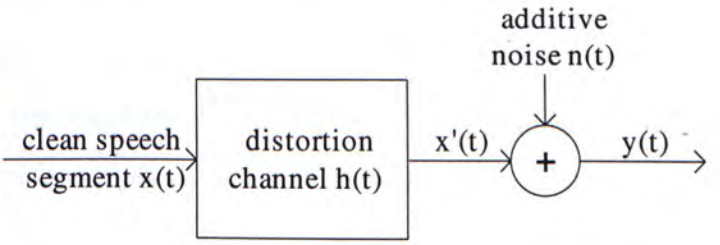


Figure 1.1: The signal model for corrupted speech segments.

1.1 An Overview on Automatic Speech Recognition

Speech recognition is the problem of determining the sequence of words that have been spoken in an utterance. It is essentially a statistical pattern classification that determines a given speech segment into one of speech sound classes [2, 3]. The classification is performed using a sequence of features. A feature is a parametric form of the speech signal. Typical representations include the log filterbank output, mel-frequency cepstral coefficient (MFCC) and linear predictive coding (LPC) coefficients. The speech recognition system first learns the distribution of the features for different classes through a process called training. During recognition (also referred as testing), a speech segment is assigned to the speech sound class whose distribution is most likely to generate the sequence of features.

Let W represent an arbitrary sequence of words. Let $P(X|W)$ denote the

distribution of the speech sound class that associated to the word sequence W, where X now represents an arbitrary signal. The speech recognition problem can be stated as:

$$X_s \rightarrow A \quad \textit{if } P(A)P(X_s|A) > P(B)P(X_s|B) \quad \textit{for all } A \neq B \tag{1.1}$$

where A and B are different instances of W. This can be rewritten as,

$$X_s \rightarrow A \quad : \quad A = \arg\max_w \{P(w)P(X_s|W)\} \tag{1.2}$$

where X_s is the feature of the signal to be recognized. $P(w)$ is the a priori probability of the word sequence W. It may be given by a language model in some cases, which is irrelevant to the robustness problem studied in this thesis.

Most ASR systems exhibit unacceptable degradations in performance when the acoustical environments used for training and testing are not the same. When speech signal $x(t)$ is corrupted by noise $n(t)$, a noisy speech $y(t)$ is generated, one of the consequences is that the distribution of the features of $y(t)$ are no longer similar to the distribution of $x(t)$ that learned from the training data. This mismatch results in degradation in recognition performance. For example, a clean speech connected digit recognition system with accuracy of 99% attains accuracy of only 40% when the signal-to-noise ratio (SNR) decreases to 5 dB.

In recent years much effort has been directed to reducing this mismatch, so as to enhance the recognition performance. Basically, these methods can be classified into three groups, namely, speech enhancement, feature compensation and model-based adaptation [4, 5, 6].

- **speech enhancement**

  Most of the early work towards robustness has been derived from the classical techniques developed in the context of speech enhancement. As a pre-processing step for recognition, speech enhancement techniques are intended to recover the waveform of the clean speech embedded in noise [7]. Normally, the enhanced speech signal is reconstructed at the end.

  One of the most widely studied speech enhancement methods is spectral subtraction [8, 9, 10]. The spectral subtraction method assumes that the

speech and noise are uncorrelated and additive in the time domain. In this case, the noisy speech power spectrum is the sum of the speech and noise power spectra. The method also assumes that the noise characteristics change slowly relative to those of speech signals, so that the noise spectrum estimated during non-speech periods can be used for suppressing the noise.

Spectral subtraction is simple and efficient, but with several problems. For example, the subtraction may result in negative power where these spectral values are set to zero. This non-linear operation produces an annoying distortion called musical noise [11]. Besides, it is found that the performance of a recognition which uses this method for noise reduction varies a lot. The accuracy can be ranged from 11% to 88% [12].

- **feature compensation**

  Feature compensation refers to the transformation of noisy speech features into the corresponding form in a reference environment and recognize it with a system trained in the reference environment. This category is highly similar to the speech enhancement group, where the two categories only differ in the input and output form.

  Several feature compensation methods have been proposed in the literature. One representative is the cepstral mean normalization (CMN) [13, 14]. CMN is designed to handle channel distortion, increasing the robustness of speech recognition systems to unknown linear filtering. This normalization is useful, because different microphones have distinct or even varying transfer functions. The transfer function also depends on the room configuration.

  The principle is that a convolutional distortion in time domain, such as a channel distortion, corresponds to an additive distortion in the cepstral domain. Let $x(t)$ be a speech signal and $h(t)$ be the channel impulse response. $y(t)$ is the speech signal transmitted through the channel. We have the following equivalence,

$$y(t) = x(t) \circledast h(t) \iff c_y(k) = c_x(k) + c_h(k) \tag{1.3}$$

where $\circledast$ is the convolution operation and $c_x(k)$, $c_h(k)$ and $c_y(k)$ are the cepstrum of the speech signal, channel and the transmitted speech signal respectively.

Assuming that the channel characteristics are constant and the expectation of speech cepstrum is zero, taking the expectation on the right hand side of the equivalence gives

$$\begin{aligned} E[c_y(k)] &= E[c_x(k)] + E[c_h(k)] \\ &= E[c_h(k)] = c_h(k) \end{aligned} \tag{1.4}$$

By computing the long time average of the cepstrum of $y(t)$, we have

$$c_h(k) = \frac{1}{N} \sum_{k=1}^{N} c_y(k) \tag{1.5}$$

where N is the total number of segments in the utterance. To remove the channel effect, $c_h(k)$ is simply subtracted from $c_y(k)$.

CMN may be harmful for short utterances. Assume that an utterance contains a single phoneme. The mean $c_y(k)$ will be very similar to the segments in the utterance, since the phoneme is stationary. After normalization, the mean is removed and the normalized $c_y(k)$ will be close to 0. Similar results will apply for other single phoneme utterances. Hence, CMN makes it impossible to distinguish these short utterances and the recognition error rate will be very high.

- **model-based adaptation**

  If the noise characteristics are known ahead of time, it is useful to have training under the expected condition. This method is limited, however, because it it impossible to train under all conditions. Therefore, it would be much more practical to have methods for automatically adapting the acoustic models to the environment. This is model-based adaptation.

  Parallel model combination (PMC) is one of the mature model-based adaptation methods developed recently. The distribution of the speech sound class and of the noise model are trained separately [14, 5]. During

adaptation, the probabilities of the two models are combined to give the probability of the noisy speech segment. At medium to high SNR, PMC gives a significant improvement. Nevertheless, at low SNRs, the compensated models have large variances. These large variances greatly reduce the discriminability between recognition units. In this case, signal enhancement or feature compensation outperform model-based adaptation methods.

To have high discriminability between recognition units, an approach similar to speech enhancement or feature compensation is adopted. Noisy speech features are converted to approximate the clean speech features. Particular attention has been put on the reasons why spectral subtraction cannot give accurate estimation, even if all input parameters are known a priori. By studying the deviation of noisy speech features, an effective spectral compensation method is proposed in this thesis [15]. Experimental results indicate that this compensation method is extremely powerful under noisy environments. Compared with the widely-used spectral subtraction, the proposed method shows superior performance and all sources of recognition error are substantially reduced.

1.2 Thesis Outline

The thesis outline is as follows:

In Chapter 2, the fundamentals of ASR systems and feature representation are given. In particular, the baseline recognition system is described in detail.

Chapter 3 explores the reasons of the recognition degradation in terms of matching between the training and testing conditions. A simple and effective recognition framework is then proposed to bring up the recognition accuracy, which selects the best-matched acoustic model according to the noisy speech characteristic.

Chapter 4 continues the work in previous chapter. We will present a statistical noise estimation method [16] to work with the recognition system proposed in Chapter 3.

In Chapter 5, the degradation problem is analyzed in terms of the deviations of noisy speech features from clean features. By making use of the deviation expression, a spectral compensation method is proposed. We will show the motivation and mathematical principles and conclude with experimental results.

Finally, Chapter 6 summarizes the results and provides a conclusion and discussion. There are also some suggestions for further studies.

Chapter 2

Baseline Speech Recognition System

This chapter gives a detailed description of a baseline speech recognition system from the fundamentals of recognition systems, feature representation to the recognition experiment. The recognition task is a speaker-independent connected digit recognition in the presence of additive background noise and/ or channel distortion.

2.1 Baseline Speech Recognition Framework

Automatic Speech Recognition (ASR) refers to the process of converting input speech signals to word sequences, associating speech to the related concepts or performing tasks as specified. The recognition process finds out the word sequences that best match the acoustic observations according to some models or criteria [17]. Standard ASR framework generally consists of three modules, namely front-end analysis system, back-end decoder and pattern training process. Figure 2.1 shows a baseline speech recognition framework.

Front-end analysis system carries out feature extraction and most of the speech signal processing routines if necessary, such as end-point detection, pitch estimation and noise reduction. Features of certain parametric representation are used for recognition, instead of the input signal waveform, so as to empha-

size the discriminative characteristics in speech, remove irrelevant contents like speaker characteristics or background noise and decrease information rate. Typical examples of representation are short-time energy, linear-predictive coding (LPC), mel-frequency Cepstral Coefficients (MFCC) and reflection coefficients. The generated features, which are called observations, are then input to either the back-end decoder during testing or the pattern training module during training.

The pattern training process generates a reference pattern or a statistical model for each speech unit with input features. A speech unit can be a word, syllable or phoneme. There are basically two recognition approaches, template matching and statistical modelling. While template matching uses reference patterns, the latter one uses statistical models. For simplicity, we assume the template matching approach in the following explanation.

During recognition, the back-end decoder compares the input features against each reference pattern and measures the similarity between them. Distance measurement may be used equivalently. Popular distance measurements include log spectral distance, cepstral distance and Itakura-Saito distortion. Readers may refer to [18] for their details. Based on the similarity (distance) measurement, the reference pattern with highest similarity score (smallest distance measurement) is selected as the recognized output.

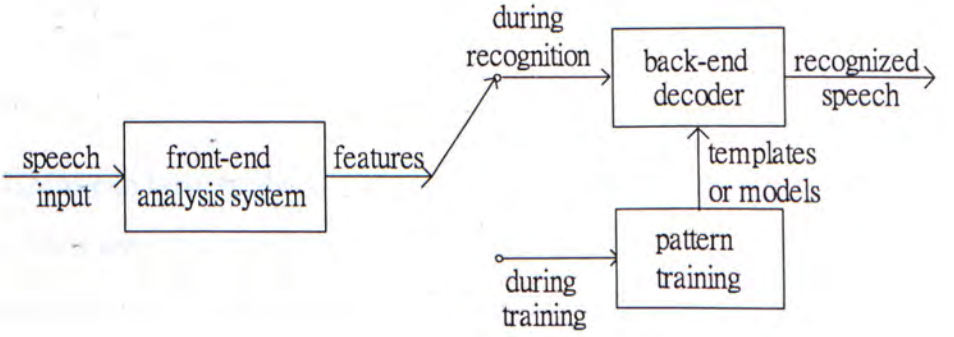


Figure 2.1: A baseline recognition framework.

Template matching has been widely used and the reference pattern for a certain speech unit can be easily obtained by averaging the input features rep-

resenting the same speech unit. Each of these reference patterns acts as the mean of the inputs. During recognition, template matching approach compares the testing features with each reference pattern. It is obvious that only the first-order statistics – mean, is concerned, but other higher-order statistics, such as the covariance, are neglected. However, speech signals have great acoustic variability and the covariance is particularly important for speech signals. Hence, a statistical approach with mean and covariance models are used in our baseline recognition system, which is the well-known hidden Markov model (HMM) approach [19, 20]. HMM is also referred to as a Markov chain. It computes the probability that a certain sequence of speech units is uttered, given the observation sequence.

Hidden Markov model is a parametric representation. In classical HMM-based speech modelling, speech is characterized by two simultaneous random processes in temporal and spectral domains. Figure 2.2 depicts a simple HMM used for speech signals.

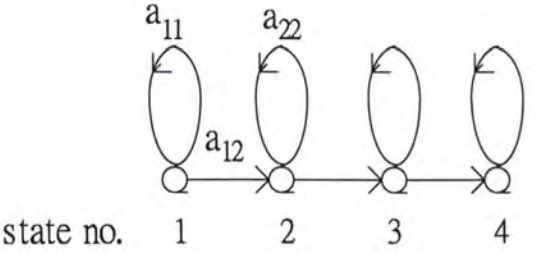


Figure 2.2: A first-order HMM with four states.

A HMM can be described by a set of states, which is denoted by a node. State transition is used to model temporal changes; probability density function (pdf) is assigned to each state to model the spectral variation at a certain frequency. As speech characteristics change over time in a successive manner, a left-to-right topology is used, meaning that only transitions going from the left to the right is allowed. It may be possible to transit from one state to another or remain in the same state, according to a set of probabilities a_{ij} associated with state i and j. Conventionally, it is assumed that current state depends only

on the immediate predecessor state. This is the so called first-order Markov chain. The word 'hidden' is used to describe Markov chains for speech signals, because the state sequence is not directly observable and certain observation can be exhibited by different states.

Speech signal is quasi-stationary so that within a short period of time, its characteristics, for instance, frequency components, periodicity and energy are roughly the same; when it is examined over a long period of time, its characteristics change with different speech units produced. Therefore, all speech analysis are short-time based and this short analysis period is called a frame.

The front-end analysis system is critical to the recognition performance. It delivers features to the back-end decoder to select the best match speech unit. To have accurate recognition results, insensitivity to speaker characteristics or environmental changes and simple computation, the front-end analysis system should be designed in such a way that it facilitates the above requirements. In this thesis, we investigate the robustness of the front-end analysis system for ASR under noisy environments. In the following section, we will talk about the core of the front-end analysis system – feature extraction and how do feature extraction and output features affect the recognition performance.

A HMM-based baseline recognition system is built with the three modules. It is a speaker-independent connected English digit recognizer.

2.2 Acoustic Feature Extraction

Recognition is not performed on the speech signal, rather it works on the basis of the observation vectors, or the so called feature vectors derived from the speech input. These feature vectors should be representative of the speech signal, helpful in distinguishing different speech units and containing any irrelevant information as little as possible [2].

2.2.1 Speech Production and Source-Filter Model

In this section, the two broad classes of speech sounds are addressed first, they are voiced and unvoiced speech. A filter model for speech production will be given. A schematic diagram of the human vocal apparatus is shown in Figure 2.3. Speech sounds can be generally classified into two types, based on the mode of excitation entered into the vocal tract.

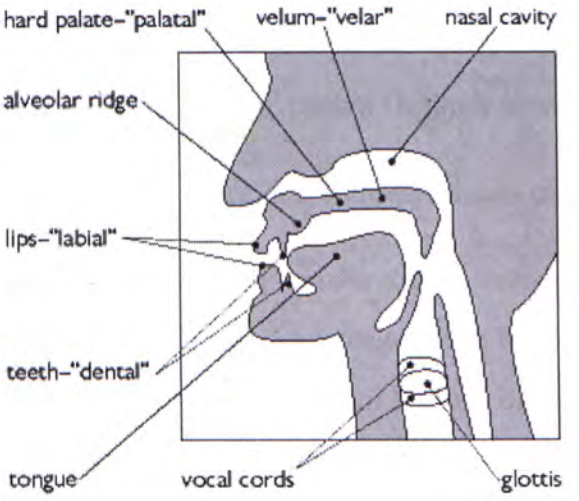


Figure 2.3: A schematic diagram of the human vocal system.

To produce a voiced sound, the vocal cords are tensed and air out of the lungs causes the vibration of the vocal cords and makes the output sound periodic. For unvoiced sounds, the vocal cords are relaxed. The air flow either (1) passes through a constriction in the vocal tract and becomes turbulent, this creates a wideband noise-like excitation or (2) pressure is built behind a point of total closure within the vocal tract and when the closure is opened, the pressure is abruptly released to produce a plosive excitation [18, 21]. Voiced sounds have regular patterns in both waveform and frequency spectrum. The energy of voiced sounds is also much higher than unvoiced sounds. When distinct sounds are generated, shapes of the vocal tract is changed accordingly. Thus, the spectral properties of the output speech vary with time as the shape varies. To model this phenomenon, tubes of non-uniform cross-sectional area with air propagation are often used. The resonance frequencies of the vocal

tract tube are called formant frequencies, or simply formants. Different sounds are produced by altering the vocal tract shapes and equivalently, the formant frequencies, hence, they are important cues for speech recognition. Regarding the speaker characteristics, most of the differences are found in the excitation source generator.

There are three types of excitation sources. They are,

1. quasi-periodic pulse-like excitation from the vocal cord vibration
2. noise-like excitation when the air passes through a constriction
3. transient excitation when there is a sudden release of pressure

With the knowledge of how speech sounds are generated, basic components of speech signals, such as the excitation source and the formant frequencies, can then be modelled. Figure 2.4 shows a commonly used block diagram of speech production, which is referred to as the source-filter model.

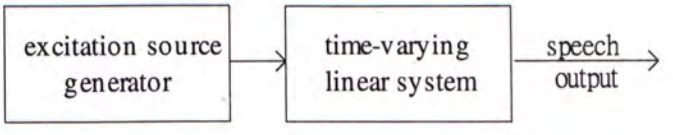


Figure 2.4: A source-filter model for speech production.

The excitation source is separated from the vocal tract. The formants correspond to poles of the filter transfer function and an all-pole filter is one of the popular representations for most speech sounds. To produce the first type of excitation, the excitation source generator outputs a quasi-periodic pulses which are spaced by a pre-defined period; to produce the remaining types of excitation, a random noise waveform is used instead. As a result, the block diagram is modified to the one shown in Figure 2.5. This model has been widely accepted for speech coding, recognition, synthesis and other speech processing for the past decades.

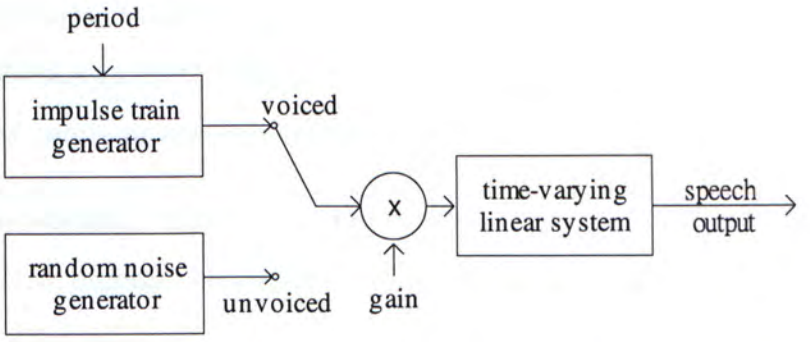


Figure 2.5: The modified source-filter model with voiced or unvoiced excitation.

2.2.2 Review of Feature Representations

Various feature representations are used in speech processing. There are basically two major categories, time and frequency domain features. Examples of time domain features include,

- **pitch** Speech sounds can be split into two basic classes, voiced and unvoiced. The rate of vibration (opening and closing) of the vocal cords during production of voiced sounds is called the fundamental frequency (f0). F0 is closely related to pitch in that pitch is defined as the perception of the rising and falling of tones in speech [14]. Pitch has important roles in many speech applications, such as speech synthesis, recognition of tonal languages and speaker recognition. However, since pitch also represents the voicing characteristics of the speaker, it may not be suitable for speaker-independent ASR.

- **energy** The energy E_n of a speech signal $x(n)$ is defined as,

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \tag{2.1}$$

where $w(n)$ is the framing window defined as,

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & otherwise \end{cases} \tag{2.2}$$

and N is the window length.

Amplitudes of voiced segments are generally larger than the amplitudes of unvoiced segments. Therefore, voiced segments always have high energy values, while unvoiced segments have much lower energies.

- **zero-crossing rate** A zero-crossing occurs if successive speech samples have different algebraic signs. The zero-crossing rate Z_n is calculated by,

$$Z_n = \sum_{m=-\infty}^{\infty} |sgn[x(m)] - sgn[x(m-1)]|w(n-m) \tag{2.3}$$

where

$$sgn[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) < 0 \end{cases} \tag{2.4}$$

and

$$w(n) = \begin{cases} \frac{1}{2N}, & 0 \leq n \leq N-1 \\ 0, & otherwise \end{cases} \tag{2.5}$$

For voiced speech, the energy is concentrated below 3 kHz, due to the speech production mechanism, whereas for unvoiced speech, most of the energy is found at higher frequencies [21]. Since low frequencies imply low zero-crossing rates, high frequencies imply high zero-crossing rates, zero-crossing rate is closed related to energy distribution with frequency. Hence, it can be generalized that if Z_n is high, the speech frame is unvoiced, while if Z_n is low, the speech frame is voiced.

Different speech units can be categorized into voiced or unvoiced nature. By determining the input speech as voiced or unvoiced, this voicing information can be used with standard recognition features to improve the recognition performance. Both energy and zero-crossing rate provide reliable cues for voiced-unvoiced classification. For recognition of noisy speech, however, energy and zero-crossing rate may not be reliable features. A noisy voiced speech may have high zero-crossing rate, because of the noise-like property of corrupted speech and the energy of unvoiced speech may be raised by the noise energy. Extra compensation may be necessary to increase reliable use of them.

- **duration** Conventional HMMs models the temporal structure of speech with exponentially decreasing probability. The probability of t consecutive observations in state i is a_{ii}^t, where a_{ii} is the self-transition probability of state i. This implies that short duration is much more likely to occur than long duration. This implicit modelling is inadequate in that short duration may not be always favorable. Explicit duration modelling is needed, especially for large vocabulary continuous speech recognition, but parameter estimation for duration modelling requires extra heavy computation. For our connected digit recognizer, duration modelling may not be applicable.

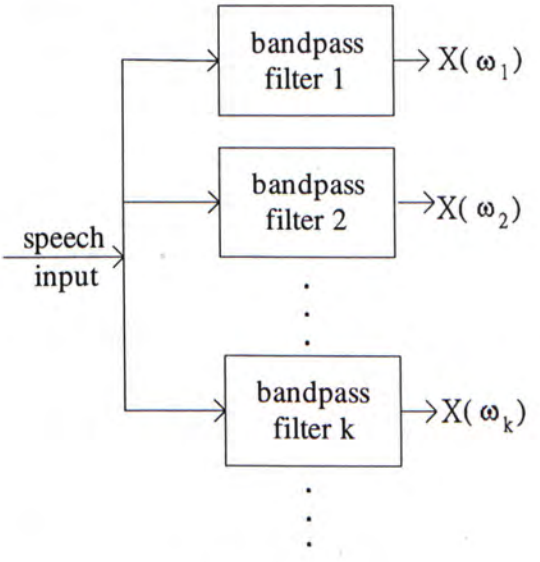
- **dynamic features *delta and delta-delta* ($\triangle$ and $\triangle^2$)** Temporal changes in spectra is useful for ASR [22], in particular, HMM-based ones. These temporal changes captured by first-order and second-order differences record the changes in coefficients over time and provide complementary information for HMM, since HMM assumes each frame is independent of past frames. The first-order and second-order differences are called *delta* $\triangle$ *and delta-delta* $\triangle^2$ coefficients and they are often used in modern ASR systems. These dynamic features also help to alleviate channel distortion in input speech. This will be explained in later sections.

For ASR, features in frequency domain have been the dominant representations. Given that speech sounds are characterized by different formant frequencies, the time-varying linear system representing the spectral envelope is much more important than the excitation source. In general, frequency domain features are much more applicable than time domain features. This is because most of the discriminative features, like formants, are better characterized in the frequency domain. Examples include,

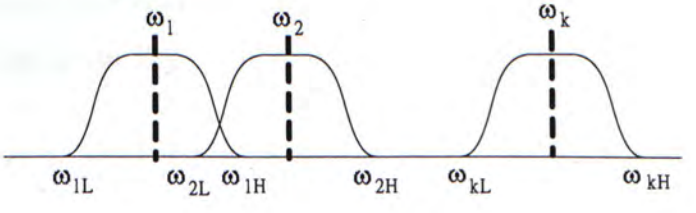
- ◇ **filterbank output** One of the most important structures in the human ear for sound perception is the cochlea, which transmits sound signals to the brain via an auditory nerve [14]. The cochlea acts like a filterbank, whose outputs are ordered by location. High frequency components are

most sensitive in the filters closest to the cochlear base; low frequency components are most sensitive in those closest to its apex. To mimic how human perceives acoustic signals, filterbank analysis has been used for ASR. It is because the output from short-time Fourier analysis is too detailed so that both spectral envelope and excitation source are kept. Performing filterbank analysis smoothes the output and emphasizes the envelope.

Figure 2.6 illustrates a block diagram for filterbank analysis. $X(\omega_k)$ is the output of filterbank k. The speech input passes through a series of bankpass filters linearly spaced in the frequency range under consideration, for example, 300-3400 Hz may be used for telephone speech. The filterbanks are generally overlapped with each other.



(a) Block-diagram



(b) Filterbank

Figure 2.6: Filterbank analysis.

◇ **linear predictive coding (LPC)** Linear predictive coding is another powerful speech analysis method. LPC assumes that a given speech sample at time n, $x(n)$, can be approximated as the linear combination of past p speech samples, such that

$$x(n) \approx a_1 x(n-1) + a_2 x(n-2) + \cdots + a_p x(n-p) \tag{2.6}$$

where $a_1, a_2, \ldots, a_p$ are the LPC coefficients for a speech frame. By adding an excitation term, Equation (2.6) becomes an equality as,

$$\begin{aligned} x(n) &= a_1 x(n-1) + a_2 x(n-2) + \cdots + a_p x(n-p) + Gu(n) \\ &= \sum_{i=1}^{p} a_i x(n-i) + Gu(n) \end{aligned} \tag{2.7}$$

where $u(n)$ is the normalized excitation of unity power and G is the gain of the excitation. LPC is a parametric representation that directly represents the speech samples with the source-filter model (an all-pole filter in most cases) described in Section 2.2.1. The LPC coefficients are found by minimizing the sum of the squared differences between the actual speech samples and the linearly predicted ones over a finite duration.

By Z-transform, we have,

$$X(z) = \sum_{i=1}^{p} a_i z^{-i} X(z) + Gu(n) \tag{2.8}$$

$$H(z) = \frac{X(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \tag{2.9}$$

where $H(z)$ is the transfer function.

If the linear combination of past speech samples is used to approximate $x(n)$ by Equation (2.6), the prediction error $e(n)$ is defined as,

$$e(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i) \tag{2.10}$$

When $x(n)$ is an auto-regressive (AR) process which an all-pole filter can exactly model and the filter order p is correct, the prediction error $e(n)$ will be equal to the excitation source $Gu(n)$. Hence, both the excitation source and the linear filter in the source-filter model can be determined

by linear predictive analysis. Similar to filterbank analysis, the spectral envelope can be found by using the coefficients $a_1, a_2, \ldots, a_p$ or $H(z)$.

◊ **Cepstral analysis** Cepstral analysis is motivated by the need of separating the excitation source and the vocal tract filter. Note that speech signal is the convolution output between an excitation source signal $Gu(n)$ and the filter with impulse response $h(n)$, such that

$$x(n) = Gu(n) * h(n) \tag{2.11}$$

In the frequency domain, convolution becomes multiplication and gives,

$$X(\omega) = GU(\omega)H(\omega) \tag{2.12}$$

By taking the logarithm of the magnitudes of the quantities in Equation (2.12), the multiplication is converted into a sum,

$$\ln|X(\omega)| = \ln|GU(\omega)| + \ln|H(\omega)| \tag{2.13}$$

The cepstrum of a signal $x(n)$ is defined as,

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln|X(\omega)|e^{j\omega}d\omega \tag{2.14}$$

The block-diagram of cepstral analsysis is shown in Figure 2.7. Since $H(\omega)$ models the spectral envelope and $GU(\omega)$ contains the high-frequency excitation source, low-order $c(n)$ and high-order $c(n)$ implicitly represent the spectral envelope and the excitation source respectively.

