

SIGNAL PROCESSING FOR ROBUST SPEECH RECOGNITION

MOTIVATED BY AUDITORY PROCESSING

CHANWOO KIM

MAY 2010

ABSTRACT

Although automatic speech recognition systems have dramatically improved in recent decades, speech recognition accuracy still significantly degrades in noisy environments. While many algorithms have been developed to deal with this problem, they tend to be more effective in stationary noise such as white or pink noise than in the presence of more realistic degradations such as background music, background speech, and reverberation. At the same time, it is widely observed that the human auditory system retains relatively good performance in the same environments. The goal of this thesis is to enhance the robustness of speech recognition systems through consideration of the human auditory system.

In our work we focus on several aspects of auditory processing. We first note that nonlinearities in the representation appear to play an important role in speech recognition. We observe that the threshold portion of the auditory nonlinearity impacts significantly on robustness while the corresponding saturation portion has relatively little effect on robustness. We propose several efficient algorithms that are motivated by this observation. These changes are shown to improve the robustness of speech recognition system significantly.

We also reconsider the impact of time-frequency resolution. We observe consistently that temporal observation windows needed to provide the best estimates of attributes of noise needed for robust recognition are of longer duration than the frame duration that provides best results for automatic speech recognition. We also identify the very important role that frequency smoothing can play in signal processing for robust recognition.

Additionally we note that humans are largely insensitive to the slowly-varying changes in the signal components that are most likely to arise from noise components of the input. These components are removed by modulation filtering or nonlinear processing based on power distribution information.

It is well known that humans are also excellent in separating sound sources based on their direction of arrival. We propose an efficient algorithm for binaural sound source separation that operates in the frequency domain. We also develop methods that enable us to determine the separation threshold.

CONTENTS

1. <i>INTRODUCTION</i>	1
2. <i>REVIEW OF PREVIOUS STUDIES</i>	4
2.1 Frequency scales	4
2.2 Temporal integration times	5
2.3 Auditory nonlinearity	6
2.4 Feature Extraction System	7
2.5 Noise Power Subtraction Algorithm	10
2.5.1 Boll's approach	10
2.5.2 Hirsch's approach	10
2.6 Algorithms Motivated by Modulation Frequency	11
2.7 Normalization Algorithm	13
2.7.1 CMN, MVN, HN, and DCN	13
2.7.2 CDCN and VTS	15
2.8 ZCAE and related algorithms	17
2.9 Discussion	18
3. <i>TIME AND FREQUENCY RESOLUTION</i>	28
3.1 Time-frequency resolution trade-off in short-time Fourier analysis	29
3.2 Time Resolution for Robust Speech Recognition	30
3.2.1 Medium-duration running average method	30
3.2.2 Medium duration window analysis and re-synthesis approach	32
3.3 Channel Weighting	33
3.3.1 Channel Weighting of Binary Parameters	33
3.3.2 Weighting factor averaging across channels	35

3.3.3	Comparison between the triangular and the gammatone filter bank . . .	36
3.4	Proposed work	36
4.	<i>AUDITORY NONLINEARITY</i>	38
4.1	Introduction	38
4.2	Human auditory nonlinearity	38
4.3	Speech recognition using different nonlinearities	41
4.4	Recognition results using human auditory nonlinearity and discussions	42
4.5	Shifted Log Function and Power Function Approach	44
4.6	Speech Recognition Result Comparison of Several Different Nonlinearities . .	46
4.7	Proposed Work	49
5.	<i>SMALL POWER BOOSTING ALGORITHM</i>	50
5.1	Introduction	50
5.2	The Principle of Small Power Boosting	50
5.3	Small Power Boosting with Re-synthesized Speech (SPB-R)	54
5.4	Small Power Boosting with Direct Feature Generation (SPB-D)	56
5.5	log spectral mean subtraction	60
5.6	Experimental results	61
5.7	Conclusion	64
5.8	Proposed Work	64
6.	<i>ENVIRONMENTAL COMPENSATION USING POWER DISTRIBUTION NOR- MALIZATION</i>	66
6.1	Medium-Duration Power bias subtraction	67
6.1.1	Medium-duration power bias removal based on arithmetic-to-geometric mean ratios	67
6.1.2	Removing the power bias	69
6.1.3	Simulation results with Power Normalized Cepstral Coefficient	70

6.2	Bias estimation based on Maximizing the sharpness of the power distribution and power flooring	70
6.2.1	Power bias subtraction	72
6.2.2	Experimental results and conclusions	75
6.3	Power-function-based power distribution normalization algorithm	76
6.3.1	Structure of the system	76
6.3.2	Arithmetic mean to geometric mean ratio of powers in each channel and its normalization	76
6.3.3	Medium duration window	79
6.3.4	On-line implementation	79
6.3.5	Simulation results of the on-line power equalization algorithm	80
6.4	Conclusions	81
6.5	Proposed Work	82
7.	<i>POWER NORMALIZED CEPSTRAL COEFFICIENT</i>	90
7.1	Derivation of the power function nonlinearity	91
7.2	Medium-duration power bias removal	94
7.2.1	Medium-duration power bias removal based on arithmetic-to-geometric mean ratios	94
7.2.2	Removing the power bias	95
7.3	Experimental results and conclusions	96
8.	<i>COMPENSATION WITH 2 MICS</i>	99
8.1	Introduction	99
8.2	Phase-difference-based binary time-frequency mask estimation	100
8.2.1	The effect of the window length and channel weighting	103
8.3	Experimental Results	103
8.4	Obtaining the ITD threshold	105
8.4.1	Complementary mask generation	105
8.4.2	Obtaining the ITD threshold using the minimum correlation criterion	106
8.4.3	Experimental Results	107

8.4.4	Conclusion	108
8.5	PROPOSED WORK	109
8.6	Threshold selection algorithm	109
9.	<i>THESIS GOAL AND TIME TABLE</i>	115
9.1	Deliverables	115
9.2	Timetable	118

LIST OF FIGURES

2.1	<i>The comparison between the MEL, Bark, and the ERB scales</i>	5
2.2	<i>The intensity-rate relation in the human auditory system simulated by the model proposed by M. Heinz. et. al. [1]</i>	7
2.3	<i>Cube-root power law nonlinearity, MMSE power-law nonlinearity, and logarithmic nonlinearity are compared. Plots are shown on two different scales: 2.3(a) in Pa and 2.3(b) in dB Sound Pressure Level (SPL).</i>	8
2.4	<i>The block diagram of MFCC and PLP</i>	9
2.5	<i>Comparison between MFCC and PLP in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	20
2.6	<i>Comparison between MFCC and PLP in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	21
2.7	<i>The frequency response of the high-pass filter proposed by Hirsch et al. [2]</i>	22
2.8	<i>The frequency response of the band-pass filter proposed by Hermansky et al. [3]</i>	22
2.9	<i>Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	23
2.10	<i>Comparison between different normalization approaches in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	24
2.11	<i>: (a) Silence appended and prepended to the boundaries of clean speech (b) 10-dB of white Gaussian noise is added to the data used in (a)</i>	25

2.12	<i>Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	26
2.13	<i>Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation</i>	27
3.1	<i>(a) The block diagram of the Medium-duration-window Running Average (MRA) Method (b) The block diagram of the Medium-duration-window Analysis Synthesis (MAS) Method</i>	31
3.2	<i>Frequency response depending on the medium-duration parameter M</i>	32
3.3	<i>Speech recognition accuracy depending on the medium-duration parameter M</i>	33
3.4	<i>(a) The spectrograms from clean speech with $M = 0$, (b) with $M = 2$, and (c) with $M = 4$ (d) The spectrograms from speech corrupted by 5 dB additive white noise with $M = 0$, (e) with $M = 2$, and (f) with $M = 4$</i>	34
3.5	<i>(a) Gammatone Filterbank Frequency Response and (b) Normalized Gammatone Filterbank Frequency Response</i>	36
4.1	The relation between the intensity and the rate. Simulation was done using the auditory model developed by Heinz. et al [4]: 4.1(a) shows the relation in a cat model at different frequencies. 4.1(b) shows the relation in a human model, and 4.1(c) shows the average across different channels, and 4.1(d) is the smoothed version of 4.1(c) using spline.	39
4.2	<i>The comparison between the intensity and rate response in the human auditory model [1] and the logarithmic curve used in MFCC. A linear transformation is applied to fit the logarithmic curve to the intensity-rate curve.</i>	40
4.3	<i>The structure of the feature extraction system 4.3(a): MFCC, 4.3(b): PLP, and 4.3(c): General nonlinearity system</i>	41
4.4	<i>Speech recognition accuracy obtained in different environments using the human auditory intensity-rate nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation</i>	43

4.5	4.5(a) Rate-intensity curve and its stretched form in the form of shifted log	
	4.5(b) Power function approximation to the stretched form of the rate-intensity curve	45
4.6	Speech recognition accuracy obtained in different environments using the shifted log nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation	46
4.7	Speech recognition accuracy obtained in different environments using the power function nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation	47
4.8	Comparison of different nonlinearities (human rate-intensity curve, under different environments: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation	48
5.1	Comparison of the Probability Density Functions (PDFs) obtained in three different environments : clean, 0-dB additive background music, and 0-dB additive white noise	51
5.2	The total nonlinearity consists of small power boosting and the subsequent logarithmic nonlinearity in the SPB algorithm	52
5.3	Small power boosting algorithm which resynthesizes speech (SPB-R). Conventional MFCC processing is followed after resynthesizing the speech.	55
5.4	Word error rates obtained using the SPB-R algorithm as a function of the value of the SPB Coefficient. The filled triangles at the y-axis represent the baseline MFCC performance for clean speech (upper triangle) and for additive background music noise at 0 dB SNR (lower triangle), respectively.	56
5.5	Small power boosting algorithm with direct feature generation (SPB-D)	57
5.6	The effects of weight smoothing on performance of the SPB-D algorithm for clean speech for speech corrupted by additive background music at 0 dB. The filled triangles at the y-axis represent the baseline MFCC performance for clean (upper triangle) and 0 dB additive background music (lower triangle) respectively. The SPB coefficient α was 0.02.	58

5.7	<i>Spectrograms obtained from a clean speech utterance using different processing: (a) conventional MFCC processing, (b) SPB-R processing, (c) SPB-D processing without any weight smoothing, and (d) SPB-D processing with weight smoothing $M = 4, N = 1$ in (5.9). A value of 0.02 was used for the SPB coefficient α. (5.2)</i>	59
5.8	<i>The effect of Log Spectral Subtraction for (a) background music and (b) white noise as a function of the moving window length. The filled triangles at the y-axis represent baseline MFCC performance.</i>	62
5.9	<i>Comparison of recognition accuracy between VTS, SPB-CW and MFCC processing: (a) additive white noise, (b) background music.</i>	63
6.1	<i>Comparison between $G(l)$ coefficients for clean speech and speech in 10-dB white noise, using $M = 3$ in (7.2).</i>	68
6.2	<i>The block diagram of the power function-based power equalization system</i>	71
6.3	<i>The structure of PNCC feature extraction</i>	71
6.4	<i>Medium duration power $q[m, l]$ obtained from the 10th channel of a speech utterance corrupted by 10-dB additive background music. The bias power level (q_b) and subtraction power level (q_0) are represented as horizontal lines. Those power levels are the actual calculated levels calculated using the PBS algorithm. The logarithm of the AM-to-GM ratio is calculated only from the portions of the line that are solid.</i>	72
6.5	<i>The dependence of speech recognition accuracy obtained using PNCC on the medium-duration window factor M and the power flooring coefficient c_0. Results were obtained for (a) the clean RM1 test data (b) the RM1 test set corrupted by 0-dB white noise, and (c) the RM1 test set corrupted by 0-dB background music. The filled triangle on the y-axis represents the baseline MFCC result for the same test set.</i>	83
6.6	<i>The corresponding dependence of speech recognition accuracy on the value of the weight smoothing factor N. The filled triangle on the y-axis represents the baseline MFCC result for the same test set. For c_0 and M, we used 0.01 and 2 respectively.</i>	84