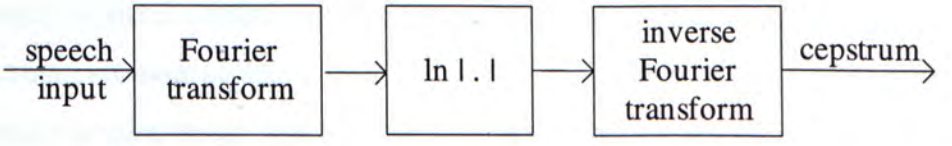


Figure 2.7: The cepstral analysis.

As the output spectra from filterbank analysis and LPC are always highly correlated with adjacent filterbanks or frequency bins, if diagonal covariances are needed in the HMM-based recognizer, a cepstral transformation is necessary.

◊ **perceptually-motivated representation** The filterbanks described before are uniformly spaced in the frequency domain. This is the simplest type of filterbank. Alternatively, non-uniform filterbanks are commonly used, because the human ear is a constant-Q system that resolves frequencies linearly in the logarithmic frequency scale. It is believed that having a feature representation that operates in a similar non-linear manner helps the recognition performance. Typical examples of non-uniform filterbank types are Bark frequency scale and mel scale. The perceptual resolution in both scales are finer in the lower frequencies and coarser in the higher frequencies. By applying one of these frequency scale in the spectral analysis like filterbank output or LPC, perceptually-motivated representation are formed. Among these representations, mel-frequency cepstral coefficients (MFCC) and linear predictive cepstral coefficients (LPCC) are two popular candidates for most ASR systems.

◊ **pitch** Pitch is not only a time domain feature, but also a frequency domain characteristic. In a narrowband spectrogram, the spectral harmonics corresponding to the pitch during voiced segments are resolved and appear as horizontal lines in the spectrogram.

2.2.3 Mel-frequency Cepstral Coefficients

Several feature representations have been introduced in Section 2.2.2. The mel-frequency cepstral coefficients (MFCC) was found to have superior performance over other representations [23]. This may be attributed to the fact that MFCC captures the non-linear property of human perception and separates the vocal tract filter from the excitation source by the cepstral analysis.

Representations derived from the Fourier spectrum, such as MFCC and the log filterbank output well preserves information in most phonemes, but parameters from the LPC spectrum are inaccurate for consonants. This is the consequence of the all-pole filter used in LPC and LPC is less effective for unvoiced segments.

Another merit of MFCC is its compact representation. Normally, 6 to 12 coefficients [17] are sufficient to capture relevant information for ASR. Higher cepstrum coefficients contain mainly for the speaker characteristics. MFCC has been adopted in our baseline ASR system and the following describes the detailed procedure of the feature extraction.

Let $x(t)$ be the speech signal. The MFCC extraction process converts it into a sequence of feature vectors $c(k)$.

MFCC extraction procedure

1. **cutting into frames**

   $x(t)$ is cut into frames. A frame is a short-time analysis period, such that speech characteristics are assumed to be stationary over this duration. The time separation between successive frames is called frame shift. The frame size is normally 20 - 30 msec. In addition, frames are often overlapped to preserve smooth transitions at frame boundaries, so frame size is always larger than frame shift, as illustrated in Figure 2.8. The following steps process the frames.

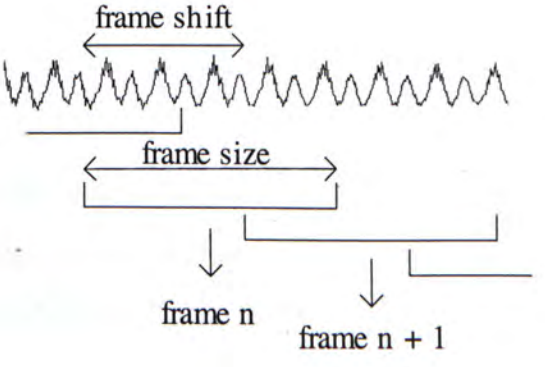


Figure 2.8: The speech signal is first cut into frames.

2. **pre-emphasis**

   It is a common practice to pre-emphasize the speech signal. In the past, the dynamic range of speech spectrum was large due to the lower energies

at high frequencies. Most of the speech coding hardware had insufficient wordlength to represent it. By using a first-order difference equation,

$$\dot{x}(n) = x(n) - \textit{pre-emcoef}\, x(n-1) \tag{2.15}$$

where the pre-emphasis coefficient *pre-emcoef* is equal to 0.97, the high frequency components is amplified, similar to having a high-pass filtering. The dynamic range of the speech spectrum is reduced.

3. **windowing**

   To avoid discontinuity at frame boundaries, the pre-emphasized signal $\dot{x}(n)$ is always tapered with a window function. Windowing is the operation of multiplying a signal by a finite duration function $w(n)$. That is,

$$\ddot{x}(n) = \dot{x}(n)w(n) \tag{2.16}$$

   Popular window functions include the Hamming and Hanning windows. In fact, windowing is always there, since the speech signal lasts only over a finite time interval and rectangular window is applied implicitly. In our baseline system, the Hamming window is used, which is defined as,

$$w(n) = 0.54 - 0.46\cos(2\pi\frac{n}{N-1}), \quad 0 \leq n \leq N-1 \tag{2.17}$$

   where N is the window length and is equivalent to the frame size. Comparing the Hamming window with the rectangular window, the sidelobes of the latter are always high and leakage between adjacent harmonics occurs. This introduces ripples in the spectrum, leading to unclear spectrum. Although the mainlobe of Hamming window is larger, a larger value of N can be used to increase the frequency resolution.

4. **magnitude spectrum and mel filterbanks**

   After windowing, the signal $\ddot{x}(t)$ is then Fourier transformed and the magnitude of each frequency bin is taken.

   The non-linear mel frequency scale is defined by

$$mel(f) = 2595\log_{10}(1 + \frac{f}{700}) \tag{2.18}$$

As shown in Figure 2.9, mel filterbanks are equally spaced along the mel frequency scale. The higher the center frequency, the wider is the bandwidth. The magnitude spectrum is then binned by correlating it with each mel filterbank. Binning means that for a given mel filterbank, each coefficient in the magnitude spectrum is multiplied by the corresponding filter gain and the products are summed. Hence, each filterbank output is a weighted sum representing the spectral magnitude in that mel filterbank.

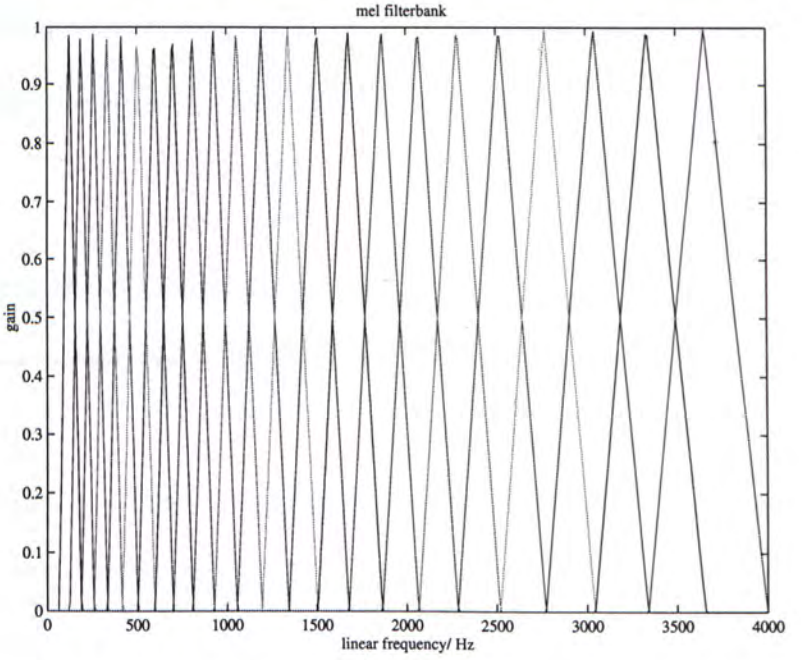


Figure 2.9: Mel filterbank with sampling frequency 8 kHz.

It is possible that the mel filterbanks cover the whole frequency range from dc to Nyquist frequency. Nevertheless, to remove undesired frequencies that may contain noise only, the frequency range is often band-limited.

5. **cepstral analysis**

Let *fbank(m)* be the log filterbank output of bank m. By applying the inverse Fourier Transform (IFT) on *fbank(m)*, that is,

$$c'(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} fbank(m) e^{jm}\, dm \tag{2.19}$$

the cepstral coefficients are computed. Since the log filterbank output is an even function, the discrete cosine function (DCT) can be used to

replace the inverse Fourier Transform.

6. **cepstral liftering**

   The principal advantage of cepstral coefficients is that $c'(k)$ is generally decorrelated and this allows diagonal covariances to be used in the HMMs. However, one minor problem is that the higher order cepstra are numerically quite small and this results in a very wide range of variances when going from the low to high cepstral coefficients [24]. Cepstral liftering is further used to re-scale $c'(k)$ to have similar magnitudes.

   Finally, the cepstral coefficient $c(k)$ is calculated by,

$$c(k) = (1 + \frac{L}{2} \sin \frac{\pi k}{L}) c'(k) \tag{2.20}$$

   where L denotes the liftering parameter.

The complete MFCC extraction process is summarized in Figure 2.10.

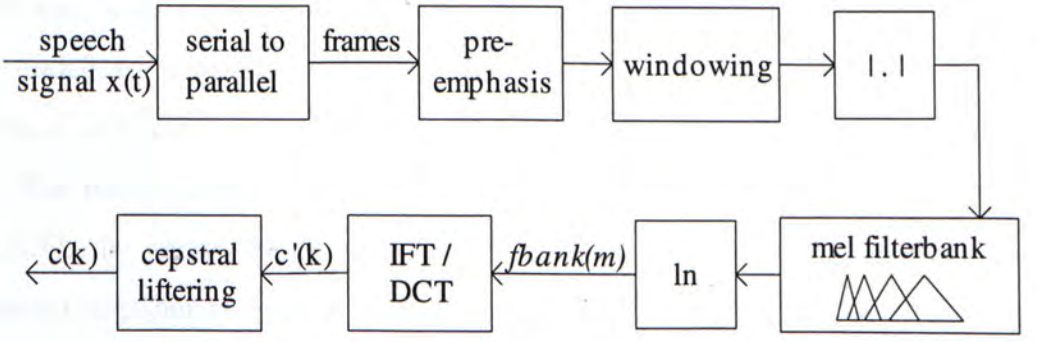


Figure 2.10: Block diagram of the MFCC extraction process used.

2.2.4 Energy and Dynamic Features

The performance of a speech recognition system can be greatly enhanced by augmenting an energy term E and time derivatives to the basic static MFCC parameters.

The energy is computed as the log energy of the speech signal. We have, for samples $x(n), n = 0, 1, \ldots, N-1$ in a certain frame,

$$E = \log \sum_{n=0}^{N-1} x(n)^2 \tag{2.21}$$

Both the first and second order time derivatives are used in our baseline system. Time derivatives help to reduce the effect of channel distortion on the feature parameters. If the channel distortion is stationary or changes slowly, and since the time derivative of a constant is zero, so time derivative is insensitive to channel effects and suffers no distortion from the channel. The first order derivatives (referred to as delta coefficients) are computed using the following regression formula,

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{2.22}$$

where d_t is a delta coefficient at time t calculated in terms of the corresponding static coefficients $c_{t-\theta}$ to $c_{t+\theta}$. Θ denotes the delta window. The same equation is applied to the delta coefficients to obtain the second order derivatives (referred to as acceleration coefficients), but the Θs for delta and acceleration can be different. For the beginning and the end of the speech, some $c_{t-\theta}$ or $c_{t+\theta}$ may be undefined, and the first or the last c(n) is used to replicate any undefined term if necessary.

The feature representation used for our baseline ASR consists of the static MFCC, the energy term, delta and acceleration coefficients. They are augmented together to form a feature vector. In Figure 2.11, an example of the final feature vector is shown.

c(1)	c(2)	...	c(12)	c(0)	E	dc(1)	dc(2)	...	dc(0)	dE	ac(1)	ac(2)	...	ac(0)	aE

c(k) is the static cepstral coefficient
E is the energy term
dc(k) and dE are the delta coefficients and
ac(k) and aE are the acceleration coefficients

Figure 2.11: Example of the feature vector with 12 cepstral coefficients.

2.3 Back-end Decoder

Parameters of training and back-end decoder, such as the number of states per HMM model, the number of mixture components in each state and the number of cepstral coefficients etc., are chosen to follow common settings. The recognition of digit strings is considered as a task without restricting the string length.

Whole word HMM models are used for the digits and every word has 16 states with two dummy states at the beginning and end. State skipping is not allowed and simple left-to-right topology is adopted. We use three Gaussian mixtures to model each state. As the cepstral coefficients are assumed to be uncorrelated, diagonal covariance matrices are used in all HMM models.

In addition to whole word models, there are two pause models used, which are the same as defined in [25]. They are 'sil' and 'sp'. 'sil' has a transition structure with three states as shown in Figure 2.12. The number of Gaussian mixtures in each state is six. It is used to model the pauses before and after the utterance.

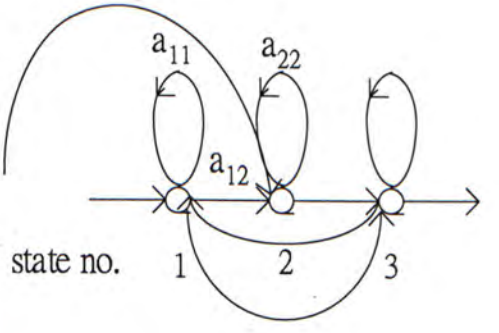


Figure 2.12: A 3-state 'sil' pause model.

'sp' consists of a single state which is tied with the middle state of the 'sil' model. It is used to model pauses between words.

During recognition, an utterance can be modelled by any digit sequence with 'sil' models at the beginning and at the end and with optional 'sp' models between two digits.

Table 2.1 summarizes the parameter values used in our baseline system.

related ASR module	parameter	value
feature extraction	fs	8 kHz
	frame size	25 msec
	frame rate	10 msec
	pre-emcoef	0.97
	hamming window	not applicable
	no. of FFT bins	256
	no. of mel filterbanks	23
	lowest frequency in mel filterbanks	64 Hz
	highest frequency in mel filterbanks	4 kHz
	no. of cepstral coefficients	12
	L	22
	delta window	2
	acceleration window	2
training and decoding	no. of state/ word	16
	topology	left-to-right
	no. of Gaussian mixtures/ word model state	3
	covariance matrix nature	diagonal
	no. of states/ 'sil' model	3
	no. of Gaussian mixtures/ 'sil' model state	6
	no. of state/ 'sp' model	1

Table 2.1: The parameter values used in the baseline recognition system.

2.4 English Digit String Corpus – AURORA2

To have fair performance comparisons between various algorithms, definitions of training and testing scenarios are necessary. A speech database called AURORA2 is used for all the recognition experiments. It was released by the Evaluation and Language resources Distribution Agency (ELDA) in 2000 [25].

The AURORA2 database is designed to evaluate the performance of speech recognition algorithms in noisy conditions. It is exceptionally suitable for the evaluation of front-end feature extraction processes, by using the pre-defined HMM-based recognition back-end.

The recognition task is a speaker-independent connected digits recognition in the presence of additive background noise and/ or convolutional distortion. Both noise and channel distortions are artificially added to the clean TIDigits database [26]. TIDigits consists of connected English digits spoken by American talkers. The speakers are male and female US-American adults speaking isolated digits and digit strings of up to seven digits. The speech samples are downsampled from 20 kHz to 8 kHz by using a low-pass filter with passband between dc to 4 kHz.

To simulate the frequency responses of several mobile terminals, additional filtering is applied. Two frequency responses G.712 and MIRS are defined [25, 27] and the clean speech samples is convolved with either one filter. Both G.712 and MIRS are bandpass filter with passbands from 300 to 3400 Hz. The major difference between the two frequency responses is that the passband of G.712 is very flat, whilst MIRS shows a rising amplitude response from low to high frequencies.

Regarding the noise corruption, eight different noise types are selected and the noise is recorded in real conditions. It is added to the clean speech over a wide range of signal-to-noise ratio (SNR): 20 dB, 15 dB, 10 dB, ..., -5 dB with a 5 dB step. A noise segment with the same length as the clean speech signal is randomly extracted from the long recording. The noise samples are collected in,

- subway (by travelling in suburban trains)
- crowds of people (the so-called babble noise)
- cars
- an exhibition hall
- restaurants
- streets
- an airport
- train stations

The long-term spectra of all noises [25] are shown in Figure 2.13. Most of the energies of the eight noise types concentrate in the low frequency region. Some noises are quite similar, such as those from an airport and train-stations, even though they are recorded from under different environments. Some noise types are fairly stationary, like the car noise and the one captured in the exhibition hall. Other noise types are non-stationary, such as those recorded on the street or at an airport.

To study the performance of front-end algorithms, the baseline training uses clean data[1]. Thus, there is no any distortion or noise in the resultant acoustical models. The models well preserve the high discriminability between clean speech units and when the testing inputs are clean data, the recognition is obtained with the highest accuracy. Hence, it is believed that given the clean training models, the cleaner the testing input, the better the recognition is.

The training data set consists of 8440 clean utterances spoken by 55 male and 55 female adults. These raw data are filtered with the G.712 frequency response.

[1]There is another training mode defined in [25] called multi-condition training. Multi-condition training refers to the case that both clean and noisy data are used for training and the distortion by noise contributes in the resultant acoustic models. The noisy data used for training are those corrupted by subway, babble, car and exhibition noise, that is, the same noises as in test set A. This leads to a highly matched condition of training and testing.

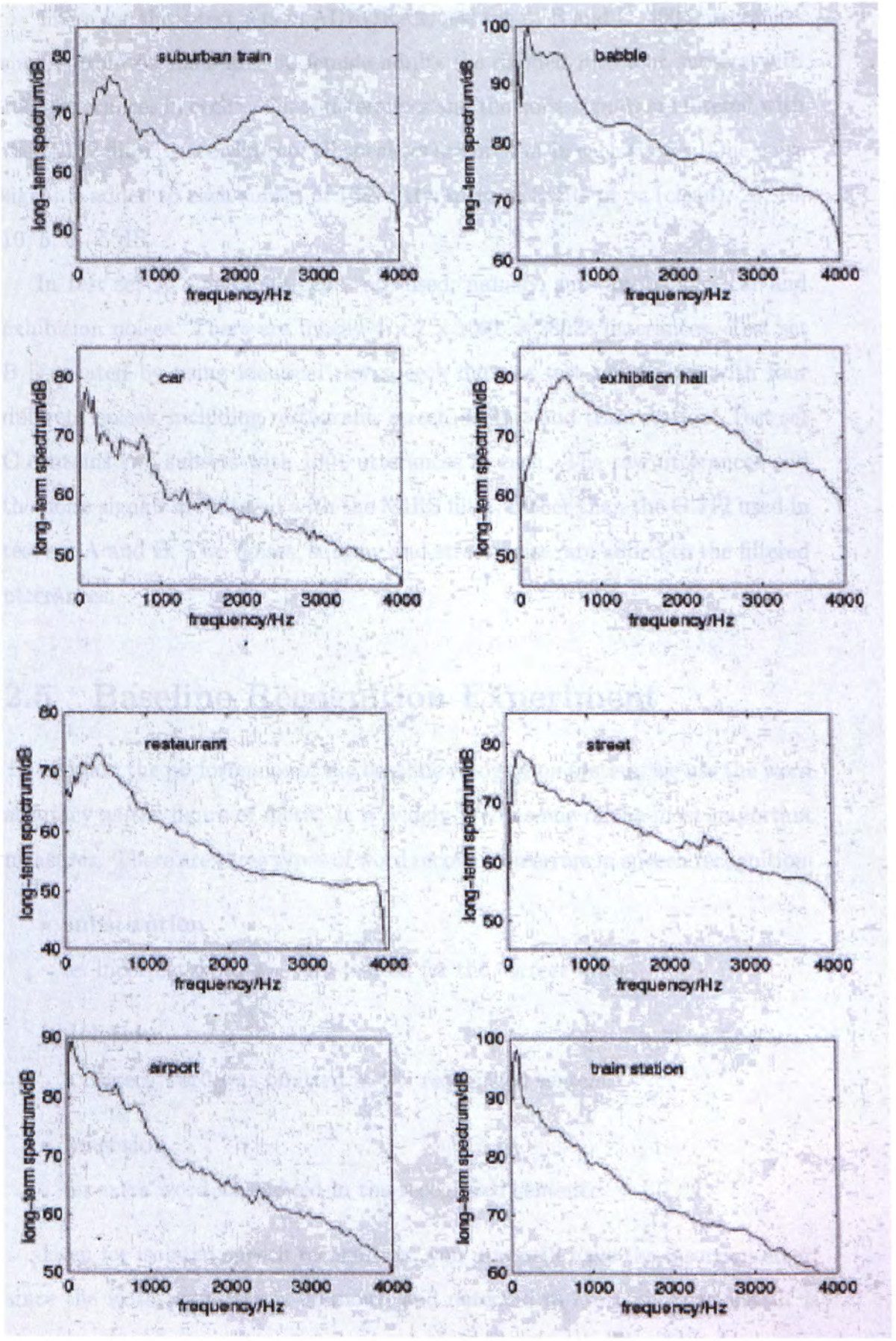


Figure 2.13: The long-term spectra of the eight noises.

There are three test sets in AURORA2, test set A, B and C. 4004 utterances spoken from 52 male and 52 female adults are divided into four subsets with 1001 utterances in each[2]. These utterances and the noise signals are filtered with the G.712 filter. Recordings of all speakers are present in each subset. One noise signal is added to each subset of 1001 utterances at SNRs of ∞ (clean), 20, 15, 10, 5, 0, -5 dB.

In test set A, four noise types are used, namely, subway, babble, car and exhibition noises. There are totally $4 \times 7 \times 1001 = 28028$ utterances. Test set B is created by using identical raw speech data as test set A, but with four different noises, including restaurant, street, airport and train station. Test set C contains two subsets with 1001 utterances in each. The raw utterances and the noise signals are filtered with the MIRS filter, rather than the G.712 used in test set A and B. Two noises, subway and street noise, are added to the filtered utterances.

2.5 Baseline Recognition Experiment

To evaluate the performance of the baseline recognition system, we use the word accuracy as the figure of merit. It is widely used as one of the most important measures. There are three types of word recognition errors in speech recognition:

- **substitution**
  an incorrect word was substituted for the correct word
- **deletion**
  a correct word was omitted in the recognized sentence
- **insertion**
  an extra word was added in the recognized sentence

Even for isolated speech recognition, you may still have the insertion error, since the word boundary is unknown and detected in most applications. It is thus possible that a isolated utterance is recognized as two words.

After counting the number of substitution, deletion and insertion errors, the word accuracy can be calculated. Let S, D, I and N be the number of substitution error, deletion error, insertion error and the total number of word in the correct sentence respectively. The word accuracy is defined as,

$$word\ accuracy = \frac{N - (S + D + I)}{N} \times 100\% \tag{2.23}$$

and is in the unit of percentage.

The corresponding word accuracy of the baseline system is shown in Table 2.2. Although the recognizer performs well when the inputs are clean speech, there is a significant degradation in recognition performance when the signal-to-noise ratio, SNR changes from high to low, such as from 15 dB to 10 dB. This may be due to the mismatched conditions between training and testing scenarios, as most standard recognition systems, including our baseline system, are trained from clean speech data, while testing is commonly done on various noisy environments. Hence, recognition degradation is unavoidable and it is necessary to improve the recognition performance under noisy environments.

HMM has been adopted in our baseline system, where the statistical properties of certain speech classes are modelled, such as the mean and the covariance. From the recognition result shown in Table 2.2, the word accuracy is found to be unable to reach 100% even for clean speech. Possible reasons for this phenomenon include (1) speech samples in the test sets which are outliers from the training set or (2) the configurations of acoustic models are not optimal that the trained models are not good representatives for different speech classes. However, the baseline system achieves nearly 99% for clean speech, which is already comparable with other current connected-digit recognizers.

[2]The utterances in each subset are distinct, hence, the clean speech recognition results in each of subset are different.

test A in clean training

SNR/ dB	subway	babble	car	exhibition	average
clean[2]	98.83	98.97	98.81	99.14	98.94
20	96.96	89.96	96.84	96.20	94.99
15	92.91	73.43	89.53	91.85	86.93
10	78.72	49.06	66.24	75.10	67.28
5	53.39	27.03	33.49	43.51	39.36
0	27.30	11.73	13.27	15.98	17.07
-5	12.62	4.96	8.35	7.65	8.40
average between 0 and 20 dB	69.86	50.24	59.87	64.53	61.13

test B in clean training

SNR/ dB	restaurant	street	airport	train-station	average
clean[2]	98.83	98.97	98.81	99.14	98.94
20	89.19	95.77	90.07	94.38	92.35
15	74.39	88.27	76.89	83.62	80.79
10	52.72	66.75	53.15	59.61	58.06
5	29.57	38.15	30.39	29.74	31.96
0	11.70	18.68	15.84	12.25	14.62
-5	5.00	10.07	8.11	8.49	7.92
average between 0 and 20 dB	51.51	61.52	53.27	55.92	55.56

test C in clean training

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	99.02	98.97	99.00
20	94.47	95.19	94.83
15	87.63	89.69	88.66
10	75.19	75.27	75.23
5	52.84	48.85	50.85
0	26.01	21.64	23.83
-5	12.10	10.70	11.40
average between 0 and 20 dB	67.23	66.13	66.68

Table 2.2: Word accuracy of the baseline system.

Chapter 3

A Simple Recognition Framework with Model Selection

To improve the robustness of ASR, in this chapter, the reasons of the performance degradation are first explored. Knowing that there is often a mismatch between the training and testing conditions, a recognition framework is introduced to reduce this mismatch by looking at the noise type and the signal-to-noise ratio (SNR). Finally, a simple and effective framework is proposed to improve the recognition accuracy, which selects the best-matched acoustic model according to the SNR of the input noisy speech. The term SNR below refers to the global SNR unless specified.

3.1 Mismatch between Training and Testing Conditions

The recognition experiment shown in Chapter 2 indicates a problem. Even if a speech recognition system performs remarkably well in laboratory evaluations with a clean environment, it often performs not nearly as well in real situations where background noise always exists. This is mainly because the speech that actually has to be recognized varies from conditions to conditions and usually differs from the training speech. Some of the previous research work have reported that even the awareness of speaking to a speech recognizer could make

the speaker produce a noticeable difference [28].

Conventional ASR frameworks are based on training using clean speech data. These ASR systems are very sensitive to additive noise and/ or channel distortion found in the input speech, which causes a mismatch between the clean training speech and the corrupted input. The ASR performance, hence, severely degrades.

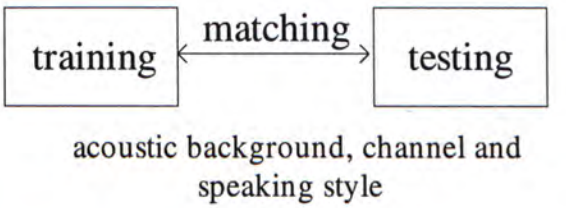


Figure 3.1: Matching between training and testing conditions.

Matching refers to the similarity between training and testing conditions, such as additive background noise, channel distortion or speaking style, as illustrated in Figure 3.1. It is highly critical for ASR [4]. Even when the test data is obtained in a reasonably quiet environment, the recognition accuracy may decrease if training is done in a much higher SNR condition, such as when the test data are collected using a close-talk high quality microphone in a soundproof chamber. On the other hand, if training is performed under the same condition as those under which the speech is to be recognized, better matching and recognition performance could be achieved.

Take an example. Dautrich, Rabiner and Martin [29] demonstrated that an isolated word recognizer trained in clean condition and capable of achieving a recognition accuracy of 95% has an order of magnitude decrease in word accuracy when tested with noise-corrupted speech at SNR of 18 dB. Figure 3.2 shows the recognition accuracy at various SNRs.