6.7	<i>Speech recognition accuracy obtained in different environments for different training and test sets. The RM1 database was used to produce the data in (a), (b), and (c), and the WSJ0 SI-84 training set and WSJ0 5k test set were used for the data of panels (d), (e), and (f).</i>	85
6.8	<i>The logarithm of the ratio of arithmetic mean to geometric mean of power from clean (a) and noise speech corrupted by 10 dB white noise (b). Data is collected from 1,600 training utterances of the resource management DB</i>	86
6.9	<i>The assumption about the relationship between $P_{cl}[m,l]$ and $P[m,l]$</i>	86
6.10	<i>Speech recognition accuracy as a function of the window length for the DARPA RM database corrupted by (a) white noise and (b) background music noise.</i>	87
6.11	<i>Sample spectrograms illustrating the effects of on-line PPDN processing. (a) original speech corrupted by 0-dB additive white noise, (b) processed speech corrupted by 0-dB additive white noise (c) original speech corrupted by 10-dB additive music noise (d) processed speech corrupted by 10-dB additive music noise (e) original speech corrupted by 5-dB street noise (f) processed speech corrupted by 5-dB street noise</i>	88
6.12	<i>Performance comparison for the DARPA RM database corrupted by (a) white noise, (b) street noise, and (c) music noise.</i>	89
7.1	<i>Comparison of the PNCC feature extraction discussed in this paper with MFCC and PLP feature extraction.</i>	91
7.2	<i>Upper panel: Observed frequency-averaged mean rate of auditory-nerve firings versus intensity (dotted curve) and its piece-wise linear approximation (solid curve). Lower panel: Piece-wise linear rate-level curve with no saturation (solid curve) and best-fit power function approximation (dotted curve).</i>	93
7.3	<i>Comparison between $G(i)$ coefficients for clean speech and speech in 10-dB white noise, using $M = 3$ in (7.2).</i>	95
7.4	<i>Speech recognition accuracy obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of white Gaussian noise added to the data used in panel (c).</i>	97

8.1	<i>The block diagram of the Phase Difference Channel Weighting (PDCW) algorithm</i>	100
8.2	<i>Sample spectrograms illustrating the effects of PDCW processing. (a) original clean speech, (b) noise-corrupted speech, (c) reconstructed (enhanced) speech (d) the time-frequency mask obtained with (8.11b) (e) gammatone channel weighting obtained from the time-frequency mask in (3.7) (e) final frequency weighting shown in (5.7) (f) enhanced speech spectrogram using the entire PDCW algorithm</i>	110
8.3	<i>The dependence of word recognition accuracy (100% – WER) on the window length, using an SIR of 10 dB and various reverberation times. The filled symbols at 0 ms represent baseline results obtained with a single microphone.</i>	111
8.4	<i>Speech recognition accuracy using different algorithms (a) in the presence of an interfering speech source as a function of SNR in the absence of reverberation, (b,c) in the presence of reverberation and speech interference, as indicated, and (d) in the presence of natural real-world noise.</i>	112
8.5	<i>The block diagram of the optimal ITD selection algorithm for sound source separation</i>	112
8.6	<i>Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at 45 degrees at different reverberation times (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms.</i>	113
8.7	<i>Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at different locations at different reverberation time (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms.</i>	114

1. INTRODUCTION

In recent decades, speech recognition systems have significantly improved. However, obtaining good performance for noisy environment still remains as a very challenging task. The problem is if the training condition is not matched to the test condition, then performance degrades significantly. These environmental differences might be due to speaker differences, channel distortion, reverberation, additive noise, and so on.

To tackle this problem, many algorithms have been proposed up to now. The simplest way of environmental normalization is assuming that the mean of each element of cepstral feature vector is zero for all utterances. This is often called Cepstral Mean Normalization (CMN) [5]. CMN is known to be able to remove convolutional distortion, if the impulse response is very short, and it is also helpful additive noise as well. Mean Variance Normalization (MVN) [5] [6] can be considered to be an extension of this idea. In MVN, we assume that both the mean and the variance of each element of feature vectors are the same across all utterances. More general case is the histogram normalization. In this approach, it is assumed that the Cumulative Distribution Function (CDF) of all features are the same. Recently, it is found that if we do histogram normalization on the delta cepstrum as well, the performance is better than the original histogram normalization.

Another class of ideas try to estimate the noise components for different clusters and use this information to estimate the original clean spectrum. Codeword Dependent Cepstral Normalization (CDCN) [7] and Vector Taylor Series (VTS) [8] belong to these kinds of idea. Spectral subtraction [9] is subtracting the noise spectrum in the spectrum domain.

Even though a number of algorithms have shown improvements for stationary noise (*e.g.*[10, 11]), improvement in non-stationary noise remains a difficult issue (*e.g.* [12]). In these environments, auditory processing (*e.g.*[13]) and missing-feature-based approaches (*e.g.*[14]) are promising. In [13], we could observe that better speech recognition accuracy

can be obtained by using more faithful human auditory model.

An alternative approach is signal separation based on analysis of differences in arrival time (*e.g.* [15, 16, 17]). It is well documented that the human binaural system bears remarkable ability in speech separation (*e.g.* [17]). Many models have been developed that describe various binaural phenomena (*e.g.* [18, 19]), typically based on interaural time difference (ITD), interaural phase difference (IPD), interaural intensity difference (IID), or changes of interaural correlation. The Zero Crossing Amplitude Estimation (ZCAE) algorithm was recently introduced by Park [16]. These algorithms (and similar ones by other researchers) typically analyze incoming speech in bandpass channels and attempt to identify the subset of time-frequency components for which the ITD is close to the nominal ITD of the desired sound source (which is presumed to be known *a priori*). The signal to be recognized is reconstructed from only the subset of “good” time-frequency components. This selection of “good” components is frequently treated in the computational auditory scene analysis (CASA) literature as a multiplication of all components by a binary mask that is nonzero for only the desired signal components.

The goal of this thesis is to develop a robust speech recognition algorithm motivated by the human auditory systems at the level of peripheral processing and simple binaural analysis. These include time and frequency resolution analysis, auditory nonlinearity, power normalization, and source separation using two microphones.

In time-frequency resolution analysis, we will discuss what would be the optimal window length for noise compensation. We will also talk about frequency weighting or channel weighting. We will propose an efficient way of normalizing the noise component based on this observation.

Next, we focus on the role that auditory nonlinearity plays in robust speech recognition. Even though the relationship between the intensity of a sound and its perceived loudness is well known, there have not been many attempts to analyze the effects of rate-level nonlinearity. In this thesis, we discuss several different nonlinearities derived from the rate-intensity relation models of processing by the human auditory nerve, and will show that power function nonlinearity is more robust than the logarithmic nonlinearity which is currently being used in MFCC.

Power normalization is based on the observation that noise power changes less rapidly

than speech power. As a convenient measure, we propose the use of the AM-to-GM (Arithmetic Mean-to-Geometric Mean) ratio. If the signal is highly non-stationary like speech, then the AM-to-GM ratio will have larger values. However, if the signal is more smoothly changing, then this ratio will decrease. By estimating the ideal AM-to-GM ratio from training database of clean speech, we developed two algorithms : the Power-function based Power Equalization (PPE) algorithm and the Power Bias Subtraction (PBS) algorithm.

This thesis proposal is organized as follows: Chapter 2 provides a brief review of background theories and several related algorithms. We will briefly discuss the key concepts and effectiveness of each idea and algorithm. In Chapter 3, we will discuss time and frequency resolution and its effect on speech recognition. We will see that the window length and frequency weighting have significant impact on speech recognition accuracy. Chapter 5 deals with auditory nonlinearity and how it affects the robustness of speech recognition systems. Auditory nonlinearity is the intrinsic relation between the intensity of the sound and representation in auditory processing, and it plays an important role in speech recognition. In Chapter 7, we introduce a new feature extraction algorithm called power normalized cepstral coefficients (PNCC). PNCC processing can be considered to be an application of some of principles of time-frequency analysis as discussed in Chapter 3, auditory nonlinearity as discussed in Chapter 5, and power bias subtraction as discussed in Chapter 6. In Chapter 8, we discuss how to enhance speech recognition accuracy using two microphones. We will talk about our new algorithm which is called Phase Difference Channel Weighting (PDCW).

2. REVIEW OF PREVIOUS STUDIES

In this chapter, we will review some background theories relevant to this thesis.

2.1 Frequency scales

Frequency scales relate how the physical frequency of an incoming signal is related to the representation of that frequency by the human auditory system. In general, the peripheral auditory system can be modeled as a bank of bandpass filters, of approximately constant bandwidth at low frequencies and of a bandwidth that increases in rough proportion to frequency at higher frequencies. Because different psychoacoustical techniques provide somewhat different estimates of the bandwidth of the auditory filters, several different frequency scales have been developed to fit the psychophysical data. Some of the widely used frequency scales include the MEL scale [20], the BARK scale [21], and the ERB (Equivalent rectangular bandwidth) scale [22]. The popular Mel Frequency Cepstral Coefficients (MFCC) incorporate the MEL scale, which is represented by the following equation:

$$Mel(f) = 2595 \log(1 + f/700) \quad (2.1)$$

The MEL scale that was proposed by Stevens et al [20], describes how a listener judges the distance between pitches. The reference point is obtained by defining a 1000 Hz tone 40 dB above the listener's threshold to be 1000 mels.

Another frequency scale which is called the Bark scale was proposed by E. Zwicker [21]:

$$Bark(f) = 13 \arctan(0.00076f) + 3.5 \arctan\left(\frac{f}{7500}\right)^2 \quad (2.2)$$

In PLP [23], the Bark-Frequency relation is based on the transformation given by Schroeder:

$$\Omega(f) = 6 \ln\left(\frac{f}{600} + \left(\frac{f}{600}\right)^{0.5}\right) \quad (2.3)$$

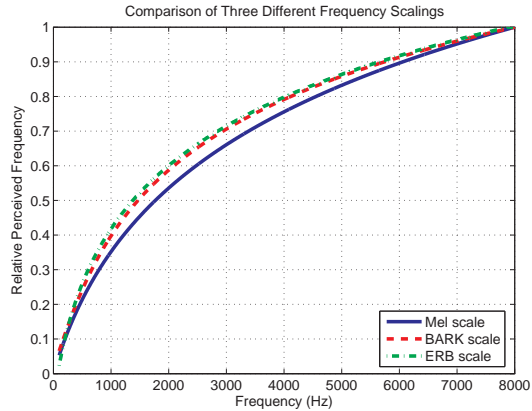


Fig. 2.1: The comparison between the MEL, Bark, and the ERB scales

Later on, Moore and Glasberg [22] proposed the ERB (Equivalent Rectangular Bandwidth) scale modifying the Zwicker’s loudness model. The ERB scale is a measure that gives an approximation to the bandwidth of filters in human hearing using rectangular bandpass filters; several different approximations of the ERB scale exist. The following is one of such approximations relating the ERB and the frequency f :

$$v = 11.17 \log \left(1 + \frac{46.065 f}{f + 14678.49} \right) \quad (2.4)$$

Fig. 2.1 compares the three different frequency scales in the range between 100 Hz and 8000 Hz. It can be seen that they describe very similar relationships between frequency and its representation by the auditory system.