The line with $\triangle$ marker shows the baseline recognition accuracy of the Dautrich system. Although the baseline performance is worse than most of the current insolated ASR systems, this recognition experiment illustrates several major considerations. In particular, the recognizer can maintain the perfor-

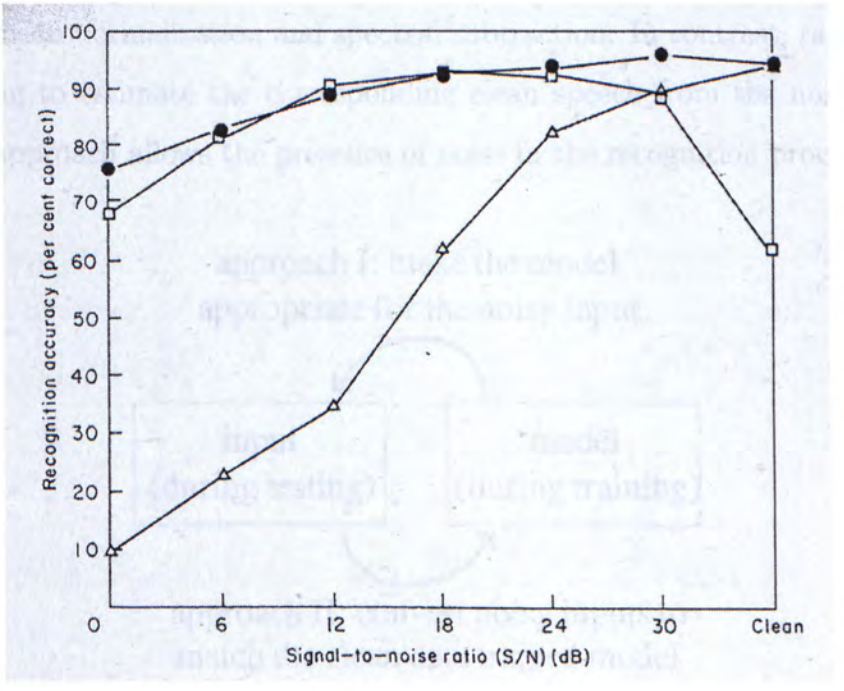


Figure 3.2: The word accuracy of noisy speech recognition under various SNRs. •: both training and testing are under same SNR; △: only clean speech is used for training and testing inputs are under different SNRs indicated by the marker; □: training and testing conditions are mismatched with testing SNRs all at 18 dB and training SNRs are indicated by the marker.

mance with only moderate degradation when the SNR decreases. This is shown by the line with the • marker. For proper recognition performance, matching should be maintained.

To increase the degree of matching between the two conditions, there are two possible directions. Referring to Figure 3.3, (1) the acoustic model is adjusted to the input speech; or (2) the characteristics of the noisy input are adjusted to fit the model trained from clean data. Typical examples of the second approach are zero-mean normalization and spectral subtraction. In contrast, rather than attempting to estimate the corresponding clean speech from the noisy input, the first approach allows the presence of noise in the recognition process.

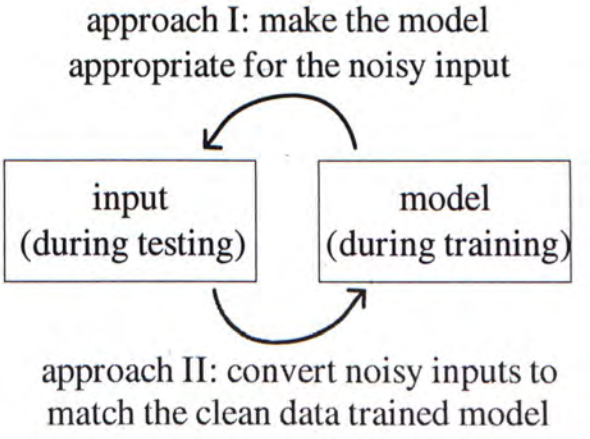


Figure 3.3: Two possible directions to increase the degree of matching between testing and training conditions.

The first approach is used in this chapter. The acoustic model can be adjusted to match one single or multiple properties of the input speech. These properties include the noise type, SNR and the speaker style. Since our recognition task is speaker-independent, the potentials of noise type and SNR matching for noisy speech recognition are investigated in the following.

3.2 Matched Training and Testing Conditions

Assume that a number of acoustic models are available. By choosing the model which is matched to the given noisy input, the training and testing conditions are matched and it is expected that when the SNR decreases, the recognition degradation should be reduced.

The two conditions can be matched according to the noise types, SNRs or both. Given a noisy speech input $y(t)$, let $x(t)$ and $n(t)$ be the corresponding clean speech and the noise signal respectively. We have,

$$y(t) = x(t) + n(t) \tag{3.1}$$

Several experiments have been conducted, so as to evaluate the effectiveness of the two matching on noisy speech recognition. The experimental details and results are reported below.

3.2.1 Noise type-Matching

In the training data set in AURORA2, there are four noise types. They are subway, babble, car and exhibition noises. One model is trained for each noise type and hence, the whole training set is divided into four subsets. Originally, the training set contains $422 \times 5 \times 4 = 8440$ utterances. After dividing into four subsets, only $422 \times 5 = 2110$ utterances are used for each model training. This is the recognition system with noise type-matching.

During testing, the noise type of the input noisy speech is assumed to be known and the model trained with the same noise type is used. For example, for inputs corrupted by babble noise, the model trained with babble noisy speech is applied.

The recognition results are shown in Table 3.1. It is found that the recognition system with noise type-matching outperforms the baseline (Table 2.2) by $89.85 - 61.13 = 28.72\%$ average in test A absolute word accuracy. In low SNR conditions, the gains are even more promising, even when SNR equals 0 dB, the recognition accuracy increases up to about 67.82% from the baseline 17.07%.

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.34	98.49	98.57	98.33	98.43
20	98.19	97.85	98.06	97.75	97.96
15	97.39	97.31	97.91	97.35	97.49
10	95.95	95.56	96.66	94.91	95.77
5	92.05	88.27	91.14	89.48	90.24
0	71.81	61.88	67.10	70.47	67.82
-5	29.17	26.57	22.04	29.03	26.70
average between 0 and 20 dB	91.08	88.17	90.17	89.99	89.85

test B

SNR/ dB	restaurant	street	airport	train-station	average
clean	98.34	98.49	98.57	98.33	98.43
20	86.52	96.70	94.81	90.81	92.21
15	74.27	95.13	91.17	77.57	84.54
10	58.06	91.44	83.66	56.93	72.52
5	38.38	79.56	72.71	34.77	56.36
0	9.15	52.60	50.76	15.43	31.99
-5	-9.86	19.95	14.38	2.78	6.81
average between 0 and 20 dB	53.28	83.09	78.62	55.10	67.52

test C

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	98.53	98.37	98.45
20	98.10	94.11	96.11
15	97.02	90.08	93.55
10	94.38	78.75	86.57
5	84.65	60.25	72.45
0	50.72	32.16	41.44
-5	21.55	16.29	18.92
average between 0 and 20 dB	84.97	71.07	78.02

Table 3.1: Word accuracy of the recognition system with noise-type matching.

Regarding the recognition performance of test B and C, significant improvements are observed in street, airport, subway(MIRS) and street(MIRS) cases only, but not in restaurant nor train-station cases. From the experimental results, it is concluded that this noise-type matching is useful for noisy speech recognition.

However, since some of the noise are non-stationary and only one-fourth of the original training amount is used in each model training, we may alternatively have smaller number of models trained with larger amount of training data for each of them. Among the eight noise types, some of them are similar to each other in properties, for example, babble is much close to exhibition, and street and train-station noise contain human speech and noise from travel vehicles. The following experiment divides the eight noise types into three groups and studies how this grouping may affect the recognition accuracy.

According to the noise types, they are categorized into three groups of,

- group I, street and train-station
- group II, subway and car
- group III, babble, exhibition, restaurant and airport

Recall that the AURORA2 training data set contains only speech corrupted from subway, babble, car and exhibition noise. For group II and III, the training process uses the corresponding speech data. For group I (street and train-station), the model trained by group II is used for recognition, which is more similar in noise property. Therefore, there are two models which are trained by group II (subway and car) and group III (babble and exhibition) respectively. For each model training, $422 \times 5 \times 2 = 4220$ utterances are used. During testing, the noise type of the input noisy speech is assumed to be known and the recognition model used is selected according to the grouping. This is the recognition system with similar noise type-matching. The recognition results are shown in Table 3.2.

Comparing the recognition results of this experiment with the baseline performance (Table 2.2), this similar noise-type matching brings recognition im-

test A

SNR/ dB	II, subway	III, babble	II, car	III, exhibition	average
clean	98.50	98.52	98.33	98.49	98.46
20	97.88	97.85	97.61	97.28	97.66
15	96.99	97.13	97.58	96.58	97.07
10	95.39	95.50	96.00	93.80	95.17
5	89.68	88.09	89.05	87.69	88.63
0	67.18	61.85	59.26	63.93	63.06
-5	28.28	26.57	20.46	23.94	24.81
average	89.42	88.08	87.90	87.86	88.32

test B

SNR/ dB	III, restaurant	I, street	III, airport	I, train-station	average
clean	98.53	98.52	98.45	98.49	98.50
20	97.08	97.64	97.38	97.13	97.31
15	94.96	95.95	96.15	95.31	95.59
10	91.99	94.07	92.84	92.69	92.90
5	83.48	83.22	85.24	82.04	83.50
0	58.70	57.26	62.78	53.75	58.12
-5	23.49	24.18	26.66	19.01	23.34
average	85.24	85.63	86.88	84.18	85.48

test C

SNR/ dB	II, subway(MIRS)	I, street(MIRS)	average
clean	98.46	98.46	98.46
20	97.30	96.77	97.04
15	96.41	95.74	96.08
10	92.97	92.32	92.65
5	81.76	81.77	81.77
0	48.45	51.48	49.97
-5	20.14	22.07	21.11
average	83.38	83.62	83.50

Table 3.2: Word accuracy of the recognition system with similar noise-type matching.

provement in nearly all cases, except when the inputs are clean speech. The average absolute improvement is $88.32 - 61.13 = 27.19\%$, which is only slightly smaller than the previous noise type-matching experiment. When the input SNR is low, the improvement over the baseline remains significant, showing a comparable result with the previous noise type-matching experiment.

Although the number of speech utterances used for training in the first experiment is only half of the current experiment, the recognition performance is not affected.

Comparing the two recognition accuracies for test A, the first experiment always achieves better performance than the current experiment. As the data in test A are used to train models, the four noises, subway, babble, car and exhibition are seen. During recognition, the first experiment uses the model which is trained by data corrupted by the same noise. The current experiment (recognition system with similar noise type-matching) uses the model trained by the same group only, but there are training data with a different noise type. For example, to recognize the subway data set, the current experiment uses the model trained by subway or babble data. In the sense of matching between the training and testing condition, the first experiment (recognition system with noise type-matching) is better matched than the current experiment (recognition system with similar noise type-matching).

Superior recognition performance has been found in test set B, even in restaurant and train-station cases, where the first experiment (recognition system with noise type-matching) does not produce apparent improvement. The current experiment uses the model trained by babble and exhibition data set for recognition, whilst in the first experiment, one noise type is used for each model training. As test B is a testing data set, where the noises are unseen during training, using different noises for training is expected to produce better recognition performance than using only a single noise type.

In the current experiment (recognition system with similar noise type-matching), restaurant and train-station test sets use the model from group III. Note that the group III model is trained by the babble and exhibition data. In

the first experiment, restaurant test set uses the model trained from subway data and train-station test set uses the model trained from exhibition data. From the noise property and perception aspects, restaurant noise is non-stationary and contains human speeches, similar to the babble or exhibition environment, but subway contains stationary vehicle noise. For the train-station data, it contains non-stationary human speech and noise from vehicles. Hence, it is more appropriate to use the group III model to recognize restaurant data, rather than the subway model.

3.2.2 SNR-Matching

As shown in the previous section, by choosing the matched model for a given noisy speech, the recognition degradation due to the noise contamination can be greatly reduced. Rather than the noise type, the global signal-to-noise ratio (SNR) is used in this section, to choose the most appropriate model for recognition.

Let SNR_g be the global signal-to-noise ratio given by

$$SNR_g = 10\log_{10}\{\frac{\sum_t[y(t)]^2}{\sum_t[n(t)]^2} - 1\} \tag{3.2}$$

The AURORA2 database contains noisy utterances at SNRs: ∞ (clean), 20 dB, 15 dB, 10 dB, ..., -5 dB with a 5 dB step. Training data sets include all utterances at SNRs from 5 dB to 20 dB and the clean data set. It is divided into three groups, according to the SNR values. These groups are called high SNR, medium SNR and low SNR. Each group has its own model. The high SNR group contains all clean utterances. There are $422 \times 4 = 1688$ utterances used for training the high SNR model. The medium SNR group contains speech data from either 15 dB or 20 dB data sets. The low SNR group contains speech data from either 5 dB or 10 dB data sets. For both medium and low SNR groups, the number of utterances used for training is $422 \times 2 \times 4 = 3376$.

During testing, the SNR of the input noisy speech is assumed to be known and is calculated with Equation (3.2) by finding the corresponding clean speech $x(t)$. Testing is carried out by using the model trained by the matched SNR,

except to those testing data with SNR equal to -5 or 0 dB, which uses the low SNR system. Table 3.3 shows the recognition results.

Concerning the recognition results, the test A accuracies found in noisy speech inputs are similar to those in the previous two experiments, with an average absolute improvement of $89.83 - 61.13 = 28.7\%$ and when the SNR is 0 dB, the recognition accuracy increases from 17.07% to 64.46%. For clean speech inputs, the decrease in clean speech accuracy is the smallest one among the three experiments. This may be because the training mode is matched to SNR, rather than the noise type.

For test B, significant improvement is found with the SNR-matching in all cases. For test C, comparing the recognition results of the three experiments, the system with SNR-matching brings the highest recognition accuracies.

3.2.3 Noise Type and SNR-Matching

From previous experiments, it is found that both SNR-matching and noise type-matching are essential to reliable noisy speech recognition. Both the first experiment (recognition system with noise type-matching) and the last experiment (recognition system with SNR-matching) reach satisfactory word accuracy with two different approaches – matching noise type or matching SNR. In this experiment, combined noise type and SNR matching is used and many more models are trained.

For every combination of noise type and SNR, a model is built. There are $4 \times 5 = 20$ models in total. For training a model for noise type α and SNR β dB, any utterance that is corrupted by α or with a global SNR β dB is used for training. For example, to obtain a model for the babble noise 10 dB system, any utterance that is corrupted by babble noise or with a global SNR equal to 10 dB is used for training this babble noise at 10 dB system. The number of utterances used in each system is $(4 + 5 - 1) \times 422 = 3376$.

The noise type and the SNR of the input noisy speech are assumed to be known during testing and used to select a model for recognition. Testing is carried out by using the model trained with matched SNR and noise type,

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.86	98.88	98.78	99.01	98.88
20	98.16	98.04	98.12	97.84	98.04
15	97.05	97.31	97.64	97.13	97.28
10	94.38	94.95	96.06	93.71	94.78
5	90.08	88.24	91.02	88.98	89.58
0	72.77	62.64	68.00	74.42	64.46
-5	34.51	19.41	25.74	34.90	28.64
average between 0 and 20 dB	90.49	88.24	90.17	90.42	89.83

test B

SNR/ dB	restaurant	street	airport	train-station	average
clean	98.86	98.88	98.78	99.01	98.88
20	97.67	97.73	97.70	97.59	97.67
15	95.86	96.74	96.51	95.93	96.26
10	88.58	93.23	91.41	92.47	91.42
5	79.18	83.71	84.46	83.89	82.81
0	51.92	59.07	63.20	59.73	58.48
-5	6.45	21.64	18.55	20.89	16.88
average between 0 and 20 dB	82.64	86.10	86.66	85.92	85.33

test C

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	99.02	98.88	98.95
20	97.85	97.04	97.45
15	96.19	95.86	96.03
10	93.40	92.62	93.01
5	85.20	84.28	84.74
0	53.58	57.47	55.53
-5	18.94	21.34	20.14
average between 0 and 20 dB	85.24	85.45	85.35

Table 3.3: Word accuracy of recognition system with the SNR matching.

except to those testing data with SNR equal to -5 or 0 dB, which uses the model trained by matched noise type and 5 dB data. Table 3.4 shows the recognition results.

Observing the word accuracy rates, the current recognition (system with noise type and SNR-matching) produces the best average results in both test A and B. The average word accuracy in test A and B are about 90.11% and 85.5% respectively. Even in test C, the average word accuracy of 84.9% is close to the maximum accuracy rate produced by the third experiment (recognition system with SNR-matching), which is about 85.35%. In low SNR conditions, such as, when SNR is 0 dB, the average word accuracy in test A is the best at about 70%.

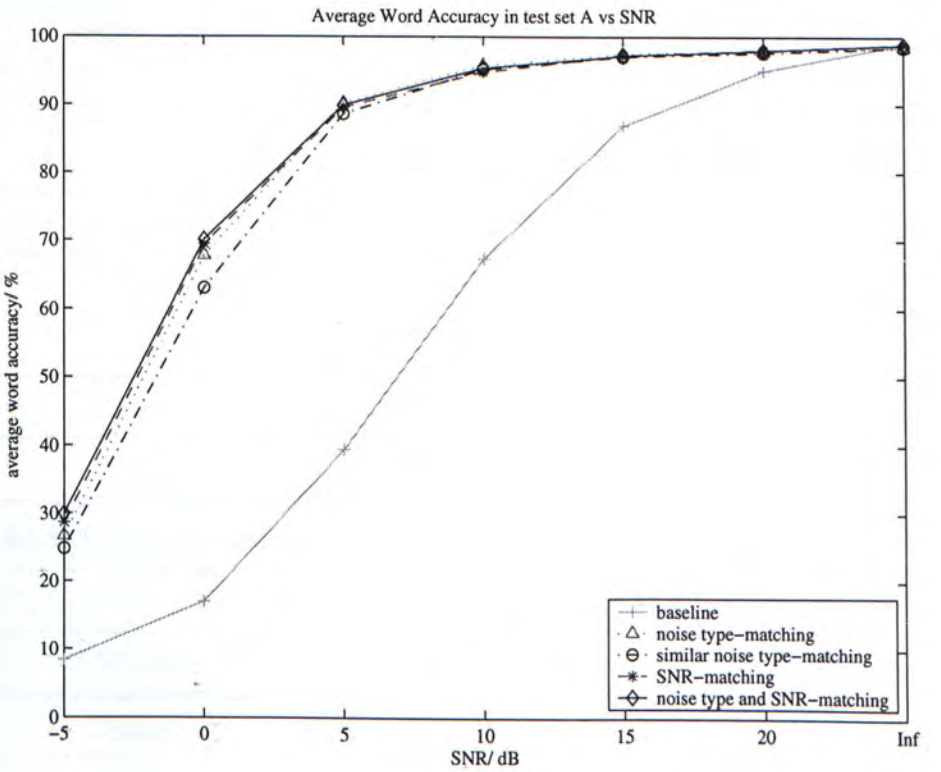


Figure 3.4: The average word accuracy in test set A versus SNR.

An overall average word accuracy plot is shown in Figure 3.4. Comparing the performance between the four experiments and the baseline system, all the systems with matching outperform the baseline system. When SNR increases, the differences between the four experiment systems become insignificant. Besides,

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.77	98.76	98.87	99.01	98.85
20	98.37	98.00	98.06	97.38	97.95
15	97.24	97.40	97.70	96.45	97.20
10	95.12	95.53	96.18	94.63	95.37
5	90.85	87.88	91.35	89.66	89.94
0	74.06	64.06	69.07	73.25	70.11
-5	33.93	28.11	23.77	33.94	29.94
average between 0 and 20 dB	91.13	88.57	90.47	90.27	90.11

test B

SNR/ dB	restaurant	street	airport	train-station	average
clean	98.77	98.76	98.87	99.01	98.85
20	96.96	97.61	97.05	97.01	97.16
15	93.86	96.34	94.96	95.16	95.08
10	90.76	94.01	92.78	92.93	92.62
5	81.21	83.19	83.27	82.38	82.51
0	56.65	61.85	64.00	57.95	60.11
-5	17.19	27.60	22.88	20.49	22.04
average between 0 and 20 dB	83.89	86.60	86.41	85.09	85.50

test C

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	98.96	98.64	98.80
20	97.94	97.01	97.48
15	96.81	95.62	96.22
10	93.64	92.62	93.13
5	83.48	81.80	82.64
0	52.99	57.16	55.08
-5	18.94	24.76	21.85
average between 0 and 20 dB	84.97	84.84	84.91

Table 3.4: Word accuracy the recognition system with matched noisetype or SNR training

the rate of increase in average word accuracy becomes smaller and smaller when SNR increases. In low SNR conditions, the accuracy rates of the system with noise type-matching, SNR-matching and both noise type and SNR-matching are close to each other and higher than the one in the system with similar noise type-matching.

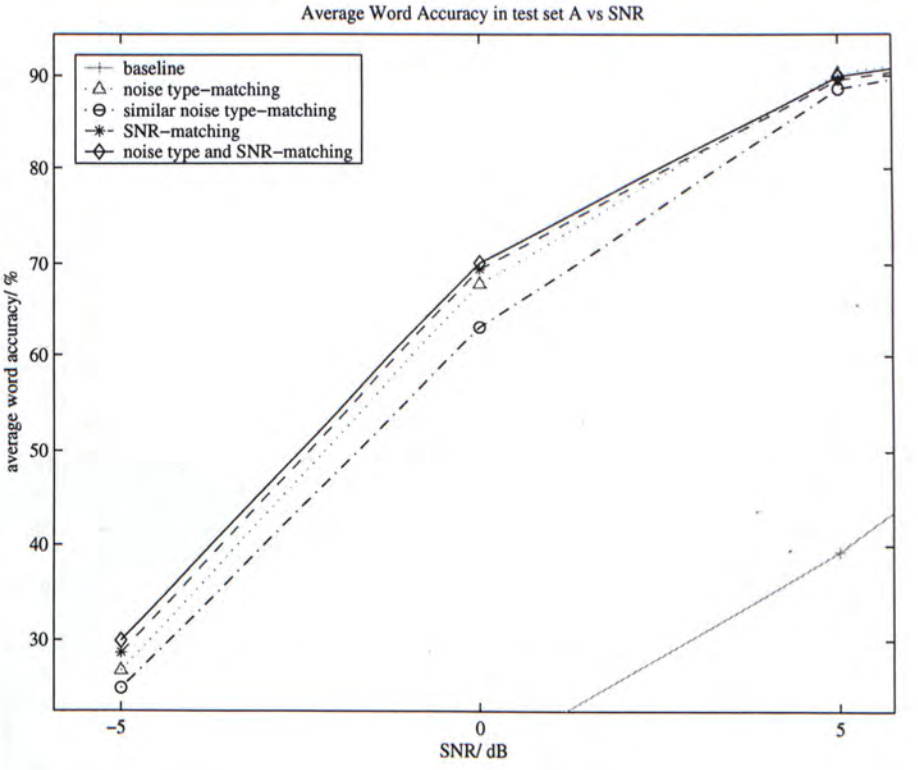
Figure 3.5(a), 3.5(b) and 3.5(c) show the magnified views of the word accuracy versus SNR plot. Using systems with SNR-matching or noise type and SNR-matching can always produce satisfactory recognition. This implies that SNR-matching is very useful in robust speech recognition.

This is extremely valuable for noise types which are not present during training. For test set B, the recognition performance of the systems with SNR-matching or noise type and SNR-matching is much better than the one using noise type-matching (referring to the average word accuracy in test set B among the four experiments in Figure 3.6). Note that in the last two experiments, the number of speech samples used in each individual system training is only 3376.

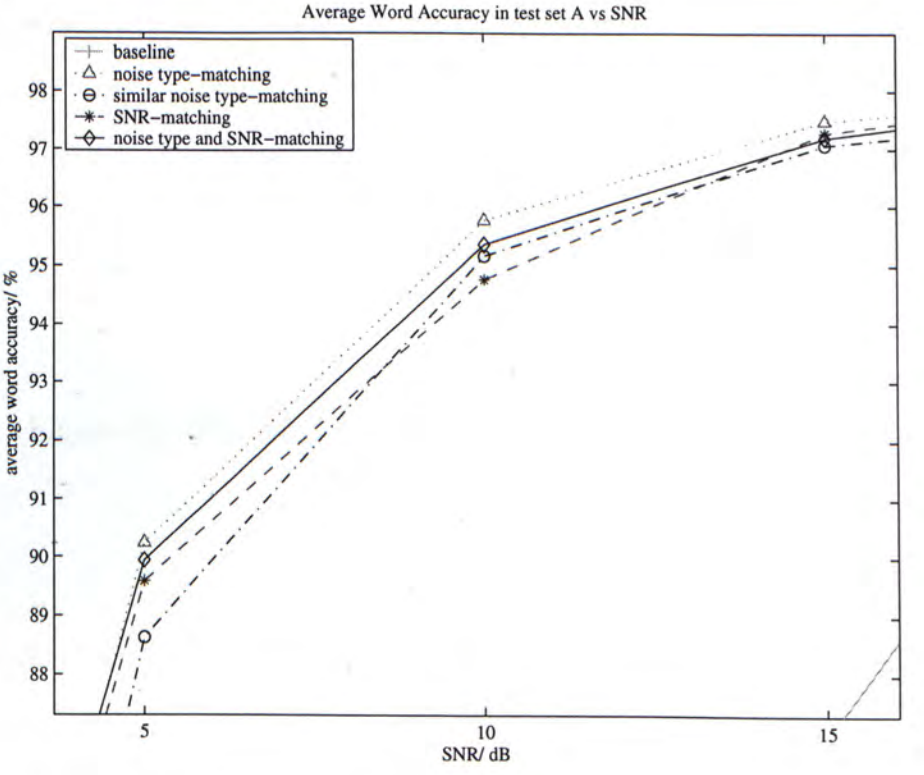
Figure 3.7 shows the average word accuracy in test set C. Similar recognition performance as with test set B is achieved. The systems with the noise type and SNR-matching or simply only the SNR-matching are always the best in recognition performance. The performance difference to the system with similar noise type-matching is much larger. As there is no severe recognition degradation when moving from test set B to C, it can be concluded that SNR-matching has certain robustness towards channel responses which are different from the one seen during training.

3.3 Recognition Framework with Model Selection

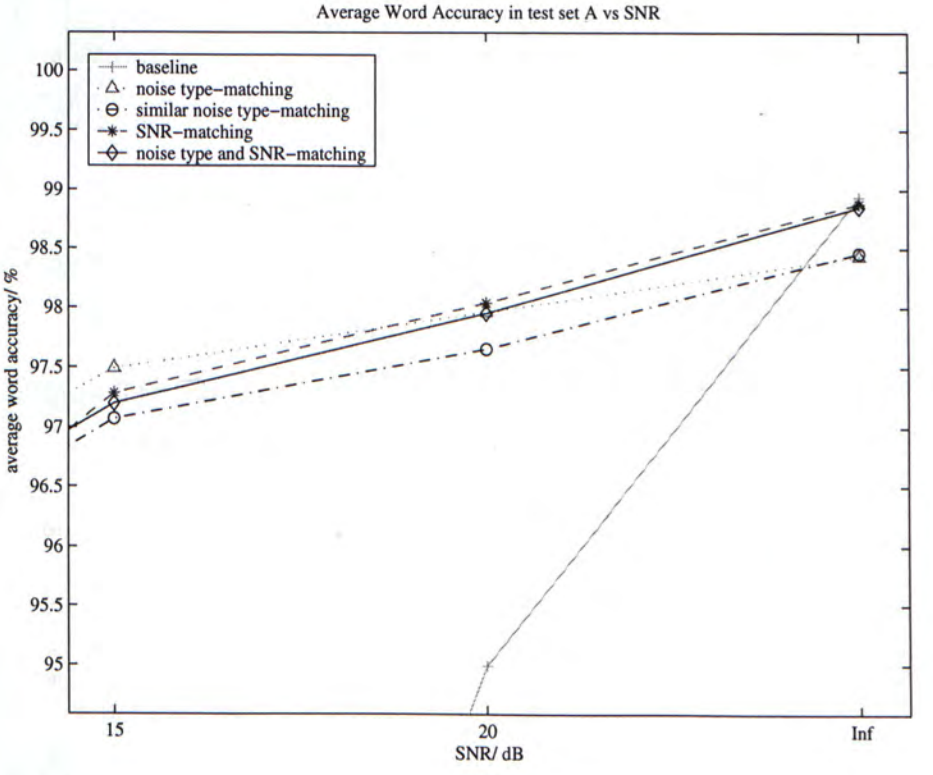
From the experiments shown in Section 3.2, SNR-matching or noise type and SNR-matching are very effective for noisy speech recognition. They both provide promising improvements. The training and testing conditions are matched by selecting the most appropriate one out from a pool of models. It is previously



(a) low SNR



(b) medium SNR



(c) high SNR

Figure 3.5: The magnified average word accuracy versus SNR.