2.2 Temporal integration times

It is well known that there is a trade-off between the time-resolution and the frequency resolution that depends on the window length (*e.g.* [24]). Longer windows provide better frequency resolution, but worse time resolution. Usually in speech processing, we assume that a signal is quasi-stationary within an analysis window, so typical window durations for speech recognition are on the order of 20 ms to 30 ms. [25].

2.3 Auditory nonlinearity

Auditory nonlinearity is related to how humans perceive loudness. There are many different ways of measuring this.

One kind of nonlinearity is obtained by physiologically measuring the average rate of the neural firing times of fibers of the auditory nerve as a function of the intensity of the pure tone input at a specified frequency. As shown in Fig. 2.2, this nonlinearity is characterized by the auditory threshold and the saturation point. The curves in Fig. 2.2 are obtained using the auditory simulation system developed by Heinz et al. [1].

The other way of representing auditory nonlinearity is based on psychophysics. One of the well known rules is Steven's power law of hearing [26]. This rule relates intensity and perceived loudness by fitting data from multiple observers using a power function:

$$L = (I/I_0)^3 \quad (2.5)$$

This rule has been used in Perceptual Linear Prediction (PLP).

Another commonly-used relationship is that is used in MFCC the logarithmic curve, which relates intensity and loudness using a log function. The definition of sound pressure level (SPL) is also motivated by this rule, as given by:

$$L_p = 20 \log_{10} \left(\frac{p_{rms}}{p_{ref}} \right) \quad (2.6)$$

The commonly used value of p_{ref} is $20\mu\text{Pa}$, which was once considered to be the threshold of human hearing, when the definition was established.

In Fig. 2.3, we compare these nonlinearities. In addition to the nonlinearities mentioned in this Subsection, we included another power law nonlinearity which is an approximation to the physiological model between 0 dB SPL and 50 dB SPL in the Minimum Mean Square Error (MMSE) sense. In this approximation, the estimated power coefficient is around 1 / 10.

In Fig. 2.3(a), we compare these curves using an x-axis in Pa. In this figure, with the exception of the cube power root, all nonlinearity curves are very similar. However, as shown in Fig. 2.3(b), if we use the logarithmic scale (dB SPL) on the x-axis, we can observe a significant difference between the power-law nonlinearity and the logarithmic nonlinearity

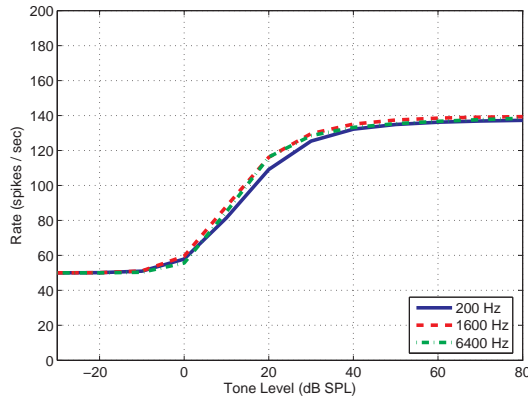


Fig. 2.2: *The intensity-rate relation in the human auditory system simulated by the model proposed by M. Heinz. et. al. [1]*

in the region below the auditory threshold. As will be discussed in Chap. 5, this difference plays an important role for robust speech recognition.

2.4 Feature Extraction System

The most widely used forms of feature extraction are Mel Frequency Cepstral Coefficient (MFCC) and Perceptual Linear Prediction (PLP) [23]. These feature extraction systems are based on the theories briefly reviewed in Section 2.1 to Section 2.3. Fig. 2.8 illustrates the block diagram of MFCC and PLP. In this section, we will briefly talk about those feature processing algorithms.

In MFCC processing, the first stage is pre-emphasis. We usually use a first-order high pass filter for pre-emphasis. Short-time Fourier Transform (STFT) analysis is performed using a hamming window, and triangular frequency integration is done for spectral analysis. The logarithmic nonlinearity stage follows, and Discrete Cosine Transform (DCT) is done to obtain the feature.

PLP processing is also similar to MFCC processing. The first stage is STFT analysis; critical band integration follows. For band integration, trapezoidal windows are employed. Unlike MFCC, pre-emphasis is done based on the equal loudness curve after the band integration. Nonlinearity in PLP is based on the power-law nonlinearity proposed by Stevens

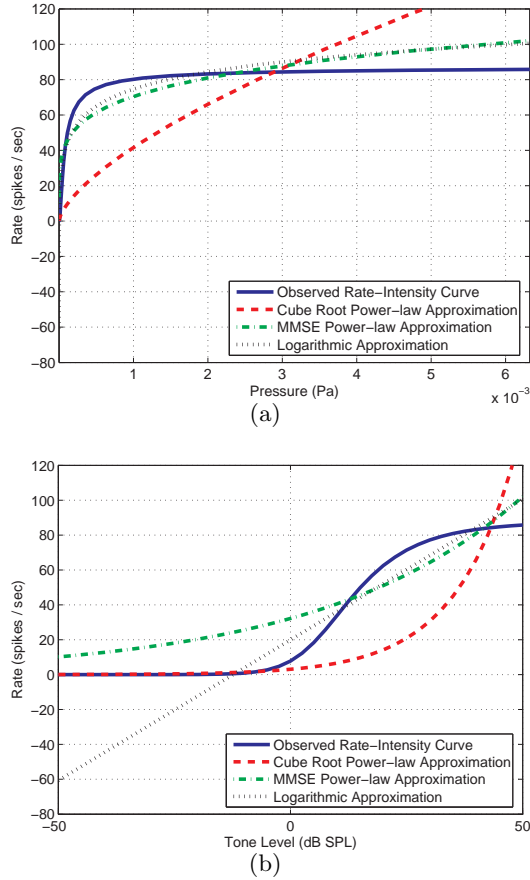


Fig. 2.3: Cube-root power law nonlinearity, MMSE power-law nonlinearity, and logarithmic nonlinearity are compared. Plots are shown on two different scales: 2.3(a) in Pa and 2.3(b) in dB Sound Pressure Level (SPL).

[23]. After this stage, Inverse Fast Fourier Transform (IFFT) and Linear Prediction (LP) analysis are performed in sequence. Cepstral recursion is also usually performed to obtain the final feature from the LP coefficients [27].

Fig. 2.5 shows speech recognition accuracies obtained under various noisy conditions. We used subsets of 1600 utterances for training and 600 utterances for testing from the DARPA Resource Management 1 (RM1). In other experiments, which are shown in Fig. 2.6, we used WSJ0-si84 training set and WSJ0 5k test set. For training the acoustical model, we used SphinxTrain 1.0 and for decoding, we used Sphinx 3.8.

For MFCC processing, we used `sphinx_fe` included in `sphinxbase` 0.4.1. For PLP

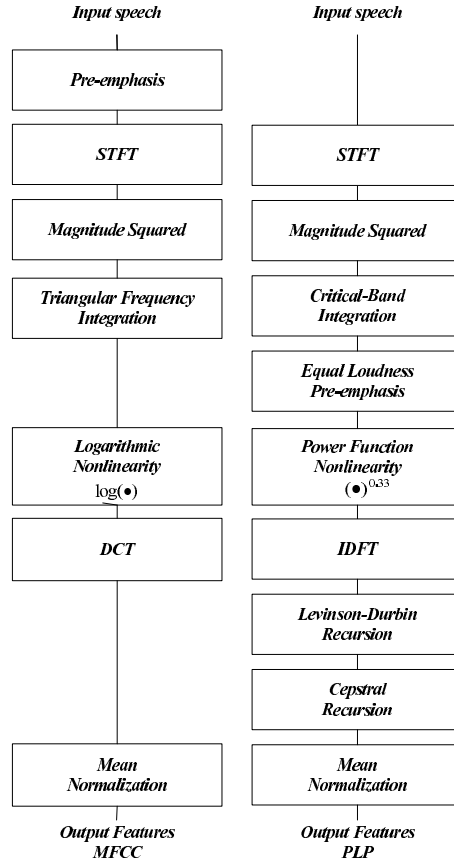


Fig. 2.4: The block diagram of MFCC and PLP

processing, we used both HTK 3.4 and the matlab package provided by the D. Ellis group [28]. Both of the PLP packages show similar performance, but for reverberation and interfering speaker environment, PLP included in HTK showed better performance.

In all these experiments, we used 12-th order feature vectors including the 0-th coefficient with their delta and delta-delta cepstra. As shown in these experiments, MFCC and PLP show comparable speech recognition results. However, in our experiments, RASTA processing is not helpful compared to the conventional Cepstral Mean Normalization (CMN).

2.5 Noise Power Subtraction Algorithm

In this section, we discuss conventional ways of accomplishing noise power compensation. The earliest form of a noise power compensation scheme was the spectral subtraction technique [9]. In spectral subtraction, we assume that speech is corrupted by additive noise. The basic idea behind this method is that we estimate the noise spectrum from non-speech segments of corrupt speech, which can be detected by applying a Voice Activity Detector (VAD). After estimating the noise spectrum, these values are subtracted from the corrupt speech spectrum.

2.5.1 Boll's approach

In Boll's approach, the first step is running a Voice Activity Detector, which decides whether the current frame belongs to speech segments or noisy segments. If the segment is determined to be a noisy segment, then the noise spectrum is estimated by that segment. For the following speech spectrum, the subtraction is done in the following way:

$$|\tilde{X}(m, l)| = \max(|X(m, l)| - N(m, l), \delta|X(m, l)|) \quad (2.7)$$

where δ is a small constant to prevent the subtracted spectrum from having a negative spectrum value, $N(m, l)$ is the noise spectrum, and $X(m, l)$ is the corrupt speech spectrum. m and l denote the frame and channel indices, respectively.

2.5.2 Hirsch's approach

In [29], Hirsch estimates the noise level in the following way: First, the continuous average of the spectrum is calculated:

$$|N(m, l)| = \lambda|N(m - 1, l)| + (1 - \lambda)|X(m, l)| \quad \text{if } |X(m, l)| < \beta|N(m, l)| \quad (2.8)$$

where m is the frame index and l is the frequency index. Note that the above equation is the realization of the 1-st order IIR lowpass filter.

If the magnitude spectrum is larger than $\beta N(m, l)$, we do not update the estimate noise spectrum. For β , Hirsch suggested using a value between 1.5 and 2.5.