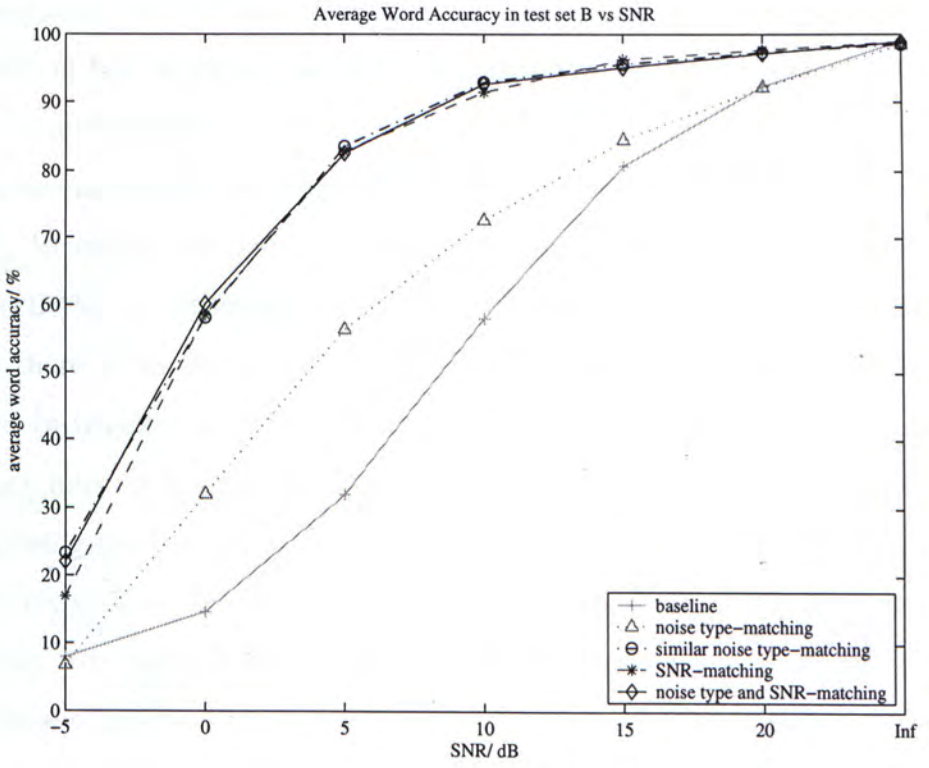


Figure 3.6: The average word accuracy in test set B versus SNR.

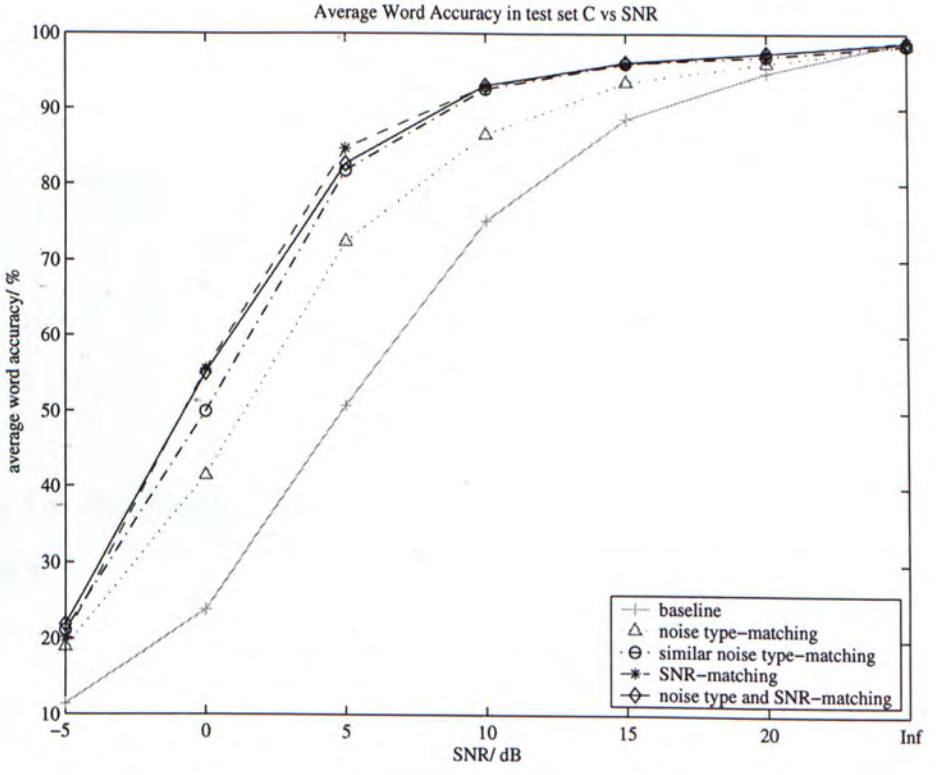


Figure 3.7: The average word accuracy in test set C versus SNR.

assumed that the SNR or the noise type of the noisy speech input are known, however, in real applications, they are unknown and required to be estimated before model selection.

To estimate the SNR, there are widely-used algorithms with different successes. Examples are, simple estimation during speech pauses, the histogram approach [30] or estimation from a microphone array. On the contrary, currently there is no standard algorithm to determine the noise type of a noisy speech. In Chapter 4, the idea of the SNR-matching is adopted to provide satisfactory improvement for noisy speech recognition and avoid the difficulties of determining the noise type. Figure 3.8 depicts the overall block-diagram of the simple recognition framework with model selection. By estimating the noise spectrum, the global SNR is calculated and the best-matched model is chosen accordingly. Identical to the experiment of SNR-matching in Section 3.2.2, the three acoustic models, namely high SNR, medium SNR and low SNR, are used.

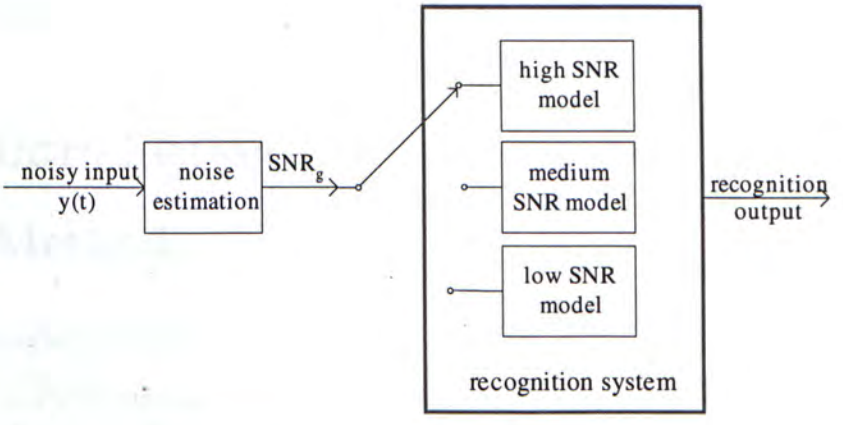


Figure 3.8: The block-diagram of the simple recognition framework with model selection.

Chapter 4

Noise Spectral Estimation

This chapter continues the work from Chapter 3 and suggests a statistical noise estimation method. A review of current estimation methods is given. By studying how speech harmonic structure affects noise estimation, an estimation method called Mainlobe-Resilient Time-Frequency Quantile-base Noise Estimation is proposed. It is designed to prevent the overestimate of the noise power and provide a good tracking at harmonic frequencies. Evaluations on the noise estimation and recognition performance are shown at the end.

4.1 Introduction to Statistical Estimation Methods

In the recognition framework suggested in Section 3.3, there are two building blocks of noise estimation and model selection.

Noise Estimation has been a popular research topic for over the past 20 years. There are various applications from speech enhancement to robust speech recognition. A statistically-based noise estimation method is proposed in this chapter, which works together with the model selection framework. Before looking at this new estimation method, the following attempts to discuss several classical ways to perform noise estimation. Most estimation methods perform the estimate in the frequency domain. Basically, these methods are classified into two groups, voice-activity detection-based or statistical-based.

4.1.1 Conventional Estimation Methods

Noise estimate is conventionally obtained from a reference signal or during speech pauses. In the application of adaptive noise cancellation (ANC) [31], two configurations can be used to capture the reference time-domain signal:

1. The reference microphone (used to collect the reference signal) is placed next to the noise source and another microphone which is called the primary microphone is placed close to the desired speech source located far from the reference microphone.

2. An acoustic barrier is used and is located between the primary and the reference microphones.

The waveform captured by the reference microphone is often used as the noise signal. In real applications, it is not always possible to place the two microphones far apart with the reference microphone very close to the noise source. For the second configuration, the acoustic barrier must provide a strong isolation between the speech signal and the interfering noise, which may require extra equipments to be put on by the speaker.

These two configurations attempt to spatially separate the noisy speech $y(t)$ into two components $x(t)$ and $n(t)$. Nevertheless, no array signal processing technique is used. When the two sources are close to each other, the reference signal always contains a strong speech component, leading to an inaccurate noise estimation. For the present task of connected digit recognition, there is only one single microphone.

Apart from using two microphones to spatially separate the noisy speech, it was proposed that the noise estimate can be found during speech pauses [9, 32]. A simple way to estimate the noise spectrum is to average the spectra within a the short duration before the speech signal commences.

To cope with non-stationary noise, the noise estimate should be updated regularly. This is usually done by detection of speech pauses to locate segments of pure noise. The detection of speech pauses is commonly referred to as the voice-activity detection (VAD), so this type of noise estimation is VAD-based.

A typical example of the VAD-based noise estimation methods is the weighted average method [30]. This method calculates the weighted sum of past spectral magnitude $\sqrt{|Y(\omega,t)|^2}$, where $Y(\omega,t)$ is the coefficient of Fourier Transform of the input speech $y(t)$ at frequency ω and at time t. For each frequency ω, an estimate of the noise magnitude is obtained by a first order recursive equation,

$$|N(\omega,t)| = (1-\alpha)\cdot\sqrt{|Y(\omega,t)|^2} + \alpha\cdot|N(\omega,t-1)| \tag{4.1}$$

where $|N(\omega,t)|$ is the estimated noise magnitude at frequency ω in time t.

In segments of pure noise, the magnitude values $\sqrt{|Y(\omega,t)|^2}$ are Rayleigh-distributed and speech activities are represented by larger magnitude values. The noise estimation should only be updated during speech pauses, hence, a threshold $\beta|N(\omega,t-1)|$ is used to roughly detect when the speech is likely to be present. β normally takes a value in the range of 1.5 to 2.5. When the input spectral magnitude $\sqrt{|Y(\omega,t)|^2}$ is larger than this threshold, a speech signal is detected and the noise estimation by Equation (4.1) is stopped. The noise estimate will be updated again when $\sqrt{|Y(\omega,t)|^2}$ is smaller than or equal to the threshold. As a result, the noise estimate is recursively found by,

$$|N(\omega,t)| = \begin{cases} (1-\alpha)\cdot\sqrt{|Y(\omega,t)|^2} + \alpha\cdot|N(\omega,t-1)|, & \sqrt{|Y(\omega,t)|^2} \le \beta|N(\omega,t-1)| \\ |N(\omega,t-1)|, & \textit{otherwise} \end{cases} \tag{4.2}$$

The weighted average method separates the noise spectrum and the speech spectrum with the use of a threshold. This is actually a VAD operation. In practice, VAD is a difficult task by itself, especially if the background noise is non-stationary or the SNR is low. VAD-based approaches are also unsuitable for fast-changing non-stationary noise, because the noise estimate cannot be updated during speech segments.

4.1.2 Histogram Technique

The histogram technique is based on the statistical analysis of the received spectral values at each frequency [30, 33, 34]. For every frequency, a histogram

of the spectral values is built from several hundred milliseconds of data. During segments of pure noise, the most frequent spectral value is related to the noise level at that particular frequency. The threshold $\beta|N(\omega, t-1)|$ introduced in the weighted average method is still used to roughly separate the segments of noise and speech. The histogram stores only those spectral values smaller than or equal to the threshold and the most frequent spectral value in the histogram is considered as the noise spectral estimate. Finally, the noise estimate is smoothed over time to eliminate rarely occurring spikes.

Note that the histograms are built in the magnitude spectrum domain. Alternatively, the histograms can be built in the log-energy spectral domain.

An evaluation of the two estimation methods in terms of the relative error of noise estimation was reported [30]. This evaluation was made by artificially adding different stationary noise signals to clean speech at different SNRs. The relative error is calculated by,

$$\textit{relative error} = \frac{\sum_{\omega}[|\hat{N}(\omega)| - |N(\omega)|]^2}{\sum_{\omega}|N(\omega)|^2} \tag{4.3}$$

where $|\hat{N}(\omega)|$ and $|N(\omega)|$ are the true average noise magnitude spectra calculated by the noise added and either one of the two estimation methods respectively. The average magnitude spectra are calculated as the sum over all frames.

Figure 4.1 shows the the relative error of adding car noise to utterances from three male and three female speakers. Comparing the two estimation methods, the histogram technique always gives lower relative error than the weighted average. The increase in the error at high SNRs may be due to the incorrect noise estimation at speech segments, because at high SNRs, even a small error leads to a large relative value.

The histogram technique does not rely on explicit speech, non-speech detection. The noise spectrum is estimated during both non-speech and speech segments continuously by finding the most frequent spectral value under a threshold. This is essentially the mode of the distribution. This statistical approach is highly favorable, since VAD can be a major problem in its own right and such

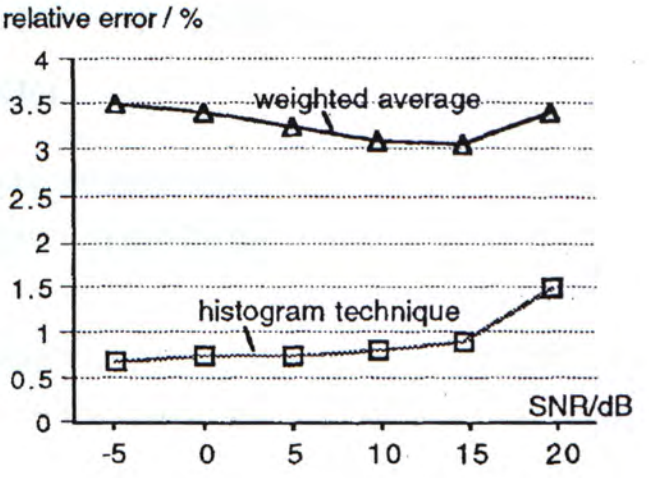


Figure 4.1: The relative error of the noise power spectrum estimation with weighted average and histogram technique.

statistical approach avoids this difficult task and allows the noise estimate to be updated not only close to the boundaries of speech segments but also during speech segments.

4.2 Quantile-based Noise Estimation (QBNE)

Instead of using the mode of the distribution, Martin [35, 36] proposes a noise estimation method which records the minimum values of a smoothed noisy power spectrum. At each frequency, a time window is defined over which the minimum statistics are derived. A similar method was suggested by Arslan *et al.* [37]. The noise estimate is continually updated and is allowed to increase much more slowly than it is allowed to decrease. The noise estimate will increase only slowly during speech segments but collapse quickly in non-speech segments. Therefore, these two methods are unlikely to respond well to increases in noise levels.

An enhanced statistical noise estimation method is proposed for the simple recognition framework with model selection, which is based on the Quantile-based Noise Estimation (QBNE) [38, 39]. The following section introduces the general ideas of QBNE.

4.2.1 Overview of Quantile-based Noise Estimation (QBNE)

The QBNE method was originally developed in [38]. Stahl *et al.* has extended the idea of histogram to quantile-based noise estimation by removing the threshold.

It is well known that even in speech segments, not all frequencies are permanently dominated by the speech power. In fact, there is a significant portion of time that the received power at a certain frequency is due to noise only. QBNE is based on the quasi-periodic characteristic of voiced segments in human speech signals. The noisy power spectrum is the superposition of the noise spectrum and the harmonic spectrum from speech. At inter-harmonic frequencies, the power values are mainly contributed by the additive background noise. At harmonic frequencies, both speech and noise signals contribute to the received power spectrum. If a buffer is used to store the received power spectrum over a short duration and a histogram is built from it, the histogram should be either:

- a uni-modal distribution for the power spectrum at inter-harmonic frequencies, representing the noise power OR
- a bimodal distribution for the power spectrum at harmonic frequencies, related to the superposition of speech and noise spectrum.

QBNE utilizes the uni-modal distribution at inter-harmonic frequencies. The following procedure describes how QBNE estimates the noise spectrum.

Given a noisy speech signal $y(t)$, it is first windowed into segments and the corresponding short-time power spectra are being computed. Let $|Y(\omega,t)|^2$ and $|N_q(\omega,t)|^2$ be the power spectrum of $y(t)$ and the estimated noise power spectrum at frequency ω and time t respectively. For each frequency bin, an buffer stores the value of $|Y(\omega,t)|^2$ over a pre-defined duration T. The buffer content is then sorted and the q-th quantile is taken as $|N_q(\omega,t)|^2$. The process can be summarized as follows:

1. For each segment, take the Fourier Transform and obtain,

$$|Y(\omega,t)|^2, \quad t=0,\ldots,T \tag{4.4}$$

2. For each frequency bin, sort $|Y(\omega,t)|^2$ in ascending order of the power spectral values and re-index,

$$|Y(\omega,b_0)|^2 \leq |Y(\omega,b_1)|^2 \leq \ldots \leq |Y(\omega,b_T)|^2 \tag{4.5}$$

3. Select the q-quantile and assign it as the noise estimate

$$|N_q(\omega,t)|^2 = |Y(\omega,b_{qT})|^2 \tag{4.6}$$

Figure 4.2 illustrates how buffers are used for estimating $|N_q(\omega,t)|^2$ in QBNE. To estimate $|N_q(\omega,t)|^2$ at different frequencies and time, the buffer is shifted accordingly.

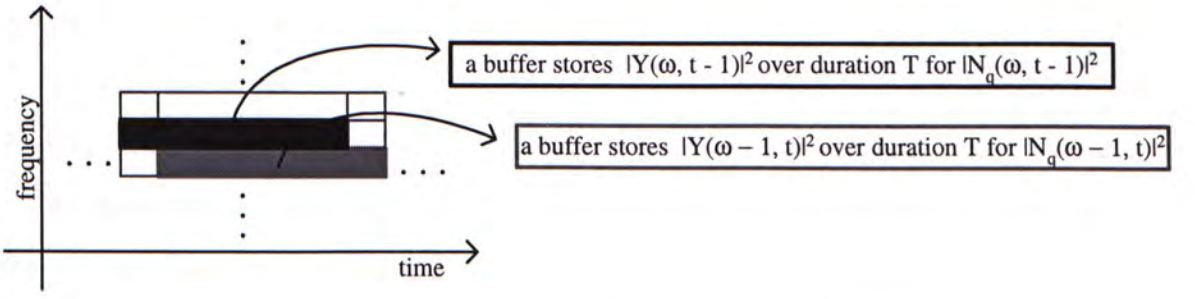


Figure 4.2: How buffers are used in the QBNE calculation.

For example, $q = 0$ yields the minimum, $q = 1$ represents the maximum and $q = 0.5$ gives the median. This algorithm is based on the assumption that each frequency bin carries noise power in at least q portion of time, even during speech segments. This is true for small values of q, but to have a robust estimation of the noise spectrum, that is not sensitive to outliers or speech signals, q should be somewhere around the median ($q \approx 0.5$).

Figure 4.3 shows $|N_q(\omega,t)|^2$ calculated by Equation (4.6) for different q values at three frequencies [38]. The input is a seven digit utterance taken from the

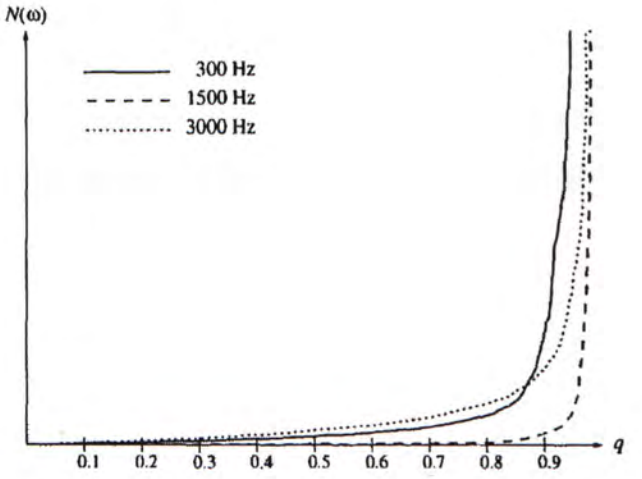
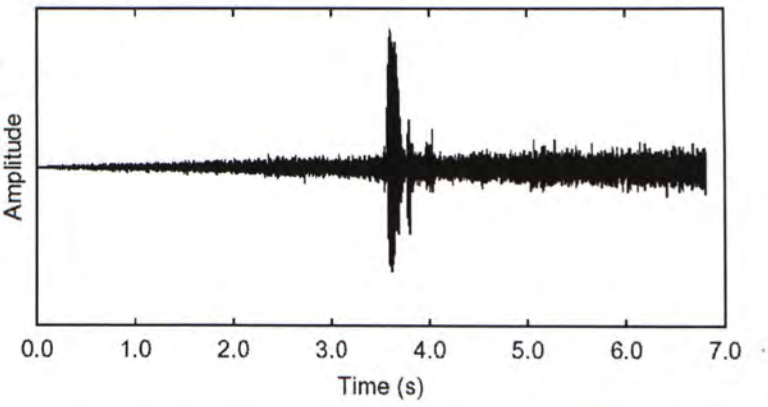


Figure 4.3: Quantiles of the energy distribution of a noisy speech at 300, 1500 and 3000 Hz.

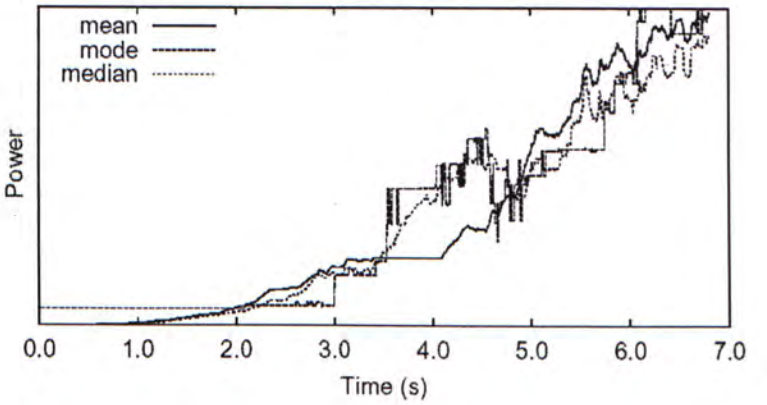
MoTiV corpus. The utterance is mostly in speech segments. It was found that about 80 - 90% of the noisy spectra are low values, which is believed to be close to the noise power level. Only 10 - 20% are high values, which indicates voiced speech segments. This observation is true for different frequencies.

To have fast tracking of non-stationary noise spectrum, the buffer duration T should not be too long. Hence, q should be reduced accordingly.

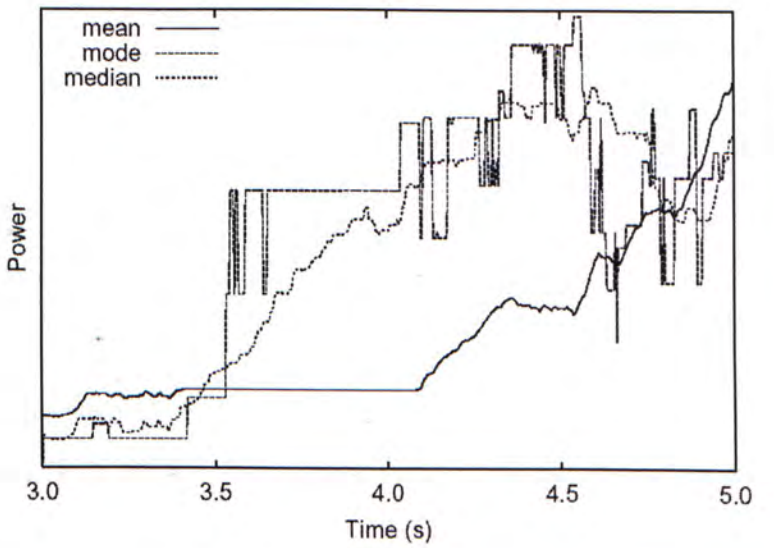
All parameters in QBNE are relative and independent of the absolute spectral values. Referring to Figure 4.4, the noise estimation performance of QBNE is compared against the histogram technique and a standard VAD-based noise estimation in hand-labelled speech pauses [39]. Hand-labelled speech pauses are used so as to circumvent any degradation caused by the VAD errors. The noise level is increased throughout the duration of the utterance. The noise estimate from the VAD-based method remains unchanged throughout the speech segments and for the histogram technique, the distribution of the magnitude spectrum is quantized and the quantization effect on the noise estimate are particularly noticeable, where QBNE does not show this problem. The result shows that QBNE always yields better noise estimation than the other two methods.



(a) time domain waveform of the noisy speech signal



(b) noise power estimate at 100 Hz



(c) magnified view of the noise power estimate during speech segment at 100 Hz

Figure 4.4: The noise power estimates from the three methods, mean represents the VAD-based noise estimation with hand-labelled speech pauses, mode represents the histogram technique and the median represents the QBNE.

4.2.2 Time-Frequency Quantile-based Noise Estimation (T-F QBNE)

In QBNE, the noisy power $|Y(\omega,t)|^2$ is placed in a buffer and the buffer content is numerically sorted. The noise estimate $|N_q(\omega,t)|^2$ is taken as the median value of the buffer. Inevitably, the noise estimate is affected by the presence of speech to some extent.

Referring to Figure 4.5, when a speech signal is present at frequency ω and time t, the current noisy power $|Y(\omega,t)|^2$ probably stays on the right hand side of the median and the buffer contents that correspond to noise power are located in a much lower quantile region. When only noise is received, the current noisy power $|Y(\omega,t)|^2$ is likely to stay on the left hand side of the median and the buffer contents that correspond to noise power are placed within low to high quantile regions.

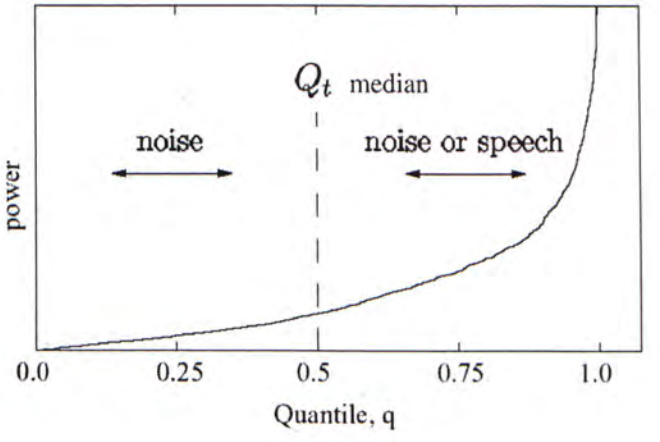


Figure 4.5: The current noisy power $|Y(\omega,t)|^2$ may enter on the left or the right of the median, depends on the presence of speech.