The major difference between Hirsch's approach compared to Boll's approach is that the noise spectrum is continuously updated.

2.6 Algorithms Motivated by Modulation Frequency

It has long been believed that modulation frequency plays an important role in human listening. For example, it has been observed that a human auditory system is more sensitive to modulation frequencies less than 20 Hz (*e.g.* [30] [31] [32]). On the other hand, very slowly changing components (*e.g.* less than 5 Hz) are usually related to noisy sources (*e.g.* [33] [34] [35]). In some articles (*e.g.* [2]), it has been noted that speaker specific information dominates for frequencies below 10Hz, while speaker independent information dominates higher frequencies. Based on these observations, researchers have tried to utilize modulation frequency information to enhance the speech recognition performance in noisy environments. Typical approaches use high-pass or band-pass filtering in either spectral, log-spectral, or cepstral domains.

In [2], Hirsch et al. investigated the effects of high-pass filtering of spectral envelopes of each subband. Unlike the RASTA (Relative Spectral) processing proposed by Hermansky in [3], Hirsch conducted high-pass filtering in the power domain. In [2], he compared the FIR filtering approach with the IIR filtering approach, and concluded that the latter approach is more effective. He used the following form of the first order IIR filtering:

$$H(z) = \frac{1 - z^{-1}}{1 - 0.7z^{-1}} \quad (2.9)$$

where λ is a coefficient adjusting the cut-off frequency.

This is a simple high-pass filter with a cut-off frequency at around $4.5Hz$.

It has been observed that on-line implementation of Log Spectral Mean Subtraction (LSMS) is largely similar to RASTA processing. Mathematically, the on-line mean log-spectral subtraction is equivalent to the on-line CMN:

$$\mu_L(m, l) = \lambda\mu_Y(m - 1, l) + (1 - \lambda)Y(m, l) \quad (2.10)$$

where $Y(m, l)$

$$Y(m, l) = P(m, l) - \mu_P(m, l) \quad (2.11)$$

This is also a high-pass filter like Hirsch’s approach, but the major difference is that Hirsch conducted the high-pass filtering in the power domain, while in the LSMS, subtraction is done after applying the log-nonlinearity.

Theoretically speaking, if we perform filtering in the power domain, it is helpful for compensating the additive noise effect, and if we conduct filtering in the log-spectral domain, it is better for reverberation [5].

RASTA processing in [3] is similar to the on-line cepstral mean subtraction or on-line LSMS. While the on-line cepstral mean subtraction is basically first order high pass filtering, RASTA processing is a bandpass processing motivated by the modulation frequency concept. This processing has been based on the observation that human auditory systems are more sensitive to modulation frequencies between 5 and 20 Hz . (*e.g.* [31] [32]). Thus, signal components outside this modulation frequency range are not likely to originate from speech. In RASTA processing, Hermansky proposed the following 4-th order bandpass filtering. Like the on-line CMN, RASTA processing is performed after nonlinearity is applied.

$$H(z) = 0.1z^4 \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \quad (2.12)$$

In his work [3], Hermansky showed that band-pass filtering approach results in better performance than high-pass filtering. In the original RASTA in (2.12), pole location is at $z = 0.98$; later, he mentioned that $z = 0.94$ seems to be optimal [3]. However, in some articles (*e.g.* [5]), it has been reported that the on-line CMN (which is a high-pass filtering) is slightly better performing than RASTA processing (which is a band-pass filtering) in speech recognition. As mentioned above, if we perform filtering after applying the log-nonlinearity, then it would be more helpful for reverberation, but it might not be very helpful additive noise.

Thus, Hermansky also proposed a variation of RASTA, which is called J-RASTA (or Lin-Log RASTA). By using the following function,

$$y = \log(1 + Jx) \quad (2.13)$$

this model has characteristics of both the linear model and the logarithmic nonlinearity.

2.7 Normalization Algorithm

In this section, we discuss some algorithms that are designed for enhancing robustness against noise. Many normalization algorithms work in the feature domain including Cepstral Mean Normalization (CMN), Mean Variance Normalization (MVN), , Code Dependent Cepstral Normalization (CDCN), and Histogram Normalization (HN). The original form of VTS (Vector Taylor Series) work in the log spectral domain.

2.7.1 CMN, MVN, HN, and DCN

The simplest way of performing normalization is using CMN or MVN. Histogram normalization (HN) is a generalization of MVN. CMN is the most basic form of noise compensation schemes, and it can remove the effects of linear filtering if the impulse response of the filter is shorter than the window length [36]. By assuming that the mean of each element of the feature vector from all utterances is the same, CMN is also helpful for additive noise as well. In equation form, CMN is expressed as follows:

$$\tilde{c}_i[j] = c_i[j] - \mu_{c_i}, \quad 0 \leq i \leq I - 1, 0 \leq j \leq J - 1 \quad (2.14)$$

where μ_{c_i} is the mean of the i -th element of the cepstral vector. In the above equation, $c_i[j]$ and $\tilde{c}_i[j]$ represent the original and normalized cepstral coefficient for the i -th element of the vector at the j -th frame index. I denotes the feature vector dimension and J denotes the number of frames in the utterance

MVN is a natural extension of CMN and is defined by the following equation:

$$\tilde{c}_i[j] = \frac{c_i[j] - \mu_{c_i}}{\sigma_{c_i}}, \quad 0 \leq i \leq I - 1, 0 \leq j \leq J - 1 \quad (2.15)$$

where μ_{c_i} and σ_{c_i} are the mean and standard deviation of the i -th element of the cepstral vector.

As mentioned in Subsection 2.6, CMN can be implemented as an on-line algorithm (*e.g.* [6] [37] [38]). In the on-line CMN, the mean of the cepstral vector is updated recursively.

$$\mu_{c_i}[j] = \lambda \mu_{c_i}[j - 1] + (1 - \lambda)c_i[j], \quad 0 \leq i \leq I - 1, 0 \leq j \leq J - 1 \quad (2.16)$$

This on-line mean is subtracted from the current cepstral vector.

As in RASTA and on-line log-spectral mean subtraction, the initialization of the mean value is very important in the on-line CMN. Otherwise, the performance would be significantly degraded (*e.g.* [5] [6]). It has been shown that using values obtained from the previous utterances is a good means of initialization. Another way is running a VAD to detect the first non-speech-to-speech transition (*e.g.* [6]). If the center of the initialization window coincides with the first non-speech-to-speech transition, then good performance is preserved, but it requires some delay.

In HN, we assume that the Cumulative Distribution Function (CDF) for an element of a feature is the same for all utterances.

$$\tilde{c}_i[j] = F_{c_i^{tr}}^{-1} \left(F_{c_i^{te}}(c_i[j]) \right) \quad (2.17)$$

In the above equation, $F_{c_i^{te}}$ denotes the CDF of the current test utterance and $F_{c_i^{tr}}^{-1}$ denotes the inverse CDF from the entire training corpus. Then, using (2.17), we can make the distribution of the element of the test utterance the same as that from the entire training corpus. We can also perform HN in a slightly different way by assuming that every element of the feature should follow a Gaussian distribution with zero mean and unit variance. In this case, $F_{c_i^{tr}}^{-1}$ is just the inverse CDF of the Gaussian distribution with zero mean and unity variance. If we use this approach, then the training database also needs to be normalized.

Recently, Obuchi showed that if we do apply histogram normalization on the delta cepstrum as well as the original cepstrum, the performance is better than the original HN [39]. This approach is called DCN (delta cepstrum normalization) [39].

Fig. 2.9 shows speech recognition experimental results on the RM1 database. First, we can observe that CMN provides significant benefit for noise robustness. MVN is performing somewhat better than CMN. Although HN is a very simple algorithm, it shows significant improvements in white noise and street noise environments. DCN shows the largest threshold shift among these algorithms. Fig. 2.10 shows the same kind of experiments conducted on WSJ0 5k test set. We used WSJ0-si84 for training.

Although these approaches show improvements in noisy environments, as shown in Fig. ??, these approaches are very sensitive to the silence length. This is because in these approaches, we assumed that all distributions are the same and if we prepend or append silences,

this assumption is no longer valid. As a consequence, DCN is doing better than Vector Taylor Series (VTS) in RM white and street noise environments, but the former is doing worse than the latter in WSJ0 5k experiment, which include more silences. VTS experimental results will be shown in the next subsection.

2.7.2 CDCN and VTS

More advanced algorithms include CDCN (Code Dependent Cepstral Normalization) and VTS (Vector Taylor Series). In this subsection, we will briefly review these techniques.

In CDCN and VTS, the underlying assumption is that speech is corrupted by unknown additive noise and linearly filtered by an unknown channel [40]. This assumption can be represented by the following equation:

$$\begin{aligned} P_z(e^{jw_k}) &= P_x(e^{jw_k})|H(e^{jw_k})|^2 + P_n(e^{jw_k}) \\ &= P_x(e^{jw_k})|H(e^{jw_k})|^2 \left(1 + \frac{P_n(e^{jw_k})}{P_x(e^{jw_k})|H(e^{jw_k})|^2} \right) \end{aligned} \quad (2.18)$$

Noise compensation can be done either in the log spectral domain [8] or in the cepstral domain [7]. In this subsection, we describe the compensation procedure in the log spectral domain. Let x , n , q , and z denote logarithms of the PSDs $P_x(e^{jw_k})$, $P_n(e^{jw_k})$, $|H(e^{jw_k})|^2$, and $P_z(e^{jw_k})$, respectively. For simplicity, we will remove the frequency index w_k in the following discussions. Then (2.18) can be expressed in the following form:

$$z = x + q + \log(1 + e^{n-x-q}) \quad (2.19)$$

This equation can be rewritten in the form of

$$z = x + q + r(x, n, q) = x + f(x, n, q) \quad (2.20)$$

where $f(x, n, q)$ is called the "environment function" [40].

Thus, our objective is inverting the effect of the environment function $f(x, n, q)$. This inversion consists of two independent problems. The first problem is estimating the parameters needed for the environment function. The second problem is finding the Minimum Mean Square Error (MMSE) estimate of x given z in (2.7.2).

In the CDCN approach, we assume that x is represented by the following Gaussian mixture and n and q are unknown constants.

$$f(x) = \sum_{k=0}^{M-1} c_k N(\mu_{x,k}, \Sigma_{x,k}) \quad (2.21)$$

we obtain \hat{n} and \hat{q} by maximizing the following likelihood.

$$(\hat{n}, \hat{q}) = \arg \max_{n,q} p(z|q, n) \quad (2.22)$$

The maximization of the above equation is performed using the Expectation Maximization (EM) algorithm. After obtaining \hat{n} and \hat{q} , \hat{x} is obtained in the Minimum Mean Square Error (MMSE) sense. In CDCN, we assume that n and q are constants for that utterance, so it cannot efficiently handle non-stationary noise [41].

In the VTS approach, we assume that the Probability Density Functions (PDF) of the log spectral density of clean utterance is represented by the GMM (Gaussian Mixture Model) and that of noise is represented by a single Gaussian component.

$$f(x) = \sum_{k=0}^{M-1} c_k N(\mu_{x,k}, \Sigma_{x,k}) \quad (2.23)$$

$$f(n) = N(\mu_n, \Sigma_n) \quad (2.24)$$

In this approach, we try to reverse the effect of the environment function in (). However, since this function is nonlinear, it is not easy to find an environmental function which maximizes the likelihood. This problem is tackled by using the first order Taylor series approximation. From (2.7.2), we consider the following first-order Taylor series expansion of the environment function $f(x, n, q)$. The resulting distribution z is also Gaussian if x follows the Gaussian distribution.

$$\begin{aligned} \mu_z = & E [x + f(n_0, x_0, q_0)] + E \left[\frac{\delta}{\delta x} f(x_0, n_0, q_0)(x - x_0) \right] \\ & E \left[\frac{\delta}{\delta n} f(x_0, n_0, q_0)(n - n_0) \right] + E \left[\frac{\delta}{\delta q} f(x_0, n_0, q_0)(q - q_0) \right] \end{aligned} \quad (2.25)$$

In a similar way, we also obtain the covariance matrix:

$$\begin{aligned} \Sigma_z &= \left(I + \frac{d}{dx} f(n_0, x_0, q_0) \right)^T \Sigma_x \left(I + \frac{d}{dx} f(n_0, x_0, q_0) \right) \\ &\quad \left(\frac{d}{dx} f(n_0, x_0, q_0) \right)^T \Sigma_n \left(\frac{d}{dx} f(n_0, x_0, q_0) \right) \end{aligned} \quad (2.26)$$

Using the above approximations of the mean and covariance of the Gaussian components, q , μ_n , and hence μ_z and Σ_z are obtained using the EM by maximizing the likelihood.

Finally, the feature compensation is conducted in the MMSE sense as shown below.

$$\hat{x}_{MMSE} = E[X|z] \quad (2.27)$$

$$= \int xp(x|z)dx \quad (2.28)$$

2.8 ZCAE and related algorithms

It has been long observed that a human being has a remarkable ability to separate the sound sources. Many works (*e.g.* [42]) have supported that binaural interaction plays an important role in sound source separation. For low frequencies, the use of Interaural Time Delay (ITD) is primarily used for sound source separation; for high frequencies, Interaural Intensity Difference (IID) plays an important role. This is because for high frequencies, space aliasing occurs, which prevents the use of the ITD.

In the ITD-based sound source separation approaches (*e.g.* [43] [16]), to avoid this space aliasing problem, we usually use a smaller distance between two microphones than the actual distance between two ears.

The conventional way of calculating the ITD is using a cross correlation after passing the signal through bandpass filters. In more recent works [16], it has been shown that the zero-crossing approach is more effective than the cross-correlation approach for accurately estimating the ITD. and results in better speech recognition results. This approach is called Zero Crossing Amplitude Estimation (ZCAE).

However, one critical problem of ZCAE is that the zero crossing point is heavily affected by in-phase noise and reverberation. Thus, as shown in [17] and [43], ZCAE did not show successful results in reverberant and omni-directional noise environments.

2.9 Discussion

While it is generally agreed that window length between 20 ms and 30 ms is appropriate for speech analysis, as mentioned in Section 2.2, there is no guarantee that this window length would be still optimal for noise estimation or noise compensation. Since the noise characteristics are usually stationary compared to speech, it is expected that longer windows might be better for noise compensation purposes. In this thesis, we will discuss what would be the optimal window length for noise compensation purposes. We note that even though longer duration windows may be used for noise compensation, we still need short duration windows for the actual speech recognition. In this these, we will discuss methods for doing so.

In Section 2.3, we discussed several different rate-level nonlinearities based on different data. Up until now, there has not been much discussion or analysis of the type of nonlinearity that is best for feature extraction. For a nonlinearity to be appropriate, it should satisfy some of the following characteristics:

- It should be robust against additive noise or reverberation.
- It should discriminate each phone reasonably well.
- The nonlinearity should be independent of the input sound pressure level, or at worst, a simple normalization should be able to remove the effect of the input sound pressure level.

Based on the above criteria, we will discuss in this thesis the nature of appropriate nonlinearities to be used for feature extraction.

We discussed conventional spectral subtraction techniques in Section 2.5. The problem with conventional spectral subtraction is that the structure is complicated and the performance depends on the accuracy of the VAD. Instead of using this conventional approach, since speech power changes faster than noise power, we can use the rate of power change as a measure for power normalization.

Although algorithms like VTS are very successful for stationary noise, they have some intrinsic problems. First, VTS is computationally heavy, since it is based on a large number of mixture components and an iterative EM algorithm, which is used for maximizing the

likelihood. Second, this model assumes that the noise component is modeled by a single Gaussian component in the log spectral domain. This assumption is reasonable in many cases, but it is not always true. A more serious problem is that the noise component is assumed to be stationary, which is not quite true for non-stationary noise, like music noise. Third, since VTS requires maximizing the likelihood using the values in the current test set, it is not straightforward to implement this algorithm for real-time applications.

Thus, in our thesis work, we will try to develop an algorithm more motivated by auditory observation, which requires small computation, and can be implemented as an on-line algorithm. Instead of trying to estimate the environment function and maximizing the likelihood, which is very computationally heavy, we will simply use the rate of power change of the test utterance.

The ZCAE algorithm described in Section 2.8 shows remarkable performance, however the performance improvement is very small in reverberant environments [17][43]. Another problem is that this algorithm requires large computation[43], since it needs bandpass filtering. Thus, we need to think about different approaches that would be more robust against reverberation. In our thesis, we will describe alternative approaches to tackle this problem.

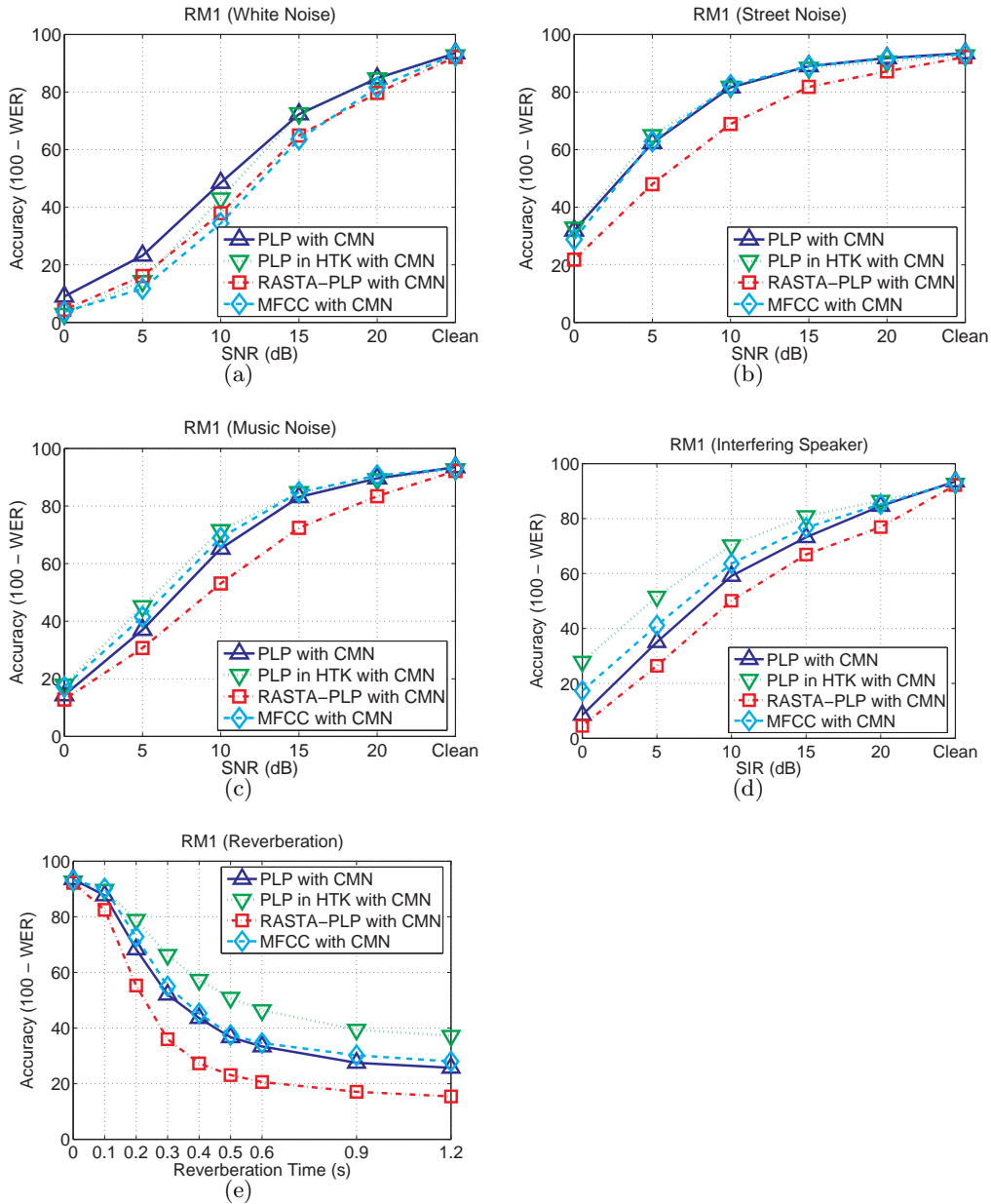


Fig. 2.5: Comparison between MFCC and PLP in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker, and (d) Reverberation