The first case is encountered sometimes because speech signal may consistently dominate at harmonic frequencies. In this case, the noise estimation from QBNE is inaccurate because QBNE only records spectral values along the time axis and most of the buffer contents are contributed by the speech signal. Taking the q-quantile as the noise estimate represents the speech power only, but

not the noise power.

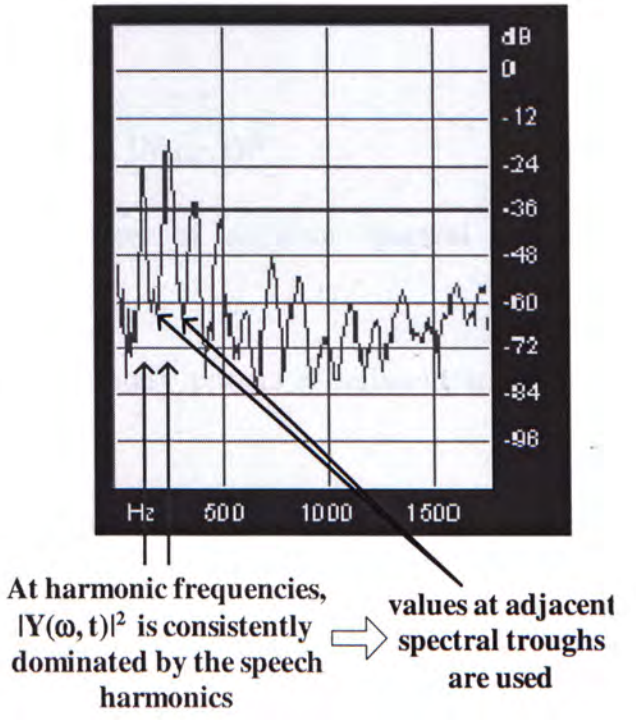


Figure 4.6: The spectral values at adjacent troughs are used for the noise estimation at harmonic frequencies.

To have better estimation at harmonic frequencies, some research studies have proposed to use spectral information in both time and frequency domains [33, 40, 41] to predict the noise level at spectral peaks, as shown in Figure 4.6. This is called the Time-frequency Quantile-based Noise Estimation (T-F QBNE). The following describes the principle of T-F QBNE in details.

T-F QBNE uses different estimation schemes for harmonic and inter-harmonic frequencies. For inter-harmonic frequencies, the noise estimate is set to the QBNE estimate or any combination of this value and the instantaneously received noisy power $|Y(\omega, t)|^2$. For harmonic frequencies, the noise estimate is found by using estimates at adjacent spectral troughs located at either side of the current frequency. Note that for harmonic frequencies, the spectral information in frequency axis is used instead, while the spectral information along the time axis is used for inter-harmonic frequencies. This is because the QBNE

estimate may be degraded by the speech powers in harmonic structure along the time course. Some information can also be used for the noise estimate at frequency ω and time t, together with the noise estimates of adjacent spectral troughs, including,

- the QBNE estimate $|N_q(\omega, t)|^2$
- the QBNE estimates at adjacent spectral troughs $|N_q(\omega_H, t)|^2$ and $|N_q(\omega_L, t)|^2$
- the instantaneous noisy powers st adjacent spectral troughs $|Y(\omega_H, t)|^2$ and $|Y(\omega_L, t)|^2$

where ω_H and ω_L denote the frequencies of the high and low spectral troughs at either side of ω.

Let $|N_{t-fq}(\omega, t)|^2$ be the noise estimate from T-F QBNE found by,

$$\begin{aligned} |N_{t-fq}(\omega,t)|^2 &= \gamma_1|N_q(\omega,t)|^2 + \gamma_2|N_q(\omega_H,t)|^2 + \gamma_3|N_q(\omega_L,t)|^2 \\ &\quad +\gamma_4|Y(\omega_H,t)|^2 + \gamma_5|Y(\omega_L,t)|^2 \end{aligned} \tag{4.7}$$

where γ_i are the weighting factors for the five components. γ_4 and γ_5 are often set to zero, meaning that only $|N_q(\omega, t)|^2$, $|N_q(\omega_H, t)|^2$ and $|N_q(\omega_L, t)|^2$ are taken into consideration.

T-F QBNE first estimates the noise power for every frequency ω at time t by QBNE. For any harmonic frequencies, it further utilizes the spectral values at adjacent troughs to revise the noise estimates. These spectral values can be the QBNE estimate and the instantaneous noisy powers. Therefore, the buffer should be a 2-dimensional array storing spectral values along both time and frequency axes, as shown in Figure 4.7.

In Equation (4.7), a weighted sum is used for noise estimation. Very often, interpolation between the neighborhoods is used to find the noise level for a given harmonic frequency. This implementation also ensures a smooth spectral change around the harmonic peak.

Regarding the interpolation, the boundaries of the interpolation are located at constant and equal distance on each side of every harmonic frequency found

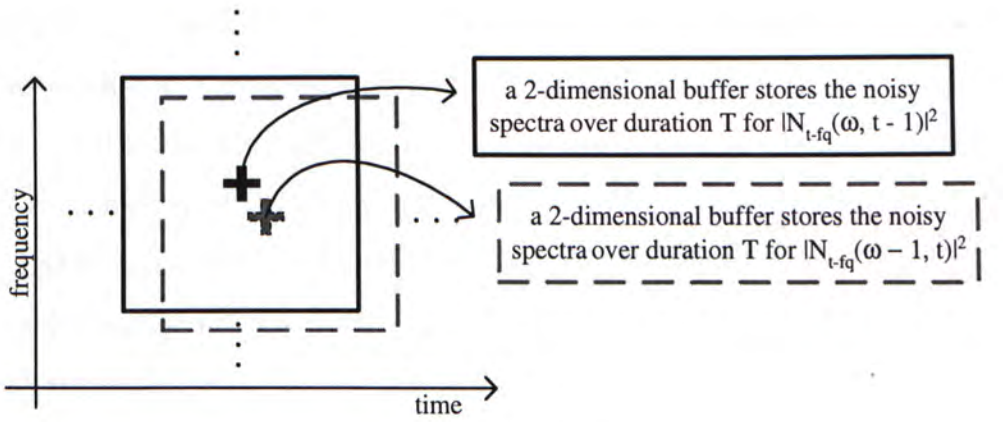


Figure 4.7: The buffer content to estimate $|N_{t-fq}(\omega, t)|^2$. The cross labels the current frequency ω and time t.

in the QBNE spectrum. Let ω_{p1}, ω_{p2} and *band* be the first harmonic frequency, second harmonic frequency and the distance between the two boundaries respectively. Figure 4.8 depicts the interpolation used in T-F QBNE.

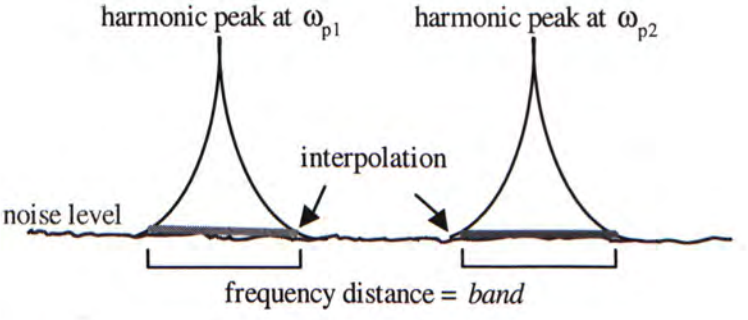


Figure 4.8: The interpolation used in T-F QBNE. The spectrum is the one estimated by QBNE. The boundaries are located at an equal distance from the harmonic frequency on each side.

4.2.3 Mainlobe-Resilient Time-Frequency Quantile-based Noise Estimation (M-R T-F QBNE)

By avoiding the adverse effects of speech power on noise estimation, T-F QBNE is expected to provide a more accurate estimation than QBNE. The major

difference between QBNE and T-F QBNE is the use of interpolation between the spectral values around the harmonic frequencies.

Recall that the interpolation boundaries are located at a constant distance from each harmonic frequency found in the QBNE spectrum. By observing the QBNE spectrum, it is found that the bandwidths of harmonic frequencies are substantially different. This observation is shown in Figure 4.9. The clean speech signal $x(t)$ is a synthetic speech generated by the source-filter model. The synthetic speech is generated by using a pitch value of 150 Hz. The formant frequencies are 700, 1220 and 2600 Hz and the corresponding bandwidths are 130, 70 and 160 Hz respectively. White noise is added to $x(t)$ to produce a SNR of 15 dB.

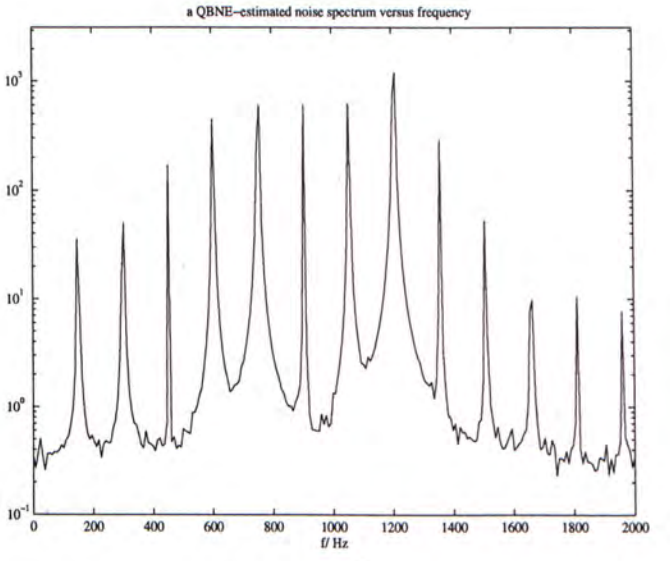


Figure 4.9: In a QBNE noise spectrum, the bandwidths of harmonic frequencies are different.

To have accurate interpolation, the bandwidths assigned for strong and weak peaks should be adjusted accordingly. When a strong peak roll-offs down to the noise level, the frequency distance is much greater than the one for the weak peak. The Mainlobe-Resilient Time-Frequency Quantile-based Noise Estimation (M-R T-F QBNE) proposes to give a larger *band* to strong harmonic peaks and a smaller *band* to weak harmonic peaks. As illustrated in Figure 4.10, it is

necessary to assign a larger *band* to stronger peaks. On the other hand, smaller *band* is also essential for weaker peaks to have better tracking of the noise level. Hence, the *band* is changed according to the strength of the harmonic peak, making the noise estimation resilient to the mainlobe height.

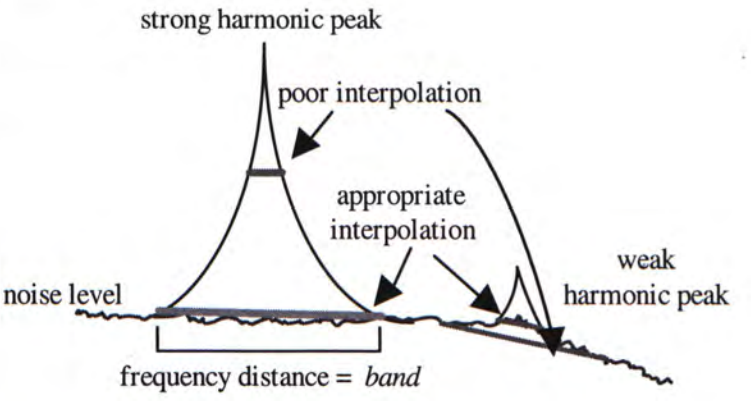


Figure 4.10: Different bandwidths are used for interpolation in M-R T-F QBNE.

The following gives a detailed description of the proposed M-R T-F QBNE. Let $y(t)$ be the noisy speech signal.

M-R T-F QBNE procedure

1. **cutting into frames**

   The noisy speech $y(t)$ is cut into overlapping frames. For every frame, do the following,

2. **QBNE**

   A coarse noise estimation is obtained by QBNE. Let T be the buffer duration used in QBNE. For frames located at the beginning and the end of an utterance, where the number of available frames is less than the total frame number in T, the buffer duration is reduced to store all the available frames only. Let $|N_q(\omega, t)|^2$ be the estimated noise power spectrum by QBNE.

3. **peak picking**

In order to apply interpolation around speech harmonic frequencies, the spectral peaks from speech signal are selected. This peak picking step is used to pick out all peaks from the QBNE noise spectrum. The QBNE spectrum is first smoothed by using a low-pass filter to remove small spikes. The low-pass filter used is a third order Butterworth filter with a cut-off frequency of 3200 Hz. The filtered spectrum is denoted by $|N_q(\omega, t)|^2$. It is assumed that there is no peak at d.c. or the frequency bin with the highest frequency.

Theoretically, a peak should have either zero or non-differentiable first derivative and negative second derivative. By using Taylor series expansion, differentiation can be approximated by second-order centered finite-divided-difference equations with high accuracy [42, 43]. The first derivative of a function $f(x)$ is calculated by,

$$\frac{df(x)}{dx} \approx \frac{-f(x+2\Delta x) + 8f(x+\Delta x) - 8f(x-\Delta x) + f(x-2\Delta x)}{12\Delta x} \tag{4.8}$$

where Δx is finite-divided-difference. The second derivative is found by,

$$\frac{d^2f(x)}{dx^2} \approx \frac{-f(x+2\Delta x) + 16f(x+\Delta x) - 30f(x) + 16f(x-\Delta x) - f(x-2\Delta x)}{12(\Delta x)^2} \tag{4.9}$$

The centered difference equations are used to have higher accuracy by incorporating more terms. The first-order first derivative and second derivative equations shown below can be used for boundary frequency bins if necessary,

$$\frac{df(x)}{dx} \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \tag{4.10}$$

$$\frac{d^2f(x)}{dx^2} \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2} \tag{4.11}$$

Any frequency bin that has a negative second derivative is recorded and consecutive frequency bins with negative second derivatives are grouped

together. For each group, the frequency bin with the smallest absolute first derivative is taken as a peak.

As the error of the numerical differentiation is proportional to the difference or the square of it, the number of FFT bins is increased.

4. **pitch estimation**

If a peak is from the speech harmonic structure, its location should be close to the multiples of the pitch frequency of the speech segment. Therefore, the pitch frequency is also estimated. A robust pitch extraction is employed by weighting the autocorrelation with the average magnitude difference function (AMDF) [44].

Let $\ddot{y}(t)$ be the windowed noisy speech segment using a Hamming window. The autocorrelation function $\phi(\tau)$ is defined by,

$$\phi(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} \ddot{y}(t)\ddot{y}(t+\tau) \tag{4.12}$$

where N, τ are the window length and the lag number. Let P be the period of the signal $\ddot{y}(t)$. Now, $\phi(0)$ has the largest magnitude among $\phi(\tau)$ and the second largest is given by $\phi(P)$. $\phi(\tau)$ has peaks at multiplies of P. In some cases, the peak located at $\tau = 2P$ may be larger than that at $\tau = P$ or there are some peaks at $\tau < P$, as shown in Figure 4.11. They are the so-called the half pitch error and the double-pitch error respectively. To avoid these errors, it was proposed to weight the autocorrelation function by an inverse AMDF.

The AMDF function is described by

$$\psi(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} |\ddot{y}(t) - \ddot{y}(t+\tau)| \tag{4.13}$$

When $\ddot{y}(t)$ is similar to $\ddot{y}(t+\tau)$, $\psi(\tau)$ becomes small. Hence, if $\ddot{y}(t)$ has a period of P, $\psi(\tau)$ has deep notches and the inverse of $\psi(\tau)$ produces peaks at multiplies of P.

As the noise included in $\phi(\tau)$ behaves differently with that included in $\psi(\tau)$, the error of pitch extraction from the AMDF-weighted autocorrela-

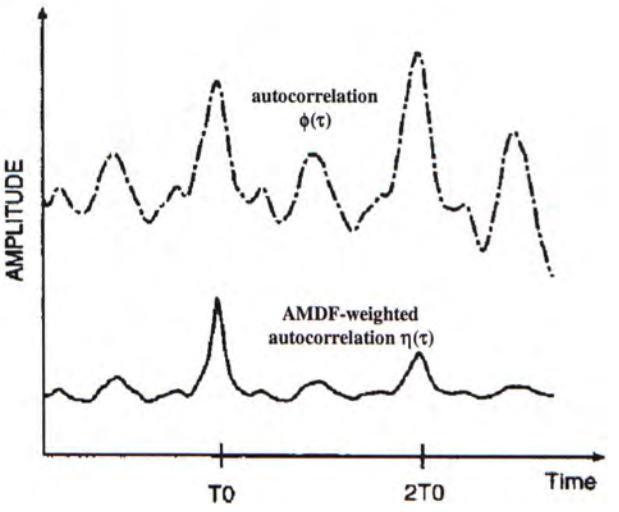


Figure 4.11: The autocorrelation function $\phi(\tau)$ and the AMDF-weighted autocorrelation function $\eta(\tau)$. T_o corresponds to the true pitch period.

tion function is expected to be reduced. The following function is used to extract the pitch value,

$$\eta(\tau) = \frac{\phi(\tau)}{\psi(\tau) + k} \tag{4.14}$$

where k is a positive constant to stabilize $\psi(\tau)$ when $\tau = 0$.

By searching the peak of the weighted function $\eta(\tau)$ from 50 Hz to 400 Hz, the pitch value of the speech segment is estimated. This range covers the region of the fundamental frequencies of most human speakers.

5. **decide if a peak comes from speech or not**

   This step is used to determine if a peak comes from speech harmonic spectrum or noise spectrum. With the pitch value and peak locations, a peak is assumed to be from speech harmonics if it is located around the pitch frequency within a small shift. This is shown in Figure 4.12.

6. **assign different *band* according to** $\log |N_q(\omega, t)|^2$

   The following applies only to the speech harmonic peaks. For all remaining frequencies, the noise estimate by ordinary QBNE is taken as the M-R T-F QBNE estimate. Let $|N_{M-Rt-fq}(\omega, t)|^2$ denote the noise estimate from M-R T-F QBNE.

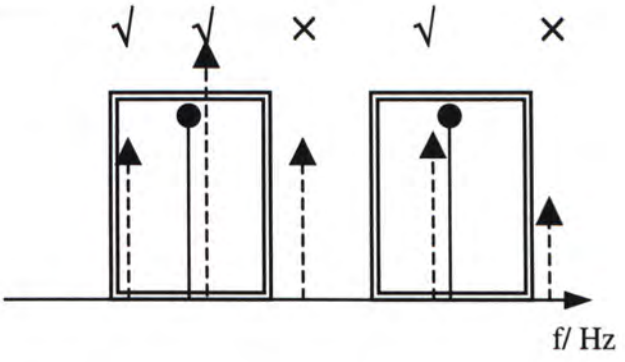


Figure 4.12: A peak is assumed to be from speech harmonic if it is enclosed by the rectangle. The stems and the arrows represent the harmonics and detected peak locations, respectively. The rectangles model the small shift region and the tick and cross above the figure show if a peak is a speech harmonic or not.

The log-scale dynamic range of peaks in each speech segment is divided into four equal portions. For each harmonic peak, one value is selected from the four possible *band* values. This value is used to define the distance between the two boundaries for interpolation. Equation 4.15 is used to assign this *band* value,

$$band = \begin{cases} band_1, & \alpha \leq \log|N_q(\omega_p,t)|^2 < \frac{\beta}{4} \\ band_2, & \frac{\beta}{4} \leq \log|N_q(\omega_p,t)|^2 < \frac{\beta}{2} \\ band_3, & \frac{\beta}{2} \leq \log|N_q(\omega_p,t)|^2 < \frac{3\beta}{4} \\ band_4, & \frac{3\beta}{4} \leq \log|N_q(\omega_p,t)|^2 \leq \beta \end{cases} \tag{4.15}$$

where $\log|N_q(\omega_p,t)|^2$, α and β are the current, minimum and maximum log power of harmonic peak, respectively.

The two interpolation boundaries are symmetrically located around the harmonic peak at a distance $band/2$.

7. **interpolation around speech harmonic peak**

   Finally, the noise estimate around speech harmonic peaks is found by linear interpolation.

By assigning different *bands* to harmonic peaks with various power, M-R T-F QBNE prevents the poor interpolation in strong harmonic peaks or inaccurate

tracking in weak peaks. The overall M-R T-F QBNE procedure is summarized in Figure 4.13.

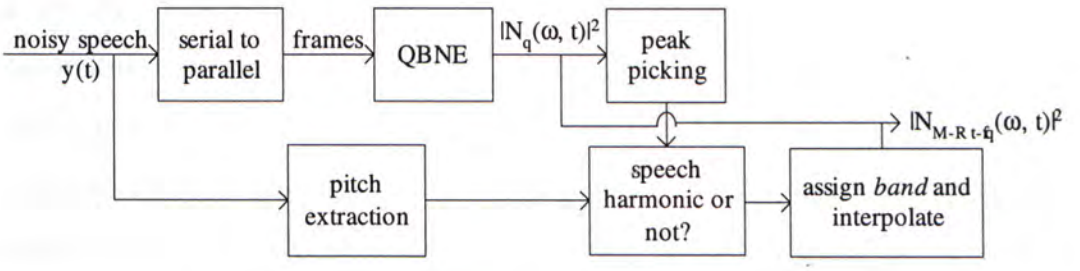


Figure 4.13: Block diagram of the M-R T-F QBNE.

4.3 Estimation Performance Analysis

The performance of M-R T-F QBNE is examined by using synthetic speech as well as real speech immersed in different background noise. There are two tests conducted in total.

The first test is about the mainlobe-resilient property of M-R T-F QBNE. It is used to study if M-R T-F QBNE prevents the overestimate of noise power from speech power at harmonic frequencies.

Synthetic speech segments are produced by the source-filter model [45]. Referring to Figure 2.5, there are two type of excitation. To synthesize voiced speech, an impulse train consisting of impulses at pitch frequency is used as the input to the filter. For unvoiced speech, a random noise-like input is used. A formant filter is a second-order recursive filter having the transfer function

$$H(z) = \frac{A}{1 - 2r\cos(\omega)z^{-1} + r^2 z^{-2}} \tag{4.16}$$

where A is a scaling constant, $b\omega$ is the formant bandwidth in radians, ω is the formant frequency (also in radians) and

$$r = e^{-(b\omega/2)} \tag{4.17}$$

Normally, the vocal tract filter is characterized by three or more pairs of formant frequencies and bandwidths. It is realized by cascading all formant filters.

Figure 4.14 shows the estimated noise spectra from several quantile-based methods. The clean speech signal is a synthetic speech with pitch frequency of 150 Hz. The formant frequencies are 700, 1220 and 2600 Hz and the corresponding bandwidths are 130, 70 and 160 Hz, respectively. White noise is added to produce a SNR of 15 dB. It is found that M-R T-F QBNE accurately estimates the noise spectrum, while the QBNE estimate is poor at speech harmonic frequencies. The estimate from T-F QBNE is also found to be affected when the speech harmonic power is high.

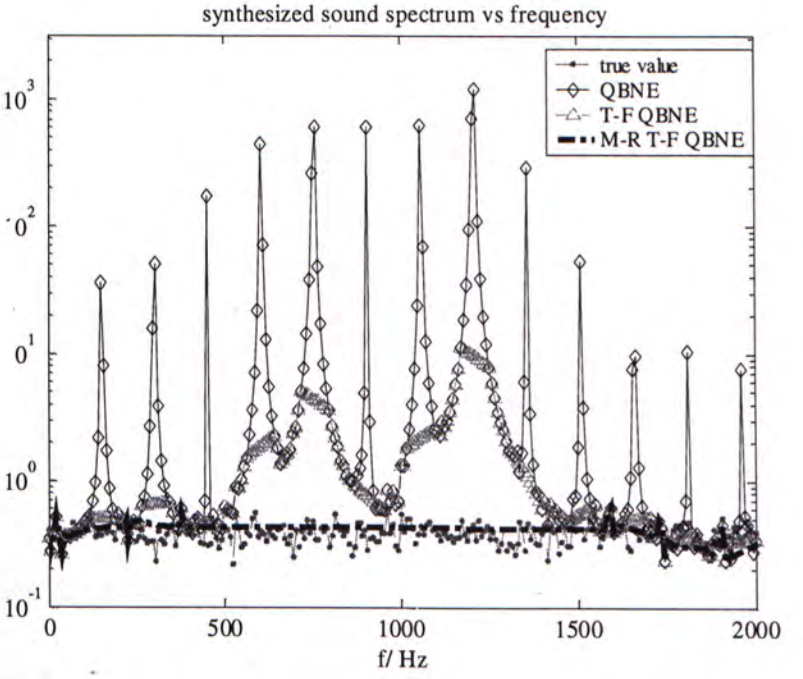


Figure 4.14: Estimated noise spectra of a synthesized speech segment. The true value refers to the exact noise spectrum found by periodogram.

The second test calculates the mean-square-error (MSE) in noise estimation from various methods. The noisy speech samples are the real speech from AURORA2 database. The true noise spectrum is found by subtracting the noisy speech waveform by the corresponding clean speech waveform and followed by

spectral estimation by a periodogram. Figure 4.15 is the plot of MSE of noise estimation versus SNR.

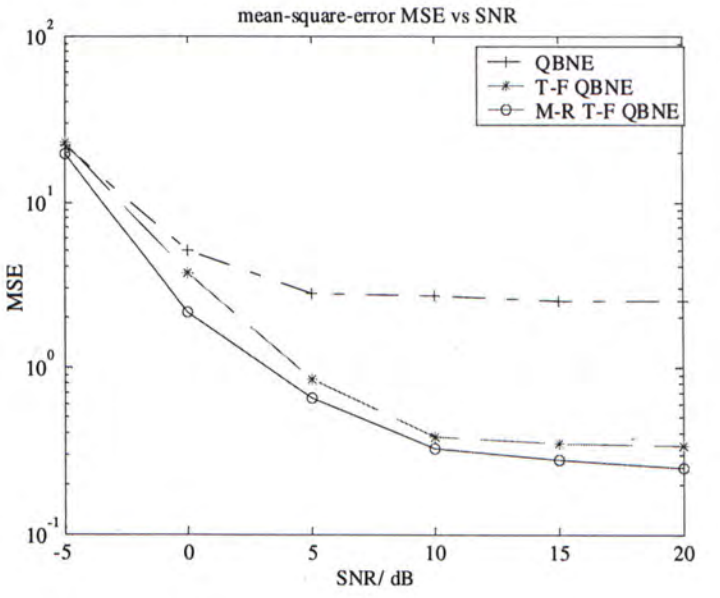


Figure 4.15: The MSE plot versus SNR.

Comparing with QBNE and T-F QBNE, M-R T-F QBNE achieves the lowest MSE at all SNR conditions. The improvement over the QBNE comes mainly from the utilization of spectral information at adjacent troughs (T-F QBNE) and M-R T-F QBNE further reduces the estimation error.

4.4 Recognition Experiment with Model Selection

A simple recognition framework with model selection capability is suggested in Section 3.3, which requires a noise estimation method. The proposed M-R T-F QBNE provides such a noise estimate to select the best matched model. Figure 4.16 gives a functional block diagram of the overall system.

After M-R T-F QBNE, the global signal-to-noise ratio SNR_g is calculated by

$$SNR_g = 10\log_{10}\{\frac{\sum_t \sum_\omega |Y(\omega,t)|^2}{\sum_t \sum_\omega |N_{M-Rt-fq}(\omega,t)|^2} - 1\} \qquad (4.18)$$

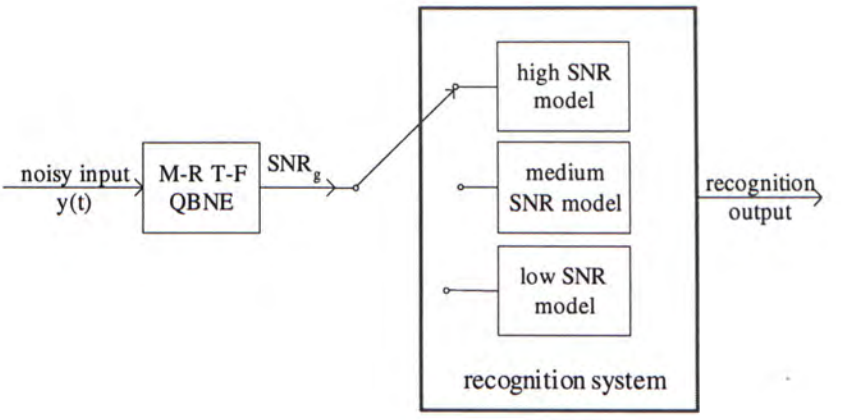


Figure 4.16: The block diagram of the recognition system with M-R T-F QBNE and model selection.

The acoustic models used are identical to the three acoustic models trained in Section 3.2.2. The best-matched model is chosen according to the SNR_g. If SNR_g is higher than or equal to 18 dB, the high SNR model is selected. If SNR_g is between 8 dB to 18 dB, the medium SNR model is used. Otherwise, the low SNR model is taken. This arrangement is used to align with the typical SNR range found by M-R T-F QBNE.

The recognition accuracy of the proposed recognition system is shown in Table 4.1. Table 3.3 (on page 45) represents the performance of model selection with known noise spectrum and Table 2.2 (on page 33) are the baseline results using clean data training. For comparison, the recognition accuracy of the multicondition training specified in AURORA2 corpus is given in Table 4.2. The number of utterances used in multicondition training is 8440. The recognition results of the four systems are listed again in Table 4.3 for the following comparison.

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.83	98.88	98.78	99.01	**98.88**
20	98.16	97.97	98.12	97.84	**98.02**
15	97.05	97.31	97.64	97.13	**97.28**
10	94.20	94.86	96.06	94.26	**94.85**
5	84.62	82.71	85.65	85.59	**84.64**
0	54.28	41.90	46.50	52.58	**48.81**
-5	24.44	1.57	18.70	25.86	**17.64**
average between 0 and 20 dB	85.66	82.95	84.80	85.48	**84.72**

test B

SNR/ dB	restaurant	street	airport	train-station	average
clean	98.83	98.88	98.78	99.01	98.88
20	97.67	97.67	97.35	97.32	97.50
15	95.86	96.74	96.51	95.93	96.26
10	91.68	93.74	80.79	92.59	92.62
5	76.85	80.62	80.79	78.62	79.22
0	37.12	46.25	44.44	40.05	41.96
-5	-8.01	16.29	7.19	11.97	6.86
average between 0 and 20 dB	79.83	82.93	82.39	80.90	81.51

test C

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	98.89	98.88	98.89
20	97.85	97.04	97.44
15	96.19	95.85	96.03
10	90.45	90.75	90.60
5	68.28	74.18	71.23
0	35.86	49.06	42.46
-5	17.29	20.62	18.95
average between 0 and 20 dB	77.73	81.38	79.55

Table 4.1: Word accuracy of the recognition system with M-R T-F QBNE and model selection.

test A in multicondition training

SNR/ dB	subway	babble	car	exhibition	average
clean	98.68	98.52	98.39	98.49	**98.52**
20	97.61	97.73	98.03	97.41	**97.69**
15	96.47	97.04	97.61	96.67	**96.94**
10	94.44	95.28	95.74	94.11	**94.89**
5	88.36	87.55	87.80	87.60	**87.82**
0	66.90	62.15	53.44	64.36	**61.71**
-5	26.13	27.18	20.58	24.34	**24.55**
average between 0 and 20 dB	88.75	87.95	86.52	88.03	**87.81**

test B in multicondition training

SNR/ dB	restaurant	street	airport	train-station	average
clean	98.68	98.52	98.39	98.49	98.52
20	96.87	97.58	97.44	97.01	97.22
15	95.30	96.31	96.12	95.53	95.81
10	91.96	94.35	93.29	92.87	93.11
5	83.54	85.61	86.25	83.52	84.73
0	59.29	61.34	65.11	56.12	60.46
-5	25.51	27.60	29.41	21.07	25.89
average between 0 and 20 dB	85.39	87.03	87.64	85.01	86.27

test C in multicondition training

SNR/ dB	subway(MIRS)	street(MIRS)	average
clean	98.50	98.58	98.54
20	97.30	96.55	96.92
15	96.35	95.53	95.94
10	93.34	92.50	92.92
5	82.41	82.53	82.47
0	46.82	54.44	50.63
-5	18.91	24.24	21.57
average between 0 and 20 dB	83.24	84.31	83.77

Table 4.2: Word accuracy of multicondition training system.

SNR/ dB	model selection with M-R T-F QBNE	model selection with known noise spectrum	baseline	multicondition training
clean	98.88	98.88	98.94	98.52
20	98.02	98.04	94.99	97.69
15	97.28	97.28	86.93	96.94
10	94.85	94.77	67.28	94.89
5	84.64	89.58	39.36	87.82
0	48.81	64.46	17.07	61.71
-5	17.64	28.64	8.40	24.55
average	84.72	89.83	61.13	87.81

Table 4.3: Average word accuracy of test set A from the four systems.

Regarding the overall average recognition accuracy with the known noise spectrum and the one from M-R T-F QBNE, both approaches outperform the baseline system in all test sets. For test set A, the M-R T-F QBNE system brings a 23.4% absolute improvement and a 60.5% relative improvement; while the system with known noise spectrum has a 28.5% absolute improvement and a 73.7% relative improvement.

Compared with the multicondition training, the model selection system with known noise spectrum has a promising performance in that the accuracy is even slightly higher and each acoustic model is trained with 20-40% of the original training size. There is nearly no degradation for high SNR inputs. This shows the great potential of the model selection capability for robust speech recognition. As the recognition result from the M-R T-F QBNE system is close to the one with known noise spectrum, it is concluded that M-R T-F QBNE works well with the model selection and improves the robustness of the speech recognition system. However, when the input SNR is extremely low (when *SNR* = 0, -5 dB), there is an apparent difference between the two system performance. The noise estimation from M-R T-F QBNE may not be accurate enough and that limits the improvement.

Note that the number of models in the proposed system is three only. It is expected that even if more models are used for selection, similar result will be achieved. This is because with the noise type and SNR-matching shown in Section 3.2.3, the recognition performance is highly similar to the system using SNR-matching only. Although the number of models in noise type and SNR-matching is 20 (many more models are used than SNR-matching), using three models only in SNR-matching provides similarly sufficient improvement in recognition.

The proposed recognition system with model selection is effective in noisy speech recognition and simple in implementation. It chooses from the available models one that best matches a given noisy speech. Only model selection is required after estimating SNR_g, skipping other computations that may appear in standard speech enhancement schemes. This robustness is believed to be the

consequence of matched conditions between training and testing. Nevertheless, even with known noise spectrum, the improvement from model selection for low SNR inputs is always smaller than the one for high SNR inputs. The accuracy at 0 dB is only 64%, whilst the accuracy at 20 dB is 98%. This is due to the phenomenon that at low SNRs, the models have large variances in MFCC [5]. The discriminability between recognition units is no longer as good as at high SNR conditions.

The optimum recognition system should have robust performance under various SNR conditions. Rather than making the model suitable for the input noisy speech, Chapter 5 proposes a new feature compensation method which converts noisy speech segments to the corresponding clean segments. It is expected that the high discriminability at high SNR conditions can be maintained.

Chapter 5

Feature Compensation: Algorithm and Experiment

In previous chapters, the reasons of the performance degradation in noisy speech recognition are analyzed in the view of matching between training and testing conditions. In this chapter, the degradation problem is investigated in terms of the deviation of noisy speech features from clean features. By making use of the deviation expression, an effective spectral compensation method is designed to approximate the clean speech feature. In the following, we will address the motivation and mathematical principles of the proposed system. At the end, we will show some recognition experiments.

5.1 Feature Deviation from Clean Speech

Recall that the speech recognition process generally consists of two parts, namely, front-end analysis and back-end decoding. When a clean speech signal is corrupted by background noise, the feature extracted from the noisy speech is expected to be different from the one of the corresponding clean speech. This discrepancy degrades the recognition performance. The following is intended to analyze how the features of noisy speech deviate from the clean features. The feature used is MFCC.

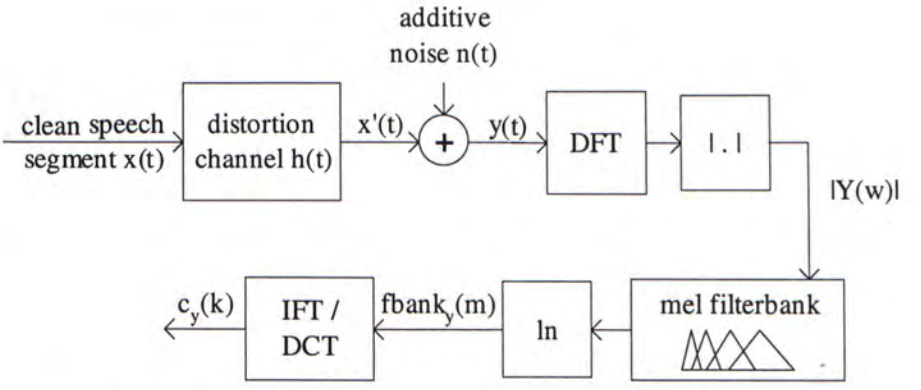


Figure 5.1: The signal model for features extracted from corrupted speech segments.

5.1.1 Deviation in MFCC Features

Figure 5.1 depicts the model used in the derivation. Both additive noise and channel distortion are encountered. Let $x(t)$, $x'(t)$ and $y(t)$ be the clean speech segment, the intermediate speech segment corrupted from distortion channel and the final noisy speech segment corrupted from both distortion channel and additive noise respectively.

$\mathscr{F}[\cdot]$ is used to denote the Discrete Fourier Transform (DFT) operation. The symbol $\circledast$ and $*$ represent the convolution operation and the conjugation, respectively. The channel distortion of $x(t)$ produces,

$$x'(t) = x(t) \circledast h(t)$$

and with additive noise,

$$y(t) = x'(t) + n(t) = x(t) \circledast h(t) + n(t) \tag{5.1}$$

$$x(t) \xrightarrow{\mathscr{F}} X(\omega)$$

$$y(t) \xrightarrow{\mathscr{F}} Y(\omega)$$

$$h(t) \xrightarrow{\mathscr{F}} H(\omega)$$

$$n(t) \xrightarrow{\mathscr{F}} N(\omega)$$

$$Y(\omega) = X(\omega)H(\omega) + N(\omega) \tag{5.2}$$

where $h(t)$ and $n(t)$ are the impulse response of the distortion channel and the

additive noise signal respectively.

$$\begin{aligned} P_y(\omega) &= |Y(\omega)|^2 = [X(\omega)H(\omega) + N(\omega)][X^*(\omega)H^*(\omega) + N^*(\omega)] \\ &= P_x(\omega)|H(\omega)|^2 + P_n(\omega) + 2Re\{X(\omega)H(\omega)N^*(\omega)\} \end{aligned} \tag{5.3}$$

where $P_y(\omega)$, $P_x(\omega) = |X(\omega)|^2$ and $P_n(\omega) = |N(\omega)|^2$ are the power spectra of $y(t)$, $x(t)$ and $n(t)$ respectively. $Re[\cdot]$ denotes the real part of $[\cdot]$.

Let $\alpha_{m\omega}$ denote the gain of filterbank m at frequency ω. The output of filterbank m is found by

$$fbank_y(m) = \ln\left\{\sum_\omega \alpha_{m\omega}|Y(\omega)|\right\} \tag{5.4}$$

Substituting Equation (5.3) to (5.4) gives

$$fbank_y(m) = \ln\left\{\sum_\omega \alpha_{m\omega}\sqrt{P_x(\omega)|H(\omega)|^2 + P_n(\omega) + 2Re\left[X(\omega)H(\omega)N^*(\omega)\right]}\right\} \tag{5.5}$$

Finally, the cepstral coefficient $c_y(k)$ is found by applying IFT on $fbank_y(m)$, that is,

$$\begin{aligned} c_y(k) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} fbank_y(m)e^{jkm}\,dm \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left\{\sum_\omega \alpha_{m\omega}\cdot \right. \\ &\qquad \left.\sqrt{P_x(\omega)|H(\omega)|^2 + P_n(\omega) + 2Re\left[X(\omega)H(\omega)N^*(\omega)\right]}\right\} e^{jkm}\,dm \end{aligned} \tag{5.6}$$

If the input speech segment $x(t)$ does not undergo any corruption, $y(t)$ is identical to $x(t)$, and

$$\begin{aligned} fbank_y(m) &= fbank_x(m) \\ &= \ln\left\{\sum_\omega \alpha_{m\omega}\sqrt{P_x(\omega)}\right\} \\ &= \ln\left\{\sum_\omega \alpha_{m\omega}|X(\omega)|\right\} \end{aligned} \tag{5.7}$$

$$\begin{aligned} c_y(k) &= c_x(k) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left\{\sum_\omega \alpha_{m\omega}\sqrt{P_x(\omega)}\right\} e^{jkm}\,dm \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left\{\sum_\omega \alpha_{m\omega}|X(\omega)|\right\} e^{jkm}\,dm \end{aligned} \tag{5.8}$$

where $fbank_x(m)$ and $c_x(k)$ denote the filterbank output and cepstral coefficient extracted from clean speech respectively.

If there exists additive noise only,

$$For\ all\ \omega, \quad H(\omega) = 1$$

$$fbank_y(m) = \ln\left\{\sum_{\omega} \alpha_{m\omega}\sqrt{P_x(\omega) + P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}\right\} \quad (5.9)$$

$$c_y(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left\{\sum_{\omega} \alpha_{m\omega} \cdot \sqrt{P_x(\omega) + P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}\right\} e^{jkm}\, dm \quad (5.10)$$

Equation (5.10) is used to study the recognition degradation due to additive noise. IFT is a bijective transformation that $fbank_y(m)$ can be uniquely found by knowing $c_y(k)$. Hence, we focus on the role of $N(\omega)$ on $fbank_y(m)$ in Equation (5.9).

Concerning how noise affects the filterbank output, there are two major observations. (1) Although the noise $n(t)$ is linearly added, the filterbank output $fbank_y(m)$ contains the noise terms in a non-linear expression which involves natural-log and square-root operations. Simple linear operations, such as adding a compensation term, cannot convert $fbank_y(m)$ back to the corresponding clean filterbank output. (2) In clean speech features, only the magnitude or power spectrum contributes to the filterbank output, as shown in Equation (5.7). However, for features extracted from noisy speech, both magnitude and phase spectra take part. To exactly recover the clean filterbank output from the noisy counterpart, it is necessary to know the complex noise spectrum $N(\omega)$.

5.1.2 Implications for Feature Compensation

From the derivation shown above, the noise effects can be compensated in several locations in the signal model illustrated in Figure 5.1. For instance,

1. **cepstral domain** $c_y(k)$

   Equation (5.10) is used. Using Taylor Series Expansion [42], the square-root term is approximated by,

$$\begin{aligned}
& \sqrt{P_x(\omega) + P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]} \\
= \; & \sqrt{P_x(\omega)}\sqrt{1 + \frac{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}{P_x(\omega)}} \\
\approx \; & |X(\omega)|\Bigg\{1 + \frac{1}{2}\frac{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}{P_x(\omega)} \\
& -\frac{1}{4}\cdot\frac{1}{2!}\{\frac{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}{P_x(\omega)}\}^2 + \\
& +\frac{3}{8}\cdot\frac{1}{3!}\{\frac{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]}{P_x(\omega)}\}^3 + \ldots\Bigg\}
\end{aligned} \tag{5.11}$$

   By further applying Taylor Series Expansion to the natural-log operation, $c_y(k)$ can be approximated by a linear combination of the term $\{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]\}/P_x(\omega)$.

   The noise corruption can be compensated by removing the IFT coefficients of all terms containing $\{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]\}/P_x(\omega)$. However, this requires the knowledge of the complex spectrum $N(\omega)$.

2. **filterbank domain** $fbank_y(m)$

   The compensation is highly similar to the case when it is performed in cepstral domain. All terms containing $\{P_n(\omega) + 2Re\left[X(\omega)N^*(\omega)\right]\}/P_x(\omega)$ after Taylor Series Expansion should be removed from $fbank_y(m)$.

3. **magnitude domain** $|Y(\omega)|$

   Both $P_n(\omega)$ and $2Re\{X(\omega)H(\omega)N^*(\omega)\}$ in Equation (5.3) should be subtracted to obtain the clean speech spectrum.

4. **just after the summation of** $x(t)$ **and** $n(t)$

   Actually, if the complex noise spectrum is available, the noise can be totally eliminated by subtracting $N(\omega)$ from $Y(\omega)$.

The compensation in all domains requires the knowledge of $N(\omega)$, both magnitude and phase. In the last two domains, only linear compensation is

involved and no approximation is taken. As a result, it is much simpler and more accurate to do compensation in either the time or frequency domain.

5.2 Overview of Conventional Compensation Methods

In this section, a number of compensation methods are reviewed. Basically, they can be classified into three groups, namely, speech enhancement, feature compensation and model-based adaptation. Speech enhancement and feature compensation approximate the clean signal or features from the noisy speech by reducing the noise contents in a certain domain. The noise is cleaned up prior to the speech recognition system. The analysis delivered in Section 5.1 uses this approach. These two families are highly similar, they only differ from the other in the input-output relationship. Speech enhancement has a noisy speech signal $y(t)$ as input and outputs a noise-reduced cleaner speech signal. The output is regarded as an approximation of the clean speech signal $x(t)$. Typical examples of speech enhancement methods include Spectral Subtraction, Wiener Filtering and Blind Source Separation [8, 10, 46]. Feature compensation accepts any features extracted from noisy speech as input. The output is the modified features which may not be in the same domain as input features. For example, it is possible to have a feature compensation method which has noisy speech $y(t)$ as input and MFCC $c_y(k)$ as output.

The following discusses some compensation methods and their pros and cons.

- **Weighted Filter Bank Analysis** Filterbank analysis is one of the most extensively employed spectral analysis techniques in ASR. By using a bank of highly overlapped bank-pass filters, the short-time spectral envelope of a input speech segment can be obtained. This measured spectral envelope is often sensitive to background noise. It was found that noise is perceptually more tolerated in the spectral formant regions than in the spectral valleys. The weighted filter bank analysis method emphasizes the

high energy parts of the log filterbank energies such that the cepstral coefficients become less susceptible to the noise [47, 48]. Similar ideas were proposed in [49, 50].

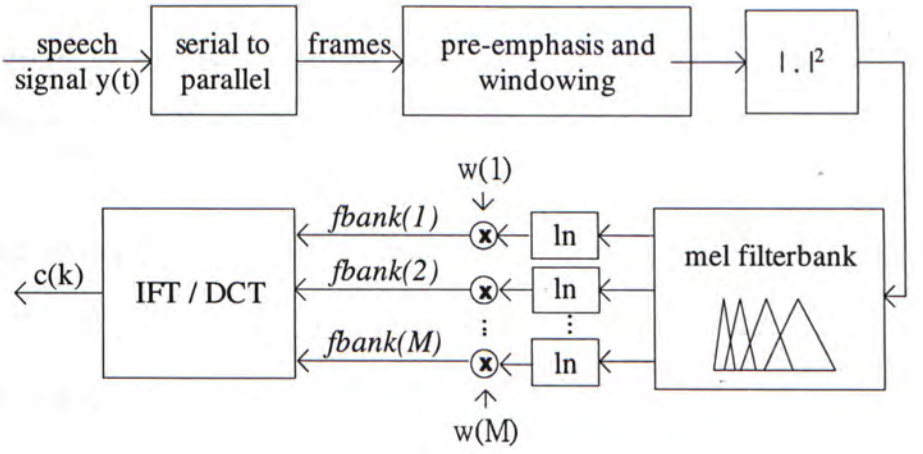


Figure 5.2: Block diagram of the weighted filter bank analysis.

Referring to Figure 5.2, let there be M filter banks in total. Let $w(m)$ denote the weighting factor for filter bank m. The log filterbank energies are multiplied by a set of weighting factors before IFT.

Before weighting,

$$fbank(m) = \ln\left\{\sum_{\omega} \alpha_{m\omega} P_y(\omega)\right\} \tag{5.12}$$

After weighting,

$$fbank(m) = w(m) \cdot \ln\left\{\sum_{\omega} \alpha_{m\omega} P_y(\omega)\right\} \tag{5.13}$$

The weighting factor $w(m)$ is related to the SNR of the frame by,

$$w(m) = \beta_m / \sum_{j=1}^{M} \beta_j \tag{5.14}$$

$$\beta_m = \sum_{r=1}^{M} \left\{\frac{\ln[\sum_{\omega} \alpha_{m\omega} P_y(\omega) + 1]}{\ln[\sum_{\omega} \alpha_{r\omega} P_y(\omega) + 1]}\right\}^{\frac{1}{F(SNR_t)-1}} \tag{5.15}$$

where SNR_t and $F(SNR_t)$ denote the frame-based SNR value and a function of SNR_t respectively.

$F(SNR_t)$ is a linear relationship between SNR_t and the function output. The function output is always positive and bounded between F_{min} and F_{max}. For low SNR_t, the function output is closed to F_{min}; for high SNR_t, the function output is near F_{max}. When $F(SNR_t)$ tends to 1 and filterbank m has the highest energy, $w(m) = 1$ and $w(j) = 0$ for $m \neq j$. When $F(SNR_t) \rightarrow \infty$, all $w(m)$ are equal to $1/M$.

The experimental results reported in [47, 48] show the use of weighted filterbank energies provide moderate improvement in medium to high SNR conditions.

- **Spectral Subtraction** Spectral Subtraction (SS) [8] was developed by Boll in 1979. It has been widely used for speech enhancement and robust speech recognition, which may be due to its simple computation. As the noise is additively mixed with the speech signal, we have,

$$y(t) = x(t) + n(t) \tag{5.16}$$

Taking the autocorrelation at both sides, assume speech is uncorrelated with the noise, the autocorrelation function is

$$\begin{aligned} R_y(\tau) &= E\{[x(t) + n(t)][x(t+\tau) + n(t+\tau)]\} \\ &= R_x(\tau) + R_n(\tau) \end{aligned} \tag{5.17}$$

where $R_x(\tau)$ and $R_n(\tau)$ are the autocorrelation function of the speech signal and noise signal respectively.

Taking the Fourier Transform at both sides,

$$\begin{aligned} P_y(\omega) &= P_x(\omega) + P_n(\omega) \\ |Y(\omega)|^2 &= |X(\omega)|^2 + |N(\omega)|^2 \\ |X(\omega))|^2 &= |Y(\omega)|^2 - |N(\omega)|^2 \end{aligned} \tag{5.18}$$

The output enhanced speech $\hat{X}(\omega)$ is found by,

$$\hat{X}(\omega) = [|Y(\omega)|^2 - |N(\omega)|^2]^{\frac{1}{2}} e^{j\angle Y(\omega)} \tag{5.19}$$

where $\angle Y(\omega)$ is the phase spectrum of $y(t)$. If the difference $|Y(\omega)|^2 - |N(\omega)|^2$ is negative, it is set to 0. Although this method reduces the noise, it usually introduces an annoying musical noise. To further reduce background noise and eliminate musical noise, oversubtraction and spectral floor are adopted [9, 13]. The substraction is modified as,

$$D(\omega) = |Y(\omega)|^2 - \alpha|N(\omega)|^2 \tag{5.20}$$

$$|X(\omega)|^2 = \begin{cases} D(\omega), & D(\omega) \geq \beta|N(\omega)|^2 \\ \beta|N(\omega)|^2, & otherwise \end{cases} \tag{5.21}$$

$$\alpha \geq 1 \tag{5.22}$$

$$0 \leq \beta \leq 1 \tag{5.23}$$

where α is the oversubtraction factor and β is the spectral floor parameter, which normally take values of 1 to 4 and 0.005 to 0.06 respectively. The result $|X(\omega)|$ is either used for reconstructing the output speech or input to standard feature extraction process.

- **Missing Data Theory** In missing data theory [32], time-frequency regions which carry reliable speech information are identified. Unreliable data are treated as missing. Recognition is then based on the reliable regions alone.

  To locate reliable regions, SNR-related criteria are often used. For example, data is considered as missing if

$$|\hat{X}(\omega)|^2 < \frac{1}{2}|Y(\omega)|^2 \tag{5.24}$$

  or

$$|Y(\omega)| - |\hat{N}(\omega)| < 0 \tag{5.25}$$

  where $|\hat{N}(\omega)|$ and $|\hat{X}(\omega)|^2$ are the estimated noise power spectrum and clean speech power spectrum, respectively.

  Figure 5.3 shows the spectrogram and the identified reliable regions from a spoken digit sequence. The utterance is corrupted by factory noise at 10

(a) Spectrogram

(b) Reliable regions identified

Figure 5.3: Reliable regions identified from a noisy speech corrupted with factory noise at 10 dB.

dB. The reliable regions found are close to the desired speech spectrogram as shown.

Missing data theory is a special case under the speech enhancement methods. No speech signal or feature is generated at the end, but only information about reliable speech regions is extracted. The information contents are actually similar to those obtaining from other enhancement schemes or compensation methods, where the estimated clean speech spectrum is the so-called reliable region.

- **Blind Source Separation** The problem of separating the desired speech from interfering sources, the so-called cocktail party effect, has been a popular research area recently. Blind source separation (BSS) assumes no information about the mixing process or the sources, apart from their mutual statistical independence. Among various techniques solving this BSS problem, Independent Component Analysis (ICA) [46, 51, 52] is a method which estimates a set of linear filters to separate the mixed signals under the assumption that the original sources are statistically independent.

  Although BSS is capable of solving such complicated scenarios, the system requirement is high. In particular, the number of microphones required must be higher than or equal to the number of speakers. For our single-channel connected digit recognition task, BSS cannot be used.

Among the compensation methods addressed, spectral subtraction, weighted filterbank analysis and missing data theory are closely related to each other. They share the same underlying principle that only spectra or features with high SNR are left at the end.

If there is a mismatch between training and testing conditions, it is sensible to retrain the acoustic models. It is always desirable to adapt the acoustic models given a relatively small amount of speech from the new environment. This is done in practice for telephone speech where only telephone speech is used during training [14]. No clean and high-bandwidth speech is involved. Model-

based adaptation modifies the acoustic models used inside the back-end decoder to adapt to the input noisy speech, by using statistics on noise or noisy speech [14, 53]. The most straightforward method is to re-train the whole acoustic model with speech in the new environment. The recognition system with model selection proposed in Chapter 3 is also under this type. The following describes a classical example of model-based adaptation methods, which is parallel model combination (PMC).

- **Parallel Model Combination (PMC)** By using the acoustic models trained with clean speech and a noise model, the distributions of corrupted speech can be approximated. This approach saves much computation, as the approximation is done on model-level, where the whole set of training data is not required on-line. This is the idea behind parallel model combination (PMC). PMC assumes the distribution of clean speech and noise is a mixture of Gaussians and further uses distribution of mixtures of Gaussian to represent the distribution for noisy speech. Figure 5.4 illustrates how PMC obtains the noisy speech distribution with the two separated models.

  Assume that the feature vector is in the MFCC representation. The clean speech model and noise model are first transformed back to the log filterbank domain by using DFT or inverse discrete cosine transform (IDCT). Then the corresponding spectral values are found with an exponential operation. The resultant models are in linear spectral domain at this moment. They are combined by simply adding them together to generate the distribution for noisy speech. Finally, the distribution is converted back to the cepstral domain by following the standard feature extraction process.