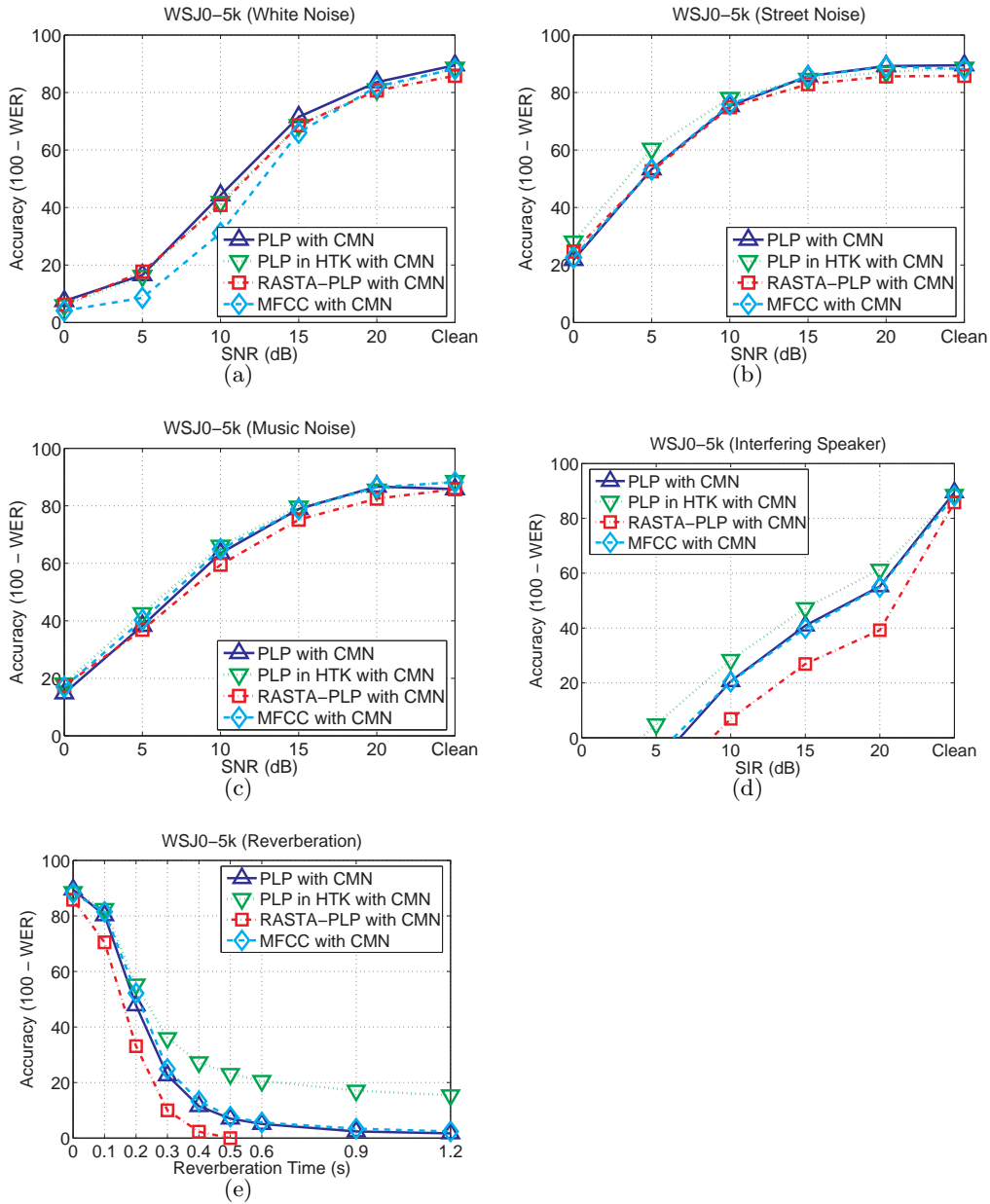


Fig. 2.6: Comparison between MFCC and PLP in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

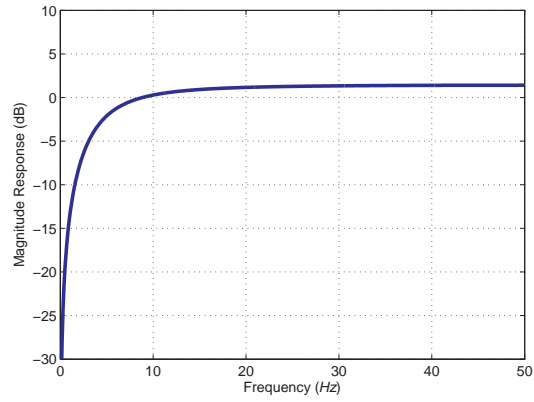


Fig. 2.7: The frequency response of the high-pass filter proposed by Hirsch et al. [2]

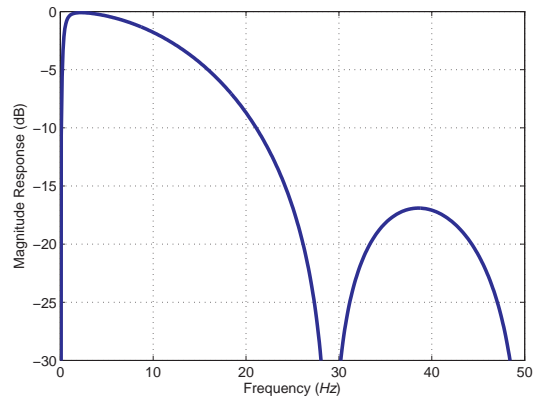


Fig. 2.8: The frequency response of the band-pass filter proposed by Hermansky et al. [3]

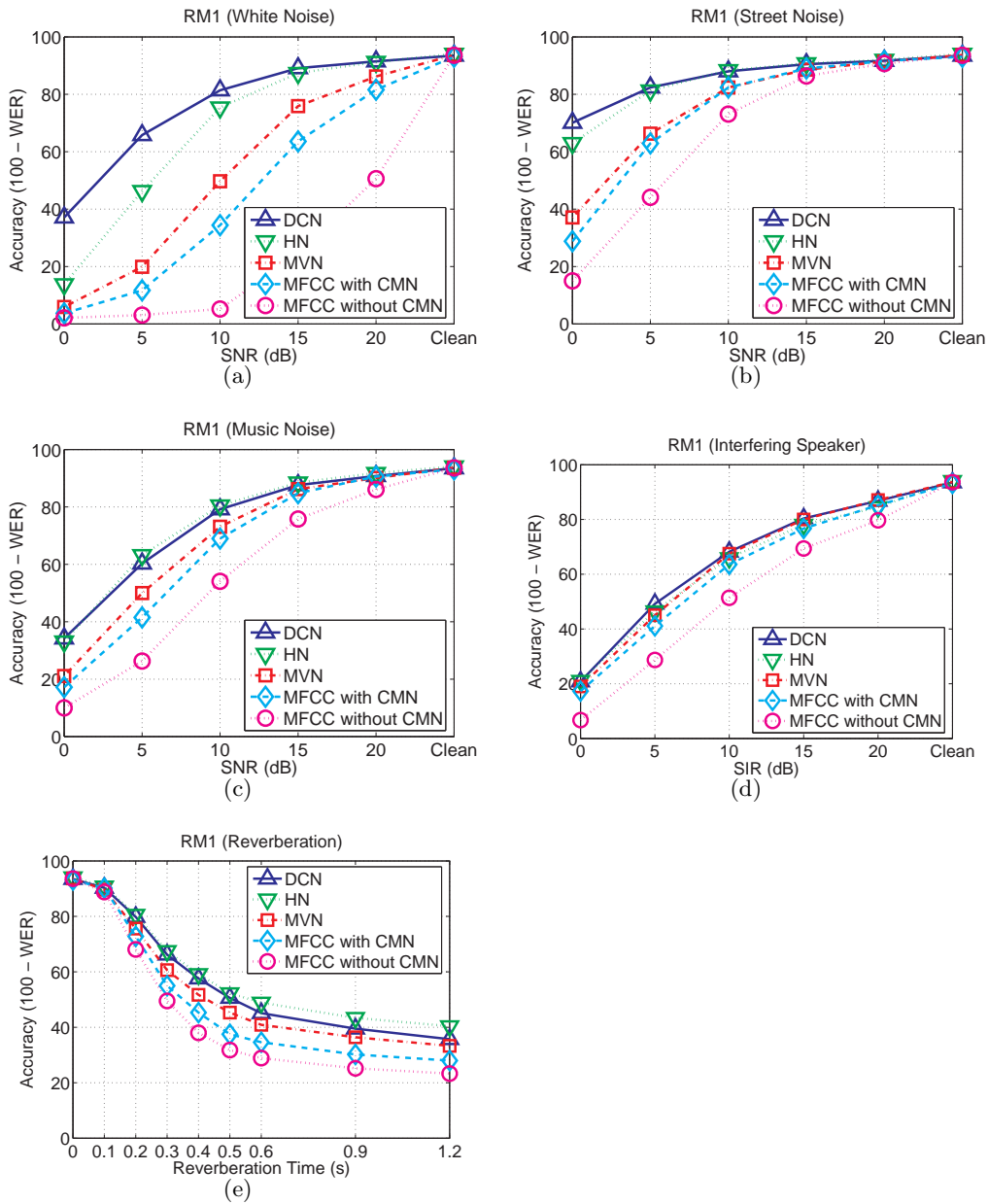


Fig. 2.9: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

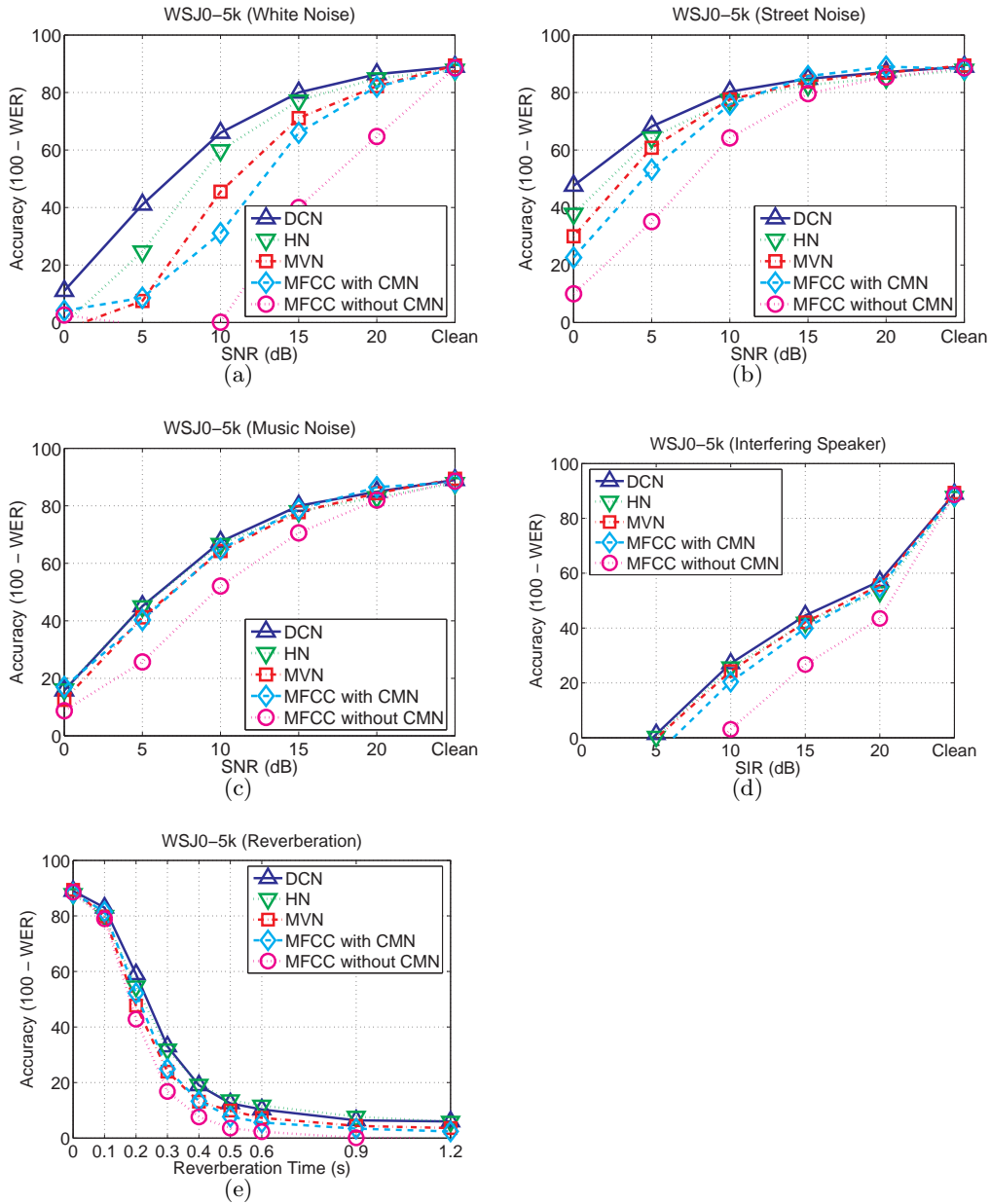


Fig. 2.10: Comparison between different normalization approaches in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

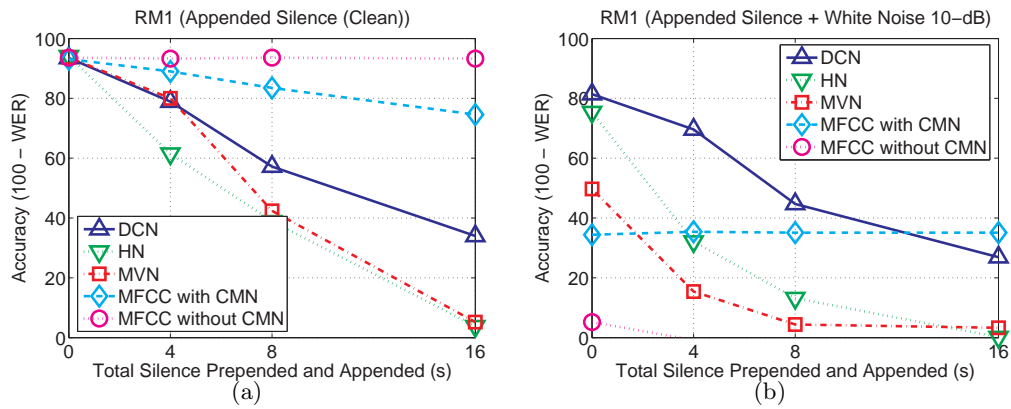


Fig. 2.11: : (a) Silence appended and prepended to the boundaries of clean speech (b) 10-dB of white Gaussian noise is added to the data used in (a)

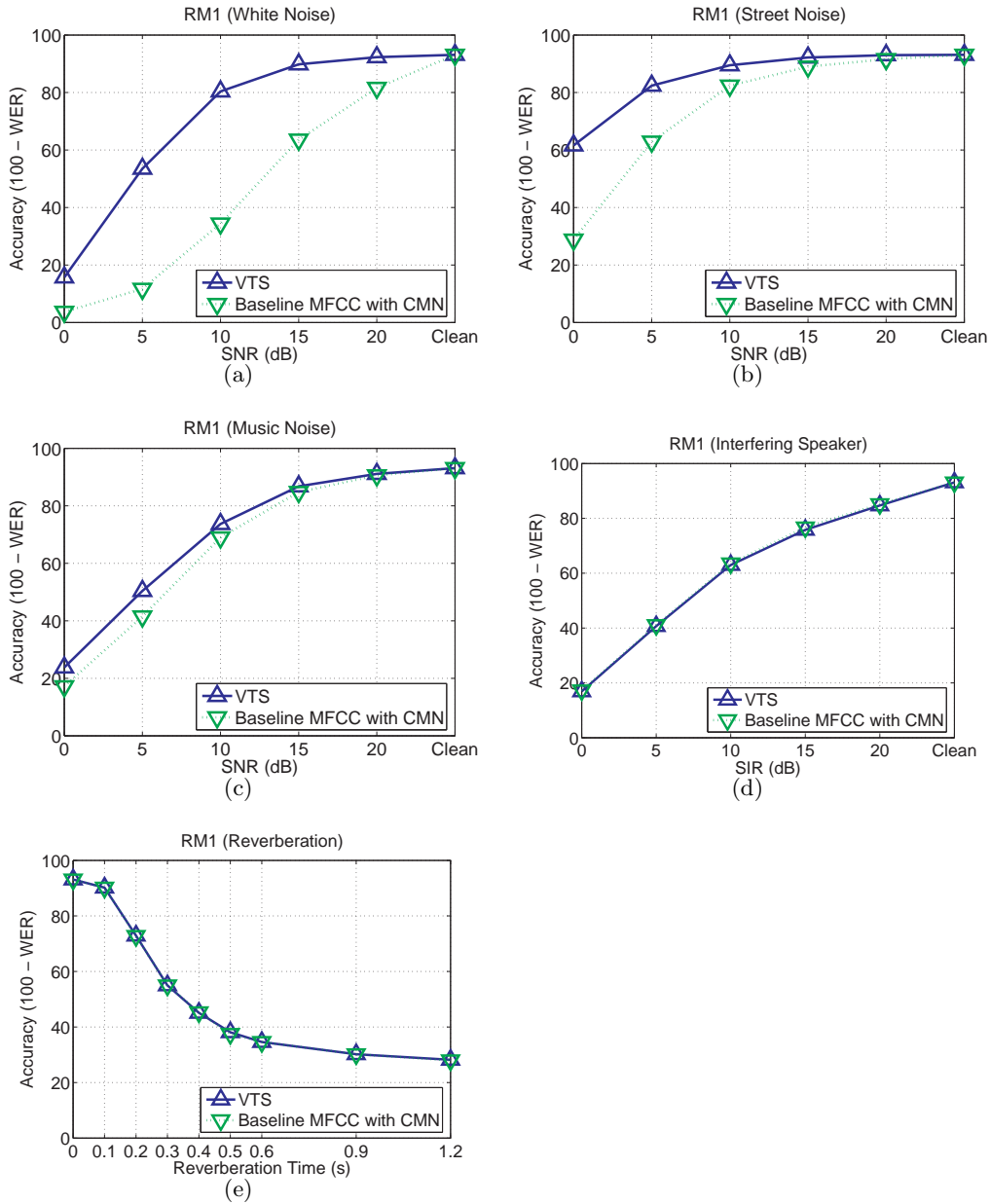


Fig. 2.12: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

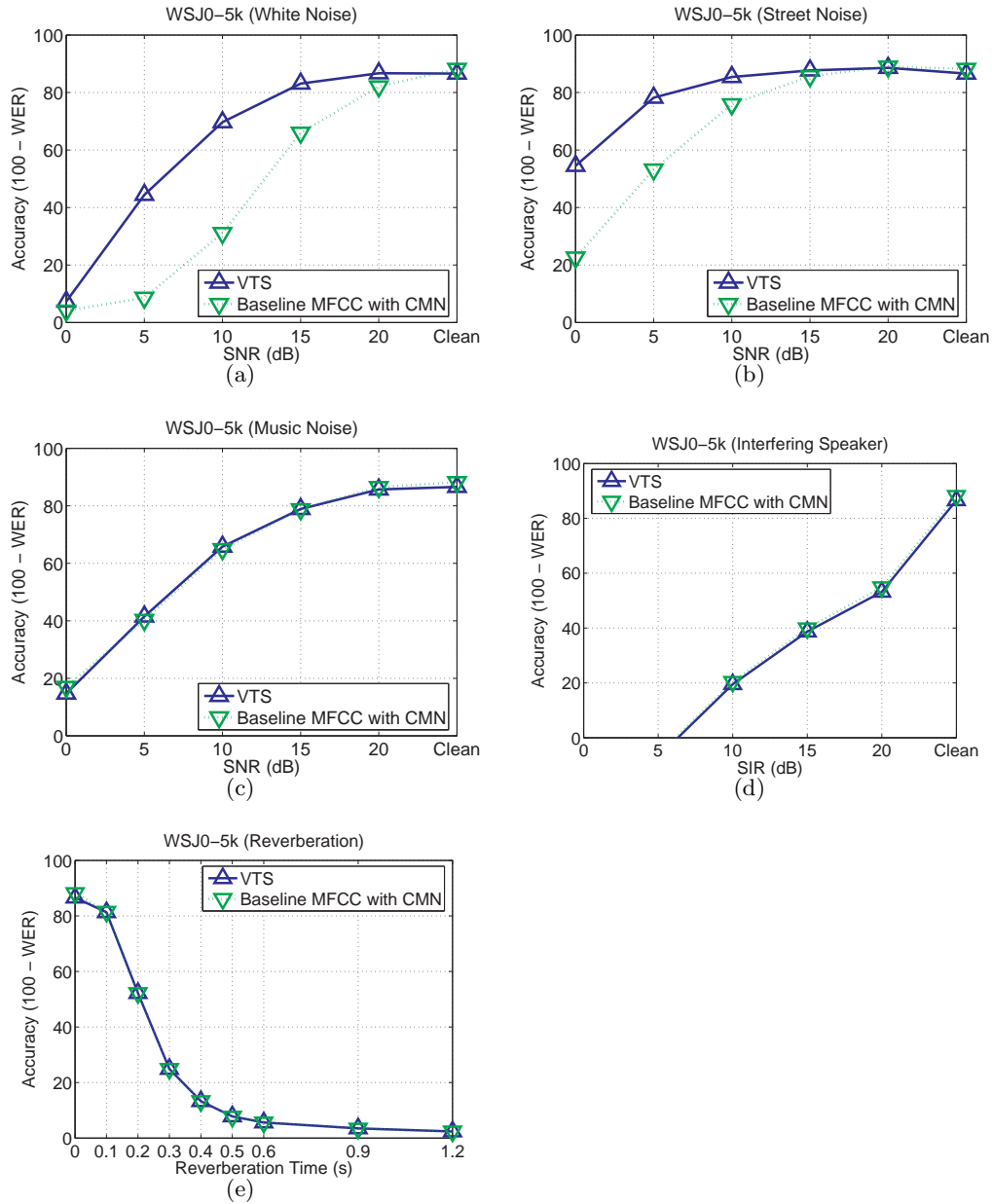


Fig. 2.13: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

3. TIME AND FREQUENCY RESOLUTION

It is a widely known fact that there is a trade-off between time-resolution and frequency-resolution when we select an appropriate window length for frequency-domain analysis (*e.g.* [24]). If we want to obtain better frequency domain resolution, then a longer window is more appropriate since the Fourier transform of a longer window is closer to a delta function in the frequency domain. However, a longer window is worse in terms of time-resolution, and this is especially true for highly non-stationary signals like speech. In speech analysis, we want the signal within a single window to be stationary. As a compromise between these tradeoffs, a window length between 20 ms and 30 ms has been widely used in speech processing [25].

Although a window of such short duration is suitable for analyzing speech signals, if a certain signal does not change very quickly, then a longer window will be better. If we use a longer window, then we can analyze the noise spectrum in a better way. Also from the large sample theory, if we use more data in estimating the statistics, then the variance of the estimation will be reduced. It is widely known that noise power changes more slowly than speech signal power; thus, based on the above discussion, it is quite obvious that longer windows might be better for estimating the noise power or noise characteristics. However, even if we use longer windows for noise compensation or normalization, we still need to use short windows for feature extraction. In this section, we discuss two approaches to accomplish this goal: the Medium-duration-window Analysis and Synthesis (MAS) method, and the Medium-duration-window Running Average (MRA) method.

When we need to estimate some unknown statistic, if we use more and more data to estimate it, then due to the large sample theory, the estimated statistic will have smaller variance, which results in better estimation. Above, we briefly mentioned this notion along the time-axis, but the same idea can be applied along the frequency axis as well.

Along with the window length, another important aspect in frequency domain analysis

is the integration (or weighting) of spectrum. In the analysis-and-synthesis approach, we perform frequency analysis by directly estimating parameters for each discrete-time frequency index. However, as will be explained later in more detail, we observe that the channel-weighting approach shows better performance. The reason for better performance with channel weighting is similar to the reason for better performance with the medium-duration window. If we use information from adjacent frequency indices, then we can estimate noise components more reliably due to averaging over frequencies.