  PMC generally provides satisfactory performance of noisy speech recognition. For stationary or slow-varying noises, only little computation cost is incurred. For fast-changing noises, PMC is computationally expensive. It is assumed that both the clean speech and noise are normal-distributed and their power spectra are log-normal. After the combination, the sum

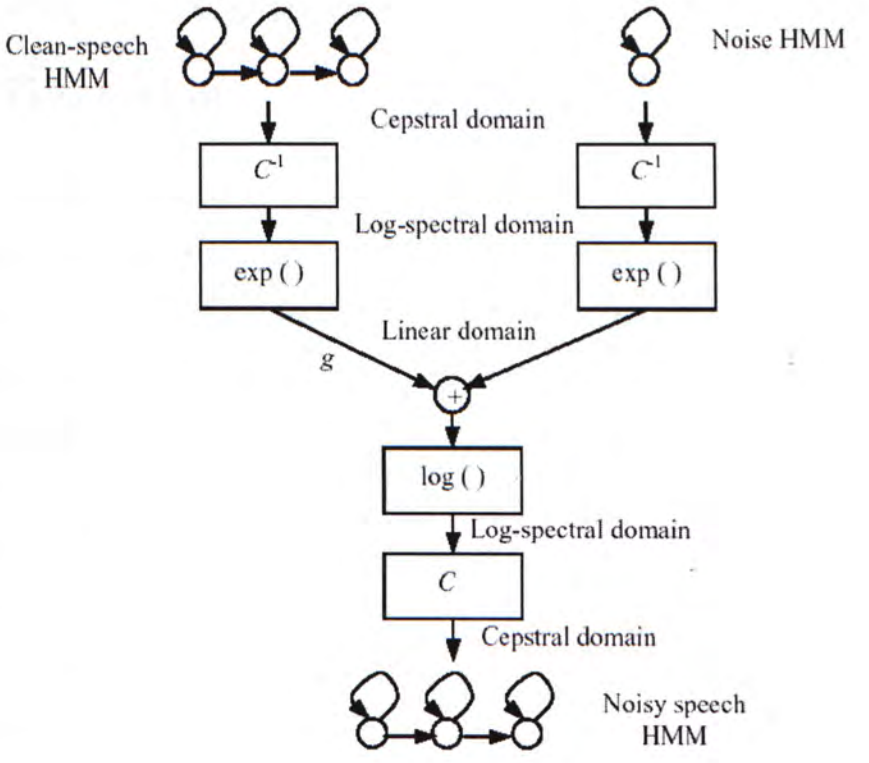


Figure 5.4: Block diagram of parallel model combination.

is also modelled by a log-normal distribution. Nevertheless, the sum of two log-normal distributions is no longer log-normal.

For model-based adaptation methods, the recognition rate for clean speech is often sacrificed for the improvement of recognition at low SNRs. This is because the model discriminability is decreased as a consequence of the large distribution variance, after incorporating the noise model. To have satisfactory recognition under various SNR conditions, feature compensation is adopted for our recognition system. In addition, spectral subtraction is used as a benchmark for evaluating the proposed method.

5.3 Feature Compensation by In-phase Feature Induction

A spectral feature compensation method called In-phase Feature Induction (IFI) is proposed to improve the robustness of ASR. IFI accurately obtains the corresponding clean speech features from a noisy speech corrupted by additive noise. By converting to clean speech features and keeping the high discriminability, it is designed to give a much better improvement than the recognition system with model selection proposed in Section 3.3.

The compensation problem is first reformulated and particular interests have been put on the phase difference between complex spectra of noisy input and interfering noise. This leads to a reasonable assumption for this phase difference and gives an accurate spectral estimation. From the approximately-clean spectrum, the recognition degradation under low SNRs is significantly reduced. In this section, we will address the deficiency of Spectral Subtraction, describe the motivation of this new compensation method and compare with other studies. Afterwards, the details of mathematical framework will be shown.

5.3.1 Motivation

Among various feature compensation methods, Spectral Subtraction (SS) has been widely used for both speech enhancement and robust speech recognition. SS requires simple computation only and keeps high discriminability between recognition units. However, SS is unable to derive the exact clean speech spectrum, even if the noise magnitude spectrum is known.

If the noise magnitude spectrum is known a priori, the average word error rate of a digit string recognizer ranges from 11% to 88% for different tuning parameters, such as the spectral floor or oversubtraction factor [12]. It is because the phase relationship or the correlation between spectra of clean speech and interfering noise is neglected. These limit the usage of SS on ASR, especially when the noise estimation is not accurate enough. In the following, the mathematical details of SS are reviewed.

For

$$y(t) = x(t) + n(t) \tag{5.26}$$

then

$$\begin{aligned} R_y(\tau) &= E\Big\{[x(t)+n(t)][x(t+\tau)+n(t+\tau)]\Big\} \\ &= E[x(t)x(t+\tau)] + E[n(t)n(t+\tau)] + E[x(t)n(t+\tau)] + E[n(t)x(t+\tau)] \\ &= R_x(\tau) + R_n(\tau) + E[n(t)x(t+\tau)] + E[x(t)n(t+\tau)] \end{aligned} \tag{5.27}$$

In SS, it is assumed speech is uncorrelated with noise, so that

$$R_y(\tau) = R_x(\tau) + R_n(\tau) \tag{5.28}$$

$$|Y(\omega)|^2 = |X(\omega)|^2 + |N(\omega)|^2 \tag{5.29}$$

This assumption is only true when a sufficient number of samples is available in $x(t)$ and $n(t)$. Within a typical frame duration, the number of samples is 160 - 480 (20 ms with sampling frequency 8 kHz to 30 ms with sampling frequency 16 kHz). Due to the small number of samples, even if the speech and the noise are uncorrelated, the numerical values computed for $E[n(t)x(t+\tau)]$ and $E[x(t)n(t+\tau)]$ in Equation (5.27) are non-zero. Hence, both (5.28) and (5.29) are not accurate representations when the sample size is small.

Taking the Fourier Transform of both sides in Equation (5.26), we have

$$Y(\omega) = X(\omega) + N(\omega) \tag{5.30}$$

$$\begin{aligned} |Y(\omega)|^2 &= [X(\omega)+N(\omega)][X^*(\omega)+N^*(\omega)] \\ &= |X(\omega)|^2 + |N(\omega)|^2 + 2Re\,[X(\omega)N^*(\omega)] \end{aligned} \tag{5.31}$$

It is seen that, the cross-term $2Re\,[X(\omega)N^*(\omega)]$ is omitted in Equation (5.29). To accurately restore the clean speech spectrum, this cross-term accounts for the non-zero correlation between $x(t)$ and $n(t)$ (equivalent to $\mathscr{F}\{E[n(t)x(t+\tau)] + E[x(t)n(t+\tau)]\}$), which is highly essential for the accurate estimation of $|X(\omega)|^2$. This explains why SS cannot derive the exact clean speech spectrum, even if the noise power spectrum is known a priori.

The term $2Re\left[X(\omega)N^*(\omega)\right]$ not only bears the power spectrum of $x(t)$ and $n(t)$, but also the information of the phase spectra. Its magnitude depends on $|X(\omega)|$ and $|N(\omega)|$ and their angles. Sometimes, it can be as large as the speech power; at another instant, it can be zero. In most cases, only the power spectra are estimated and the phase spectra are unknown, making the exact restoration difficult.

To overcome this problem, a previous research study [54] proposed a method called smoothing of time direction. This method is built on top of SS. It considers the average of noisy speech power spectra over a short period of time as the estimated noisy speech power spectrum for current time index, so as to reduce the influence of the cross-term $2Re\left[X(\omega)N^*(\omega)\right]$. The noisy speech power spectrum is expressed in terms of,

$$d = 0, 1, \ldots, D-1 \tag{5.32}$$

$$\sum_d \beta_d = 1 \tag{5.33}$$

$$\overline{|Y(\omega,t)|^2} = \sum_d \beta_d |Y(\omega, t-d)|^2 \tag{5.34}$$

where D and β_d are the number of frames for averaging and the weighting factor for $|Y(\omega, t-d)|^2$ respectively.

Assuming the speech and noise are stationary within the period D, substituting Equation (5.31) into (5.34) gives

$$\begin{aligned}\overline{|Y(\omega,t)|^2} = & \sum_d \beta_d |X(\omega,t-d)|^2 + \sum_d \beta_d |N(\omega,t-d)|^2 \\ & + \sum_d \beta_d 2Re\left[X(\omega,t-d)N^*(\omega,t-d)\right]\end{aligned} \tag{5.35}$$

$$\sum_d \beta_d |X(\omega,t-d)|^2 \approx |X(\omega,t)|^2 \tag{5.36}$$

$$\sum_d \beta_d |N(\omega,t-d)|^2 \approx |N(\omega,t)|^2 \tag{5.37}$$

$$\begin{aligned}\overline{|Y(\omega,t)|^2} \approx & |X(\omega,t)|^2 + |N(\omega,t)|^2 \\ & + \sum_d \beta_d 2Re\left[X(\omega,t-d)N^*(\omega,t-d)\right]\end{aligned} \tag{5.38}$$

The cross-term $2Re\left[X(\omega,t-d)N^*(\omega,t-d)\right]$ of successive frames is assumed to be independent of each other. This leads to,

$$\sum_d \beta_d 2Re\left[X(\omega,t-d)N^*(\omega,t-d)\right] \approx 0 \tag{5.39}$$

Equation (5.38) becomes,

$$\overline{|Y(\omega,t)|^2} \approx |X(\omega,t)|^2 + |N(\omega,t)|^2 \tag{5.40}$$

Finally, $\overline{|Y(\omega,t)|^2}$ is used to replace $|Y(\omega)|^2$ in Equation (5.20) for noise reduction.

This smoothing method uses a smoothed noisy spectrum and reduces the correlation between the speech signal and the noise. However, besides the over-subtraction factor and noise floor, there are two other parameters required to be fine-tuned for proper averaging. They are β_d and D. In [54], it was reported that the optimal value of D varies under different SNR conditions.

An alternative approach is to apply a low-pass filter on the SS output $|X(\omega)|^2$, so as to reduce the errors made during subtraction [55].

5.3.2 Methodology

The failure of SS for noise compensation is due to the improper use of $|N(\omega)|^2$ and the coarse estimate of it. In previous section, the importance of the cross-term $2Re\left[X(\omega)N^*(\omega)\right]$ is discussed. The following describes the proposed In-phase Feature Induction (IFI) method and illustrates how a better utilization of $|N(\omega)|^2$ benefits spectral estimation and noisy speech recognition.

Note that the complex spectra of $y(t)$, $x(t)$ and $n(t)$ are related by,

$$|X(\omega)|^2 = |Y(\omega)|^2 - |N(\omega)|^2 - 2Re\left[X(\omega)N^*(\omega)\right] \tag{5.41}$$

Within a frame, which is a short period of time, the removal of the critical cross-term $2Re\left[X(\omega)N^*(\omega)\right]$ is inappropriate and results in a poor estimate of

$|X(\omega)|^2$. On the contrary, we suggest the following reformulation,

$$\begin{aligned}
|X(\omega)|^2 &= |Y(\omega)|^2 - |N(\omega)|^2 - 2Re\left[X(\omega)N^*(\omega)\right] \\
&= |Y(\omega)|^2 - |N(\omega)|^2 - 2Re\Big\{\left[Y(\omega) - N(\omega)\right]N^*(\omega)\Big\} \\
&= |Y(\omega)|^2 + |N(\omega)|^2 - 2Re\left[Y(\omega)N^*(\omega)\right] \\
&= |Y(\omega)|^2 + |N(\omega)|^2 \\
&\quad -2|Y(\omega)N(\omega)|\cos\big\{\angle Y(\omega) - \angle N(\omega)\big\} \qquad (5.42)
\end{aligned}$$

where $\angle Y(\omega)$ and $\angle N(\omega)$ are the phase spectra of $y(t)$ and $n(t)$ respectively. Figure 5.5 depicts the plot of the phase difference $\angle Y(\omega) - \angle N(\omega)$, together with the corresponding cosine values of a SNR 10 dB noisy speech. For illustration purposes, the respective clean speech speech is also shown.

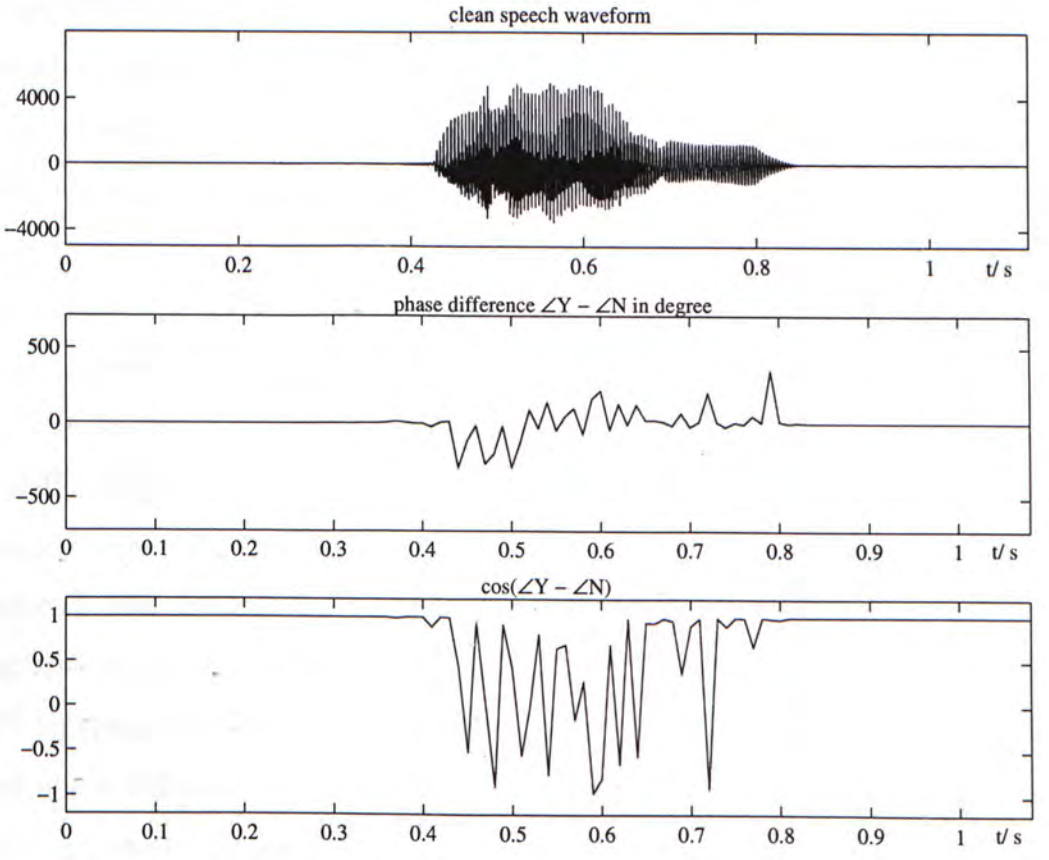


Figure 5.5: Plots of the phase difference and the corresponding cosine values versus time.

During non-speech periods, the phase difference $\angle Y(\omega) - \angle N(\omega)$ is always negligible, since $Y(\omega) \approx N(\omega)$. Thus, we assume $Y(\omega)$ and $N(\omega)$ are always

in-phase and let the phase difference be 0. Then,

$$\begin{aligned} |X(\omega)|^2 &= |Y(\omega)|^2 + |N(\omega)|^2 - 2|Y(\omega)N(\omega)|\cos(0) \\ &= |Y(\omega)|^2 - 2|Y(\omega)N(\omega)| + |N(\omega)|^2 \\ &= \Big[|Y(\omega)| - |N(\omega)|\Big]^2 \end{aligned} \tag{5.43}$$

Equation (5.43) gives the essence of the proposed method. IFI is different from SS in that speech signal is not necessarily reconstructed, but the spectral features are well compensated by the reformulation and the phase (cross-term) contribution.

While the in-phase assumption between $Y(\omega)$ and $N(\omega)$ is highly accurate in non-speech periods, $\angle X(\omega) - \angle N(\omega)$ is always unknown. Therefore, the phase difference $\angle Y(\omega) - \angle N(\omega)$ is used, instead of $\angle X(\omega) - \angle N(\omega)$ shown in Equation (5.41).

Comparing with the smoothing technique described in Section 5.3.1, IFI is supported by both mathematical derivation and the accurate in-phase relationship between $Y(\omega)$ and $N(\omega)$. The averaging operation is only a mean to reduce the influence of the cross-term $2Re\,[X(\omega)N^*(\omega)]$, but IFI directly manipulates it to improve the accuracy of spectral estimation.

The following attempts to compare and contrast the results attained by SS and IFI. Figure 5.6 gives the compensated results using exact $|N(\omega)|^2$ at frequency around 938 Hz for the same noisy speech shown in Figure 5.5. The speech signal presents from 0.4 s to 0.8 s. Both SS and IFI perform well during speech period and reduce the background noise level. It is observed that IFI compensation greatly outperforms SS in non-speech periods. This shows the phase difference of the noisy spectrum $\angle Y(\omega) - \angle N(\omega)$ and the in-phase assumption are critical.

To study whether IFI and SS is sensitive to the noise power, the IFI compensated magnitudes from different SNR inputs are plotted in Figure 5.7(a). Figure 5.7(b) shows the compensated magnitudes from SS with the same set of inputs.

Comparing the compensated magnitudes by the two methods, IFI has reli-

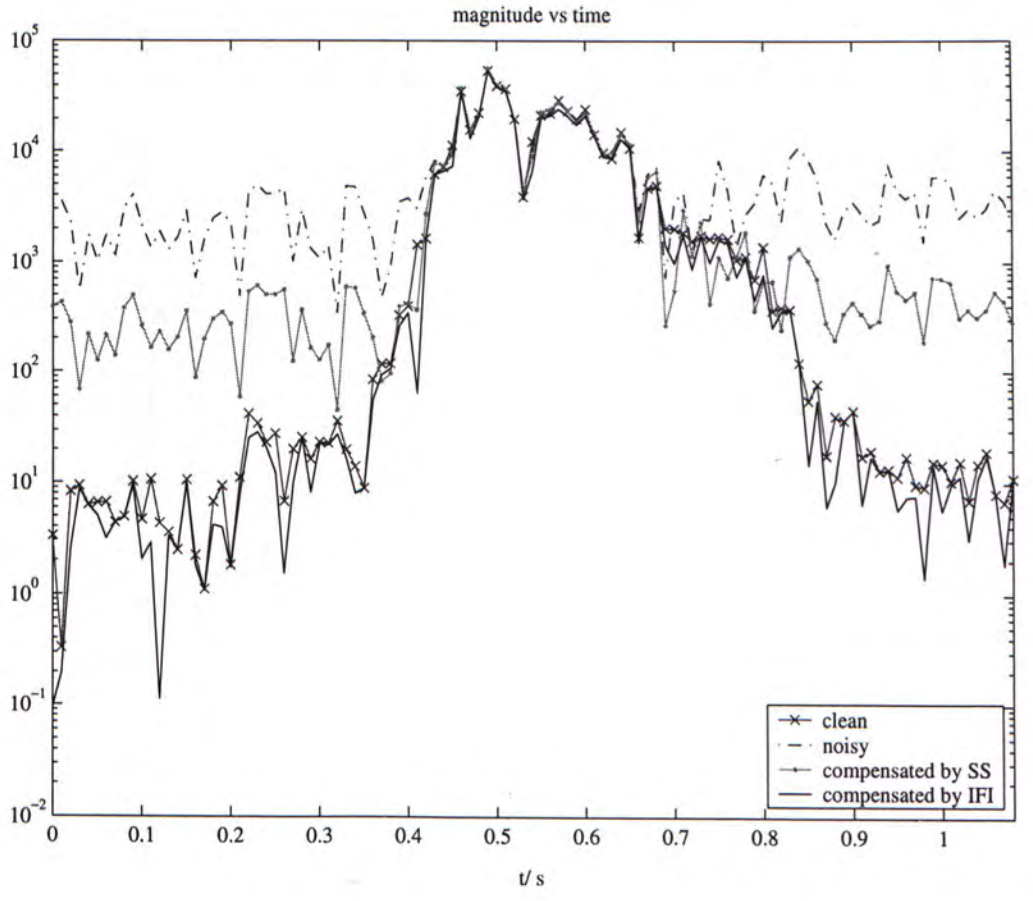
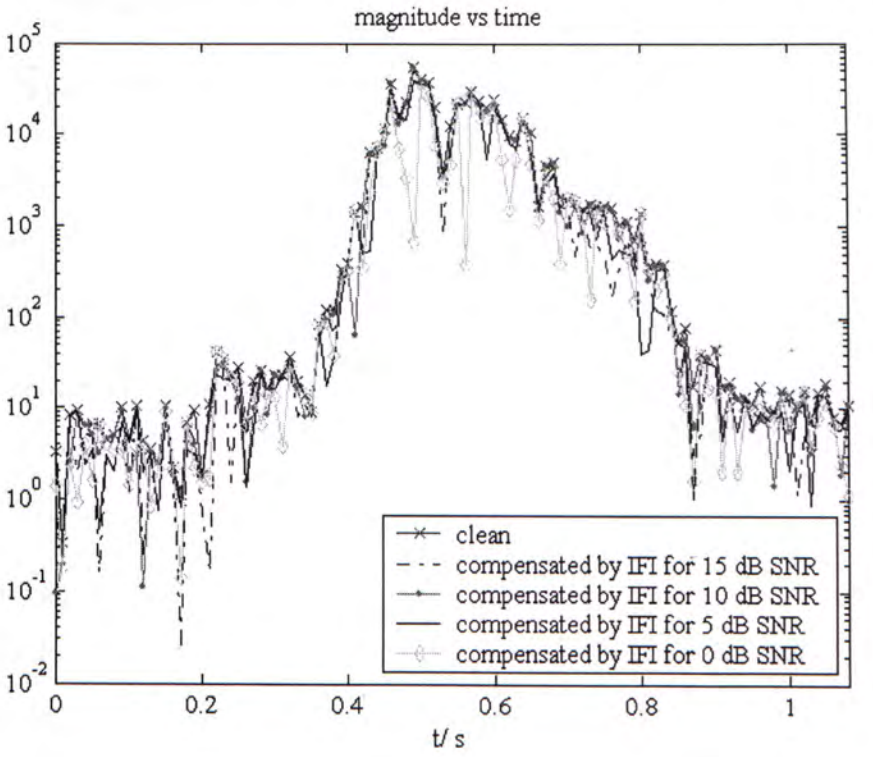
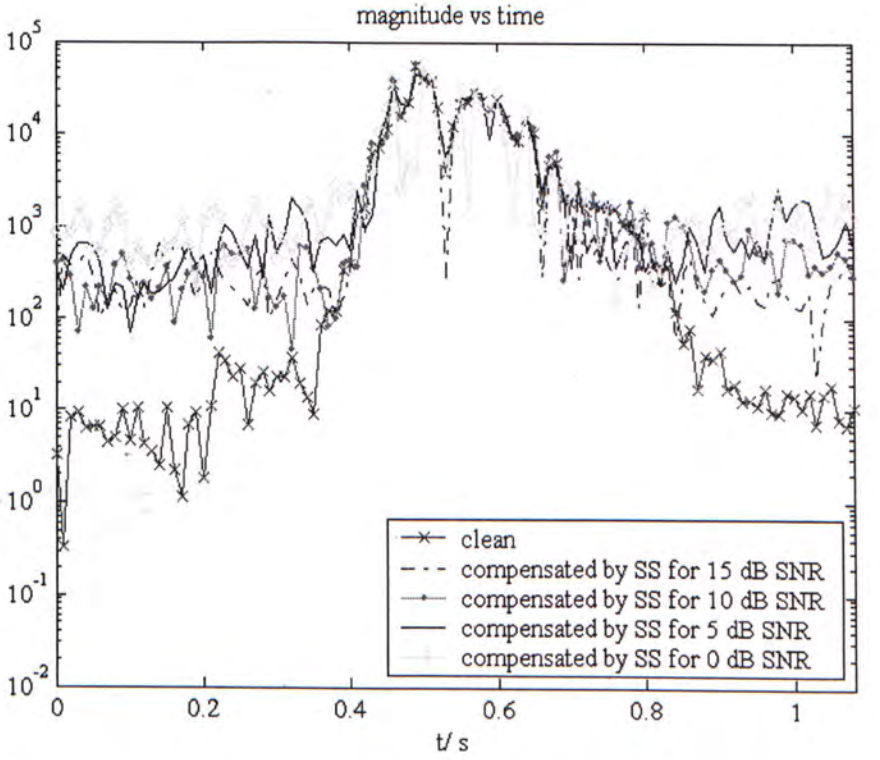


Figure 5.6: Magnitude versus time from the clean speech, noisy speech, SS-compensated speech and IFI-compensated speech.

able compensation at all SNRs and similar performance is observed, no matter how strong the noise is. For SS, residual noise is found in the compensated magnitude, especially during non-speech periods. When the SNR decreases, the estimated clean magnitude spectrum is mostly found by the noise floor and residual noise remains substantial.



(a) IFI



(b) SS

Figure 5.7: Magnitude versus time at different SNRs.

5.4 Compensation Framework for Magnitude Spectrum and Segmental Energy

To evaluate the performance of IFI for recognition, a front-end compensation framework is suggested below. The compensation process is carried out in the magnitude spectrum domain. Two kinds of compensation are adopted. Namely,

- **spectral compensation**

  The clean speech magnitude spectrum is estimated by the proposed IFI method.

- **energy compensation**

  Besides the magnitude spectrum, the energy term E in the MFCC feature vectors is also compensated. This is the energy compensation. By using the Parseval's theorem [56] for each segment,

$$\sum_t |n(t)|^2 = \frac{1}{2\pi} \int_{2\pi} |N(\omega)|^2 \, d\omega \tag{5.44}$$

  The clean speech energy term E is calculated by,

$$E = \ln\left\{ \left| \sum_t [y(t)]^2 - \frac{1}{2\pi} \int_{2\pi} |N(\omega)|^2 \, d\omega \right| \right\} \tag{5.45}$$

  If the two energies are equal, E is set to 0.

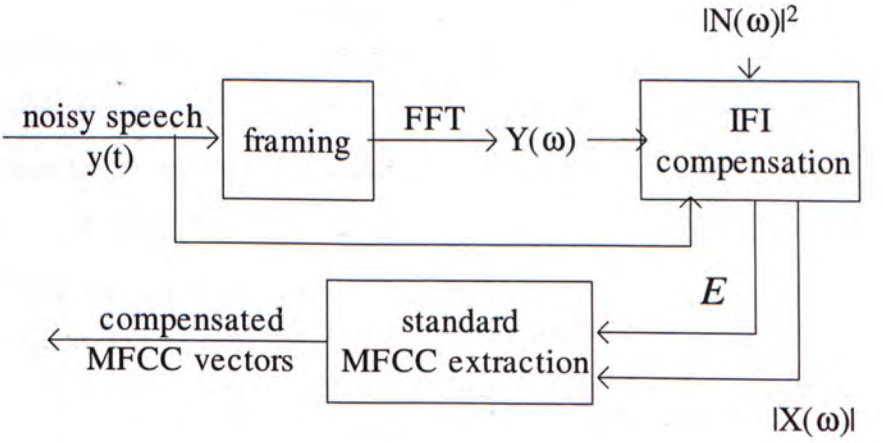


Figure 5.8: Block diagram of the noise compensated front-end system.

The delta and acceleration coefficients are computed with the compensated MFCC vectors. Figure 5.8 shows a block diagram of the noise compensated framework.

5.5 Recognition Experiments

The test set A in AURORA2 is used for the recognition experiments. Typical MFCC representation is used in the feature extraction part, which is identical to the one stated in Chapter 2, except rectangular window is used and no pre-emphasis is performed.

Although hamming window and pre-emphasis are used in standard MFCC extraction, they are not included in the compensated framework. This is because the mathematical derivation of IFI requires $Y(\omega)$ exactly equal the summation of $X(\omega)$ and $N(\omega)$. Generally speaking, $N(\omega)$ is obtained by some noise estimation methods, without Hamming window or pre-emphasis. As a consequence, rectangular window is used instead and no pre-emphasis is performed.

Two sets of recognition experiments are performed to evaluate the performance of IFI for noisy speech recognition. A reference noise power spectrum $|N(\omega)|^2$ is required for the compensation. Two estimators are adopted, they are the known noise spectrum and the weighted average method.

The known noise spectrum is calculated by subtracting the noisy speech waveform from the corresponding clean speech waveform and finding the resultant periodogram. It is used as an ideal noise estimator. On the other hand, the weighted average method provides a simple and coarse estimate. It is used to show how the recognition performance is, if a poor noise estimate is applied.

The training data set consists of 8440 clean utterances, which is the identical training set used in the baseline system. Two benchmark systems are chosen, namely the baseline system and the spectral subtraction system. The baseline system refers to the standard speech recognition system without any noise compensation or model adaptation. The spectral subtraction system is the compensation system that uses SS instead of IFI.

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.96	98.85	98.54	99.11	**98.87**
20	95.15	79.78	92.04	95.46	**90.61**
15	85.54	60.91	75.63	89.29	**77.84**
10	66.32	39.72	51.06	71.64	**57.19**
5	39.12	21.07	29.29	43.01	**33.12**
0	17.50	6.95	12.79	16.75	**13.50**
-5	9.58	2.36	7.49	7.71	**6.79**
average between 0 and 20 dB	60.73	41.69	52.16	63.23	**54.45**

Table 5.1: Word accuracy of the baseline system.

The recognition accuracy of the baseline system is shown in Table 5.1. The overall average is calculated as the average over SNRs between 0 dB and 20 dB. Table 5.2 and Table 5.3 show the results of the SS compensation system with $|N(\omega)|^2$ of the known noise spectrum and estimated from the weighted average method respectively.

Comparing the three sets of recognition results, the SS compensated system with known $|N(\omega)|^2$ always provides improvement over the baseline system, although it degrades gradually when the noise level increases. When $|N(\omega)|^2$ is not known, but estimated by the weighted average method, its recognition performance is significantly affected by the wrong estimate and the result is even worse than the one from the baseline system in most cases.

Table 5.4 and Table 5.5 show the recognition accuracy from the IFI compensation system with $|N(\omega)|^2$ of the known noise spectrum and estimated from the weighted average method respectively.

Regarding the recognition results of the IFI compensation system, for known $|N(\omega)|^2$, there is nearly no degradation found when SNR decreases and an accuracy of 97% is still achieved when SNR is equal to -5 dB. With the rough noise estimation from the weighted average method, the recognition performance is better than the one from the baseline system, especially when the SNR is above

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.96	98.85	98.54	99.11	**98.87**
20	98.74	98.52	97.97	98.52	**98.44**
15	98.37	98.16	97.70	98.40	**98.16**
10	97.61	97.58	97.32	97.69	**97.55**
5	96.41	95.71	95.94	96.33	**96.10**
0	93.00	91.02	91.65	93.40	**92.27**
-5	82.62	77.21	81.09	85.25	**81.54**
average between 0 and 20 dB	96.83	96.20	96.12	96.87	**96.50**

Table 5.2: Word accuracy of the SS compensation system with known noise spectrum.

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.43	98.67	98.09	98.64	**98.46**
20	84.92	68.59	90.22	84.45	**82.05**
15	71.42	52.00	79.78	71.83	**68.76**
10	50.81	33.43	58.31	46.93	**47.37**
5	29.01	17.74	31.94	23.48	**25.54**
0	13.08	8.65	11.93	9.07	**10.68**
-5	7.86	6.80	7.31	6.82	**7.20**
average between 0 and 20 dB	49.85	36.08	54.44	47.15	**46.88**

Table 5.3: Word accuracy of SS compensation system with noise estimate from the weighted average method.

test A

SNR/ dB	subway	babble	car	exhibition	average
clean	98.96	98.85	98.54	99.11	**98.87**
20	98.80	98.67	98.18	98.95	**98.65**
15	98.56	98.61	98.27	98.92	**98.59**
10	98.46	98.40	98.12	98.89	**98.47**
5	98.22	98.58	98.06	98.49	**98.34**
0	98.28	97.91	97.02	98.40	**97.90**
-5	97.11	96.98	96.96	97.69	**97.19**
average between 0 and 20 dB	98.46	98.43	97.93	98.73	**98.39**

Table 5.4: Word accuracy of IFI compensation system with known noise spectrum.

0 dB. If the noise spectrum is estimated from a different method, such as the histogram technique or the QBNE, it is expected that the result of IFI will be better than the weighted average method, due to the continuous and close tracking of noise estimation.

Comparing the average recognition accuracy of the two compensation systems, when $|N(\omega)|^2$ is known a priori, the performance of the SS compensation system is still affected by the substantial noise level in low conditions. As shown in Figure 5.6 previously, the SS-compensated spectrum still contains considerable amount of noise during non-speech periods even the noise estimate is exact. Besides, the rate of degradation is found to be much faster than the one from the IFI compensation system. Note that the inputs to the two systems are totally identical, hence, SS cannot take the full advantage of the accurate noise estimation for ASR. On the other hand, IFI provides significant improvement with the help of the phase difference information. When the SNR drops to a low value, there is still some negligible loss in accuracy. This is due to the unknown phase relationship between $Y(\omega)$ and $N(\omega)$, which is only necessary for exact spectral restoration, but may not be needed for ASR.

Since the magnitude trajectory is well preserved at the beginning and the

test A

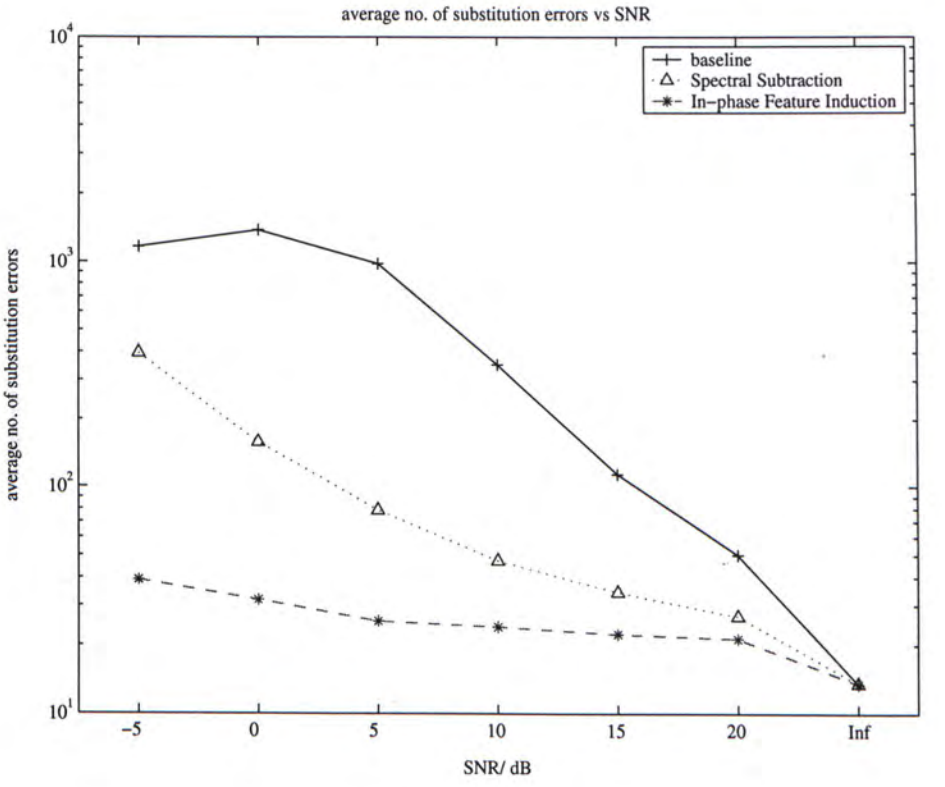
SNR/ dB	subway	babble	car	exhibition	average
clean	98.96	98.61	98.24	98.73	**98.64**
20	92.72	87.64	96.54	94.08	**92.75**
15	81.49	71.89	92.28	90.28	**83.99**
10	60.95	49.85	76.59	77.38	**66.19**
5	33.40	22.25	50.28	51.81	**39.44**
0	10.76	0.60	21.72	25.49	**14.64**
-5	7.73	-4.42	12.16	13.44	**7.23**
average between 0 and 20 dB	55.86	46.45	67.48	67.81	**59.40**

Table 5.5: Word accuracy of IFI compensation system with noise estimate from the weighted average method.

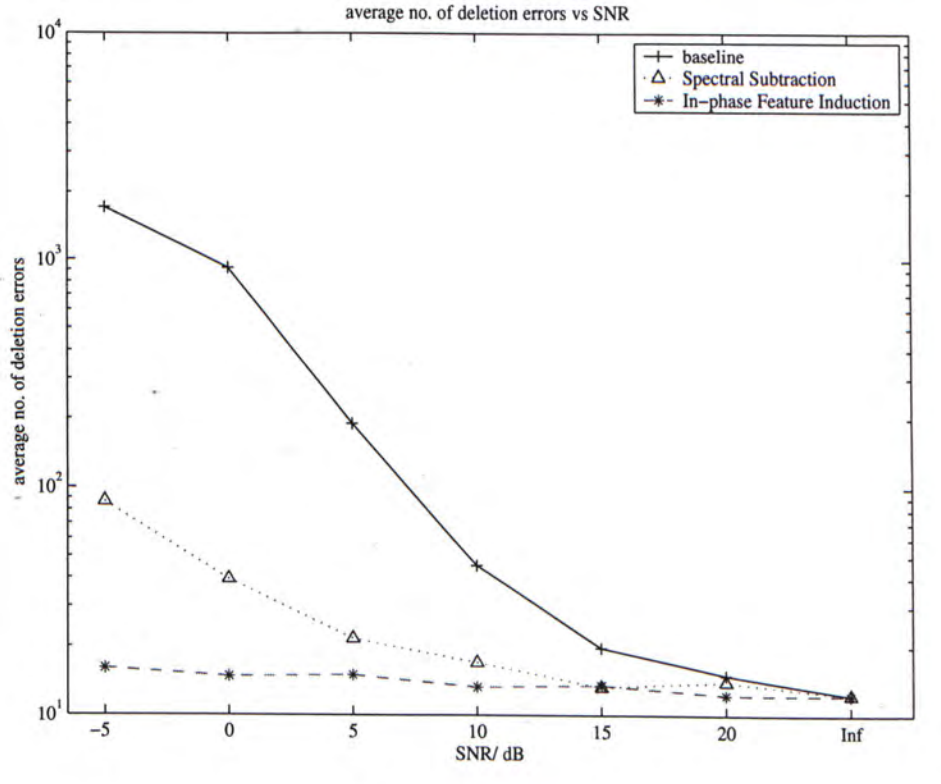
end of speech, it is expected there should be a great reduction in the number of insertion, deletion and substitution errors from the IFI compensation system. This is verified in Figure 5.9. The statistics are taken from the experiments with known $|N(\omega)|^2$.

With either compensation method, IFI or SS, the average number of error of any type is reduced. When the SNR decreases beyond 10 dB, the amounts of substitution, deletion and insertion found in the SS compensation system quickly increase. On the contrary, only slight increases are found in the IFI compensation system under the same situation.

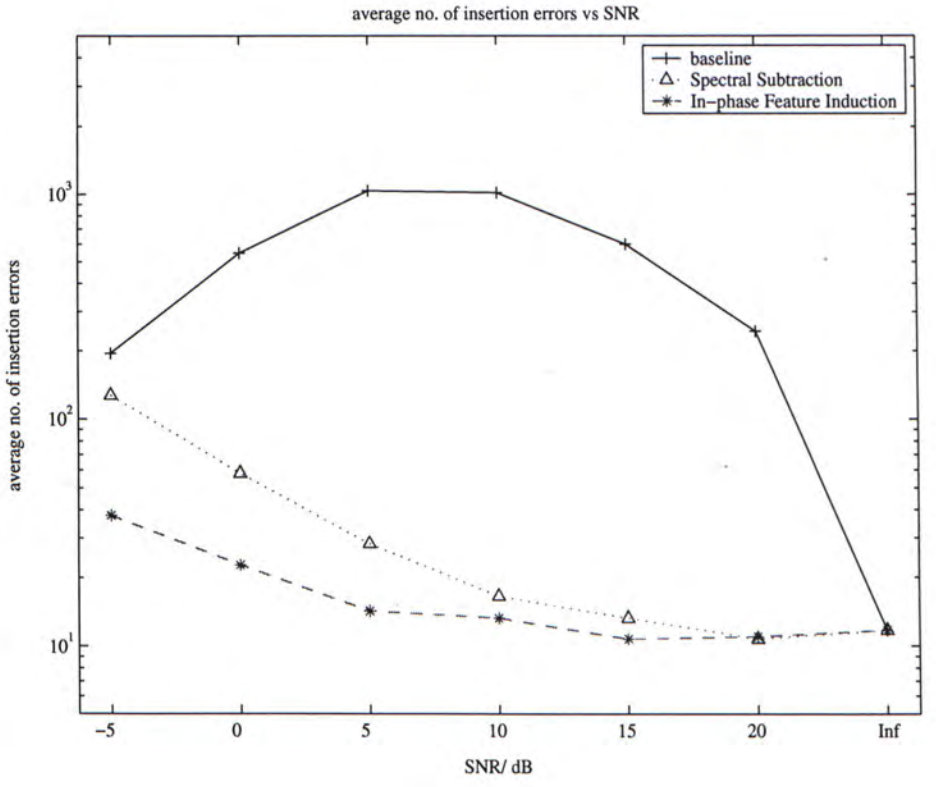
The average number of substitution, deletion and insertion errors under four types of noise are listed in Table 5.6, Table 5.7 and Table 5.8 respectively. These figures are averaged over more than 3200 words. Among the four types of noise, the baseline system suffers from excessive number of substitution and insertion error when it is a babble noise. Owning to the speech-like property of babble noise, substitution and insertion errors are highly probable. This phenomenon, however, in not encountered in the other two systems and the IFI compensation system always gives the lowest number of errors under different noise types.



(a) substitution error



(b) deletion error



(c) insertion error

Figure 5.9: Average number of errors versus SNR.

noise type	baseline	SS	IFI
subway	433.71	99.43	24.14
babble	868.50	119.86	26.29
car	568.86	120.86	34.29
exhibition	563.57	89.43	16.57
overall average	608.66	107.39	25.32

Table 5.6: Average number of substitution errors under four types of noise.

noise type	baseline	SS	IFI
subway	504.43	24.00	13.86
babble	199.86	36.00	13.86
car	579.43	32.43	16.14
exhibition	379.86	24.43	11.57
overall average	415.89	29.21	13.86

Table 5.7: Average number of deletion errors under four types of noise.

noise type	baseline	SS	IFI
subway	401.14	36.14	16.00
babble	898.29	47.14	16.57
car	447.57	37.29	20.71
exhibition	339.14	31.14	16.14
overall average	521.54	37.93	17.36

Table 5.8: Average number of insertion errors under four types of noise.

Referring to Figure 5.9(b), the number of deletion errors reported in the IFI compensation system is roughly the same, independent of the SNR. This is believed to be one of the major benefits bought from the accurate spectral estimation in IFI. Deletion error refers to the case when a correct word is omitted in the recognized sequence. As IFI closely-tracks the noise power in non-speech durations, including the between-word periods, the word boundaries are clearly defined and hence, the deletion errors are significantly reduced.

When the unknown $|N(\omega)|^2$ is estimated by some means, such as the weighted average method, the noise estimation accuracy is important to the recognition performance. With rough estimation from the weighted average method, the SS compensation system is greatly affected and becomes worse than the baseline, but the IFI compensation system is only degraded marginally. It is believed that the proposed compensation method requires an estimator with lower accuracy for $|N(\omega)|^2$ than SS needs, to provide similar recognition performance.

When the SNR is below 0 dB in subway and babble noise, IFI is found to be not working as well as SS for estimated noise spectrum. This may due to the non-stationary property of the noises and the inaccurate estimate of noise spectrum. The estimated clean spectrum in SS is often set to the spectral floor after subtraction, where the distortion from compensation is minimized. Provided the noise estimation is accurate enough, IFI compensation is reliable, as shown in Table 5.4.

Regarding the methodologies of the two methods, IFI does not require any parameter tuning, for instance, the oversubtraction factor and the noise floor.

The IFI compensation system uses the same approach as other speech enhancement schemes, where the noisy features are converted back to the clean features. In Section 3.3, a simple recognition framework is proposed to select the most appropriate acoustic model for recognition, according to the noisy speech characteristics. Comparing the performance of the two systems (Table 5.4 and Table 3.3), the accuracies of the IFI system are always higher. This is especially prominent in low SNR conditions.

Chapter 6

Conclusions

6.1 Summary and Discussions

This thesis addresses a real world problem. Even if a speech recognition system performs remarkably well in laboratory evaluations, when it is applied in practical situations, such as under a noisy acoustical environment, it often performs not nearly as well, sometimes with dramatic degradation. To deal with this problem, we consider a feature compensation which exploits the phase relationship between the input noisy speech and the background noise to find the clean speech magnitude spectrum. The phase information contributes to the correlation between the two spectra, which essentially affects the input magnitude spectrum.

It has been shown by experiments that the proposed In-phase Feature Induction (IFI) compensation method achieves a much higher recognition accuracy than the baseline system and the widely used Spectral Subtraction (SS) does. The average recognition accuracy of the baseline system is 54%. With the use of the IFI compensation method and known noise power spectrum, this figure is improved to 98%. Although the SS compensation system always brings improvements over the baseline, the improvement becomes smaller and smaller when the SNR decreases. For the IFI compensation system, when the SNR decreases, the recognition performance still remains satisfactory and the lowest average accuracy observed is 97% (the lowest average accuracy found in the SS

compensation system is 82%). In practice, the noise power spectrum needs to be estimated. From the experimental results, the proposed method is only slightly affected by the accuracy of the noise estimation and the recognition results are always better, in compared with the baseline and the SS compensation system.

Likewise, the principle of the proposed method is entirely based on the mathematical derivation of the noisy speech spectrum, such that the clean speech spectrum, noise spectrum and their correlation are all considered.

In addition to the new feature compensation method proposed, other important studies in this thesis include,

- reasons of performance degradation are explored in term of (1) degree of matching between training and testing conditions and (2) deviation of the noisy speech features from the clean speech features
- a simple recognition framework with model selection capability is firstly introduced to increasing the degree of matching
- a statistical-based noise estimation method is proposed, which is designed to prevent the overestimate of noise power and provide a good tracking at speech harmonic frequencies. It can used as an individual noise estimation for speech signals in other applications.

Both the simple recognition framework with model selection and IFI compensation method achieve satisfactory improvement over the baseline and the IFI system provides superior recognition under most cases.

Several factors are found to be extremely critical to the recognition performance under noisy conditions. Firstly, noise estimation plays an important role in feature compensation. As shown in Section 5.5, although the SS compensation system achieves reliable recognition performance when $|N(\omega)|^2$ is known, it is so sensitive to the noise estimation accuracy and deteriorates to be worse than the baseline system when only a rough estimator is used to provide $|N(\omega)|^2$. There is similar observation in the IFI compensation system, although the degradation is much smaller.

The location where the compensation takes place affects the way and the recognition performance to some extent. Generally speaking, model-adaptation method brings moderate improvement over the baseline and the performance is often less sensitive to any noise estimation accuracy. Model-adaptation is often used inside the back-end decoder. On the other hand, speech enhancement and feature compensation have shown promising noise reduction capability and better recognition performance than model-adaptation methods, provided noise estimation is accurate.

Recently, there are some robust speech recognition systems that use multiple microphones with more than one input signal or work together with some image processing such as lip reading to extract reliable visual features for speech recognition.

6.2 Future Directions

Although a spectral feature compensation method is proposed and it shows attractive recognition improvement, there are still a number of questions that remain unanswered. For example,

- In Chapter 3, we have tried to investigate how the recognition performance be affected by matching the training and testing conditions in term of noise type and SNR. It would be very useful if an analytical expression is formulated to represent the degree of matching in term of noise type and SNR.

- A noise estimation method M-R T-F QBNE is suggested in this thesis. It emphasizes the noise estimates at speech harmonic frequencies. During the noise estimation, the same method is applied for both voiced and unvoiced segments. However, speech harmonics exist in voiced segments only. Hence, it may be necessary to have a voiced/ unvoiced detection in the beginning.

- When people speak in a noisy environment, not only does the recorded

speech sum up the noise signal, but the pitch and frequency components also change. These variations are collectively called the Lombard effect [57]. These indirect influences of noise can be as great as the case when both speech and noise are recorded. The practical scenario is only realized by considering these Lombard effect together with the signal model used in Section 5.1.

Bibliography

[1] B.-H. Juang and F. K. Soong, "Hands-free telecommunications," in *Proc. International Workshop on Hands-Free Speech Communication*, 2001, pp. 5 – 10.

[2] G. M. Davis, *Noise Reduction in Speech Applications.* CRC Press, 2002.

[3] D. O'Shaughnessy, *Speech Communications: Human and Machine.* New York: IEEE Press, 2000.

[4] B.-H. Juang, "Speech recongnition in adverse environments," *Computer Speech and Language*, vol. 5, pp. 275 – 294, 1991.

[5] Y. Gong, "Speech recognition in noisy environments: a survey," *Speech Communication*, vol. 16, pp. 261 – 291, Apr. 1995.

[6] J.-C. Junqua and G. V. Noord, *Robustness in Languages and Speech Technology.* Netherlands: Kluwer Academic Publishers, 2001.

[7] J. S. Lim, *Speech Enhancement.* Englewood Cliffs, New Jersey: Prentice Hall, 1983.

[8] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 113 – 120, Apr. 1979.

[9] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc. ICASSP*, 1979, pp. 208 – 211.

[10] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals.* New York: IEEE Press, 2000.

[11] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction.* Chichester: John Wiley & Sons, Ltd., 2000.

[12] J. Droppo, A. Acero, and L. Deng, "A nonlinear observation model for removing noise from corrupted speech log mel-spectral energies," in *Proc. ICSLP*, 2002, pp. 1569 – 1572.

[13] C. Kermorvant, "A comparison of noise reduction techniques for robust speech recognition," Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, Valais, Switzerland, Tech. Rep. IDIAP-RR 99-10, 1999.

[14] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* New Jersey: Prentice Hall, 2001.

[15] S. W. Lee and P. C. Ching, "In-phase feature induction: an effective compensation technique for robust speech recognition," in *Proc. ICSLP*, 2004.

[16] S. W. Lee, P. C. Ching, and T. Lee, "Noise-robust automatic speech recognition using mainlobe-resilient time-frequency quantile-based noise estimation," in *Proc. ISCAS*, 2004.

[17] C. Becchetti and L. P. Ricotti, *Speech Recognition: Theory and C++ Implementation.* Chichester: John Wiley & Sons, Ltd., 2000.

[18] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[19] ——, "An introduction to hidden markov models," *IEEE ASSP Mag.*, vol. 3, pp. 4 – 16, Jan. 1986.

[20] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257 – 286, Feb. 1989.

[21] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* New Jersey: Prentice Hall, 1978.

[22] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 254 – 272, Apr. 1981.

[23] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357 – 366, Aug. 1980.

[24] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book.* Cambridge University Engineering Department, 2001.

[25] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR*, 2000.

[26] R. G. Leonard, "A database for speaker independent digit recognition," in *Proc. ICASSP*, 1984, pp. 328 – 331.

[27] "Transmission performance characteristics of pulse code modulation channels," Nov. 1996, international telecommunication union G.712.

[28] S. Furui, "Toward robust speech recogntion under adverse conditions," in *ESCA Workshop on Speech Workshop in Adverse Conditions*, 1992.

[29] B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 793 – 806, Aug. 1983.

[30] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *Proc. ICASSP*, 1995, pp. 153 – 156.

[31] W. A. Harrison, J. S. Lim, and E. Singer, "A new application of adaptive noise cancellation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 21 – 27, Feb. 1986.

[32] A. Vizinho, P. Green, M. Cooke, and L. Josifovski, "Missing data theory, spectral subtraction and signal-to-noise estimation for robust asr: an integrated study," in *Proc. Eurospeech*, 1999, pp. 2407 – 2410.

[33] C. Ris and S. Dupont, "Assessing local noise level estimation methods: Application to noise robust asr," *Speech Communication*, vol. 34, pp. 141 – 158, Apr. 2001.

[34] H. G. Hirsch, "Estimation of noise spectrum and its application to snr estimation and speech enhancement," International Computer Science Institute, Berkeley, USA, Tech. Rep. TR-93-012, 1993.

[35] R. Martin, "Spectral subtraction based on minimum statistics," in *Proc. EUSIPCO*, 1994, pp. 1182 – 1185.

[36] P. Motlíček and L. Burget, "Efficient noise estimation and its application for robust speech recognition," in *Proc. 5th International Conference Text, Speech and Dialogue*, 2002, pp. 229 – 236.

[37] L. Arslan, A. McCree, and V. Viswanathan, "New methods for adaptive noise suppression," in *Proc. ICASSP*, 1995, pp. 812 – 815.

[38] V. Stahl, A. Fischer, and R. Bippus, "Quantile based noise estimation for spectral subtraction and wiener filtering," in *Proc. ICASSP*, 2000, pp. 1875 – 1878.

[39] N. W. D. Evans and J. S. Mason, "Noise estimation without explicit speech, non-speech detection: a comparison of mean, modal and median based approaches," in *Proc. Eurospeech*, 2001, pp. 893 – 896.

[40] ——, "Time-frequency quantile-based noise estimation," in *Proc. EUSIPCO*, 2002, pp. 539 – 542.

[41] D. Ealey, H. Kelleher, and D. Pearce, "Harmonic tunnelling: Tracking non-stationary noises during speech," in *Proc. Eurospeech*, 2001, pp. 437 – 440.

[42] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers with Software and Programming Applications.* Boston: McGraw-Hill, 2001.

[43] B. M. Ayyub and R. H. McCuen, *Numerical Methods for Engineers.* Upper Saddle River, New Jersey: Prentice Hall, 1996.

[44] T. Shimamura and H. Kobayashi, "Weighted autocorrelation for pitch extraction of noisy speech," *IEEE Trans. Signal Processing*, vol. SP-9, pp. 727 – 730, Oct. 2001.

[45] J. Harrington and S. Cassidy, *Techniques in Speech Acoustics.* Boston: Kluwer Academic Publishers, 1999.

[46] A. Cichocki and S. ichi Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications.* Chichester: John Wiley & Sons, Ltd., 2002.

[47] W.-W. Hung and H.-C. Wang, "On the use of weighted filter bank analysis for the derivation of robust mfccs," *IEEE Signal Processing Lett.*, vol. 8, pp. 70 – 73, 2001.

[48] W.-W. Hung, "Derivation of robust mel frequency cepstral features based on snr-dependent adaptive filter bank analysis," *Electronics Letters*, vol. 37, pp. 1369 – 1370, 2001.

[49] N.-C. Wang, J.-W. Hung, and L.-S. Lee, "Data-driven temporal filters based on multi-eigenvectors for robust features in speech recognition," in *Proc. ICASSP*, 2003, pp. 400 – 403.

[50] T. Xu and Z. Cao, "Combination of feature weight and speech enhancement for robust asr at low snrs," in *Proc. IEEE TENCON*, 2002, pp. 441 – 444.

[51] P. Comon, "Independent component analysis: a new concept," *IEEE Signal Processing Lett.*, vol. 36, pp. 287 – 314, 1994.

[52] T. W. Lee, *Independent Component Anslysis: Theory and Applications.* Kluwer Academic Publishers, 1998.

[53] P. J. Moreno, B. R. Ramakrishnan, and R. Stern, "A vector taylor series approach for environment independent speech recognition," in *Proc. ICASSP*, 1996, pp. 733 – 736.

[54] N. Kitaoka and S. Nakagawa, "Evaluation of spectral subtraction with smoothing of time direction on the aurora 2 task," in *Proc. ICSLP*, 2002, pp. 477 – 480.

[55] H.-T. Hu, F.-J. Kuo, and H.-J. Wang, "Supplementary schemes to spectral subtraction for speech enhancement," *Speech Communication*, vol. 36, pp. 205 – 218, Mar. 2002.

[56] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. Upper Saddle River, New Jersey: Prentice Hall, 1997.

[57] J.-C. Junqua, "The lombard reflex and its role in human listeners and automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 1, pp. 510 – 524, 1993.