For frequency integration (or weighting), we can think of several different weighting schemes such as triangular response weighting or gammatone response weighting. In this chapter, we discuss which weighting scheme is more helpful for speech recognition.

3.1 Time-frequency resolution trade-off in short-time Fourier analysis

Before discussing the medium-duration-window processing for robust speech recognition, we will review the time-frequency resolution trade-off in short-time Fourier analysis. This trade-off has been known for a long time and has been extensively discussed in many articles (*e.g.* [24]).

Suppose that we obtain a short-time signal $v[n]$ by multiplying a window signal $w[n]$ with the original signal $x[n]$. In the time domain, this windowing procedure is represented by the following equation:

$$v[n] = x[n]w[n] \tag{3.1}$$

In the frequency domain, it is represented by the following relation:

$$V(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) * W(e^{j\omega}) \tag{3.2}$$

Ideally, we want $V(e^{j\omega})$ to approach $X(e^{j\omega})$ as closely as possible. To achieve this goal, $W(e^{j\omega})$ needs to be close to the delta function in the frequency domain [24]. In the time domain, this corresponds to a constant value of $w[n] = 1$ with infinite duration. If the length of the window increases, then the magnitude spectrum becomes closer and closer to the delta function. Thus, we can see that a longer window results in better frequency resolution.

However, speech is a highly non-stationary signal, and in spectral analysis, we want to assume that the short-time signal $v[n]$ is stationary. If we increase the window length

to obtain better frequency resolution, then the statistical characteristics of $v[n]$ would be more and more time-varying, which means that we would fail to capture those time changes faithfully. Thus, to obtain better time resolution, we need to use a shorter window.

The above discussion is the well-known time-frequency resolution trade-offs. Due to this trade-offs, in speech processing, we usually use a window length between 20 *ms* and 30 *ms*.

3.2 Time Resolution for Robust Speech Recognition

In this section, we discuss two different ways of using the medium-duration window for noise compensation: the Medium-duration-window Analysis and Synthesis (MAS) method, and the Medium-duration-window Running Average (MRA) method. These methods enable us to use short windows for speech analysis while noise compensation is performed using a longer window. Fig 3.2.1. shows the block diagrams of the MAS and the MRA methods. The main objective of these approaches is the same, but they differ in how to obtain this objective. In the case of the MRA approach, frequency analysis is performed using short windows, but parameters are smoothed over time using a running average. Since frequency analysis is conducted using short-windows, features can be directly obtained without re-synthesizing the speech. In the case of the MAS approach, frequency analysis is performed using a medium-duration window, and after normalization, the waveform is re-synthesized. Using the re-synthesized speech, we can apply feature extraction algorithms using short windows. The idea of using a longer window is actually very simple and obvious; however, in conventional normalization algorithms, this idea has not been extensively used and theoretic analysis has not been thoroughly performed.

3.2.1 Medium-duration running average method

The block diagram for the running average method is shown in Fig. 3.4(f). In the MRA method, we segment the input speech by applying a short hamming window with a length between 20 ms and 30 ms, which is the length conventionally used in speech analysis.

Let us consider a certain type of variable for each time-frequency bin and represent it by $P[m, l]$, where m is the frame index, and l is the channel index. Then, the medium-duration variable $Q[m, l]$ is defined by the following equation:

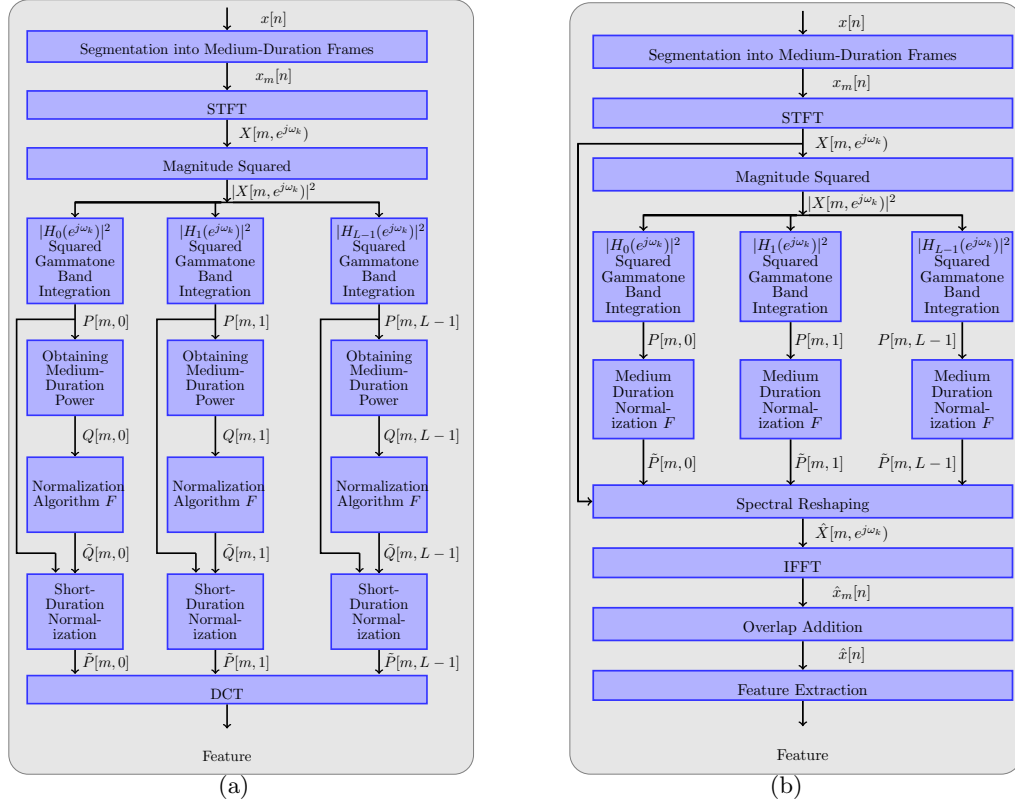


Fig. 3.1: (a) The block diagram of the Medium-duration-window Running Average (MRA) Method
 (b) The block diagram of the Medium-duration-window Analysis Synthesis (MAS) Method

$$Q[m, l] = \frac{1}{2M + 1} \sum_{m'=m-M}^{m+M} P[m, l] \quad \text{Averaging stage} \quad (3.3)$$

Averaging power of adjacent frames can be represented as a filtering operation with the following transfer function:

$$H(z) = \sum_{n=-M}^M z^{-n} \quad (3.4)$$

Thus, this operation can be considered to be a low pass filtering. The frequency response of the system is given by:

$$H(e^{j\omega}) = \frac{\sin\left(\left(\frac{2M+1}{2}\right)\omega\right)}{\sin\left(\frac{\omega}{2}\right)}, \quad (3.5)$$

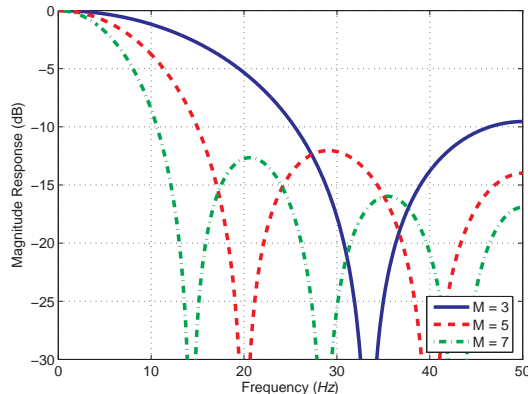


Fig. 3.2: Frequency response depending on the medium-duration parameter M

and these responses for different M values are shown in 3.2. However we observe that if we directly perform low-pass filtering, then it has the effect of making the spectrogram quite blurred, so in many cases, it induces the negative effects as shown in Fig. 3.3.

Thus, instead of performing normalization using the original power $P[m, l]$, we perform normalization on $Q[m, l]$. However, instead of directly using the normalized medium-duration power $\tilde{Q}[m, l]$ to obtain the feature, the weighting coefficient is multiplied with $P[m, l]$ to obtain the normalized power $\tilde{P}[m, l]$. This procedure is represented in the following equation:

$$\tilde{P}[m, l] = \frac{\tilde{Q}[m, l]}{Q[m, l]} P[m, l] \quad (3.6)$$

An example of MRA is the Power Bias Subtraction (PBS) algorithm, which is explained in Subsection 6.1.1. In the case of PBS, when we used a $25.6ms$ window length with a $10ms$ frame period, $M = 2 \sim 3$ showed the best speech recognition accuracy in noisy environments. So, this approximately corresponds to a window length of $75.6 \sim 85.6ms$.

3.2.2 Medium duration window analysis and re-synthesis approach

As mentioned before, the other strategy of using a longer window for normalization is the MAS method. The block diagram of this method is shown in Fig. 3.4(e). In this method, we directly apply a longer window to the speech signal to obtain a spectrum. From this spectrum, we perform normalization. Since we need to use features obtained from short windows, we

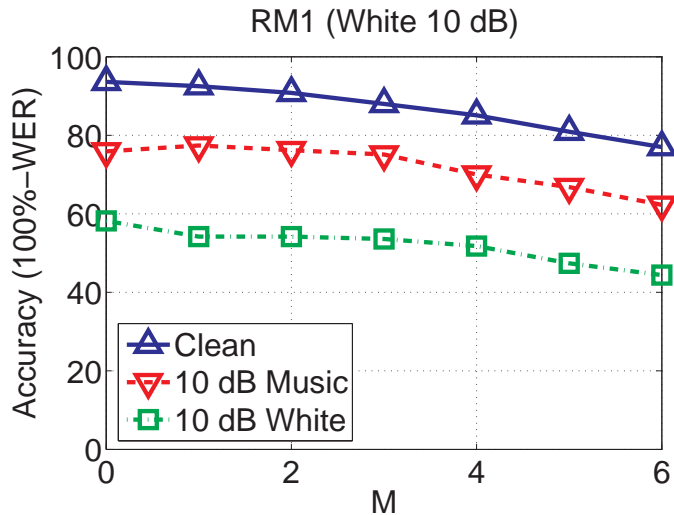


Fig. 3.3: Speech recognition accuracy depending on the medium-duration parameter M

cannot directly use the normalized spectrum from a longer window. Thus, a spectrum from a longer window needs to be re-synthesized using IFFT and the OverLap Addition (OLA) method. The Power-function-based Power Distribution Normalization (PPDN) algorithm, which is explained in Subsection 6.3, is based on this idea. This idea is also employed in Phase Difference Channel Weighting (PDCW), which is explained in Chapter 8. Even though PPDN and PDCW are unrelated algorithms, the optimal window length for noisy environments is around $75ms \sim 100ms$ in both algorithms.

3.3 Channel Weighting

3.3.1 Channel Weighting of Binary Parameters

In many cases, there are high correlations among adjacent frequencies, so performing channel weighting is helpful in obtaining more reliable information about noise and for smoothing purposes. This is especially true for a binary masking case. If we make a binary decision about whether a certain time-frequency bin is corrupted or not, then there should be some errors in the decision due to the limitation of a binary decision; the corruptness cannot be a binary value. Instead of using the decision from that particular time-frequency bin, if

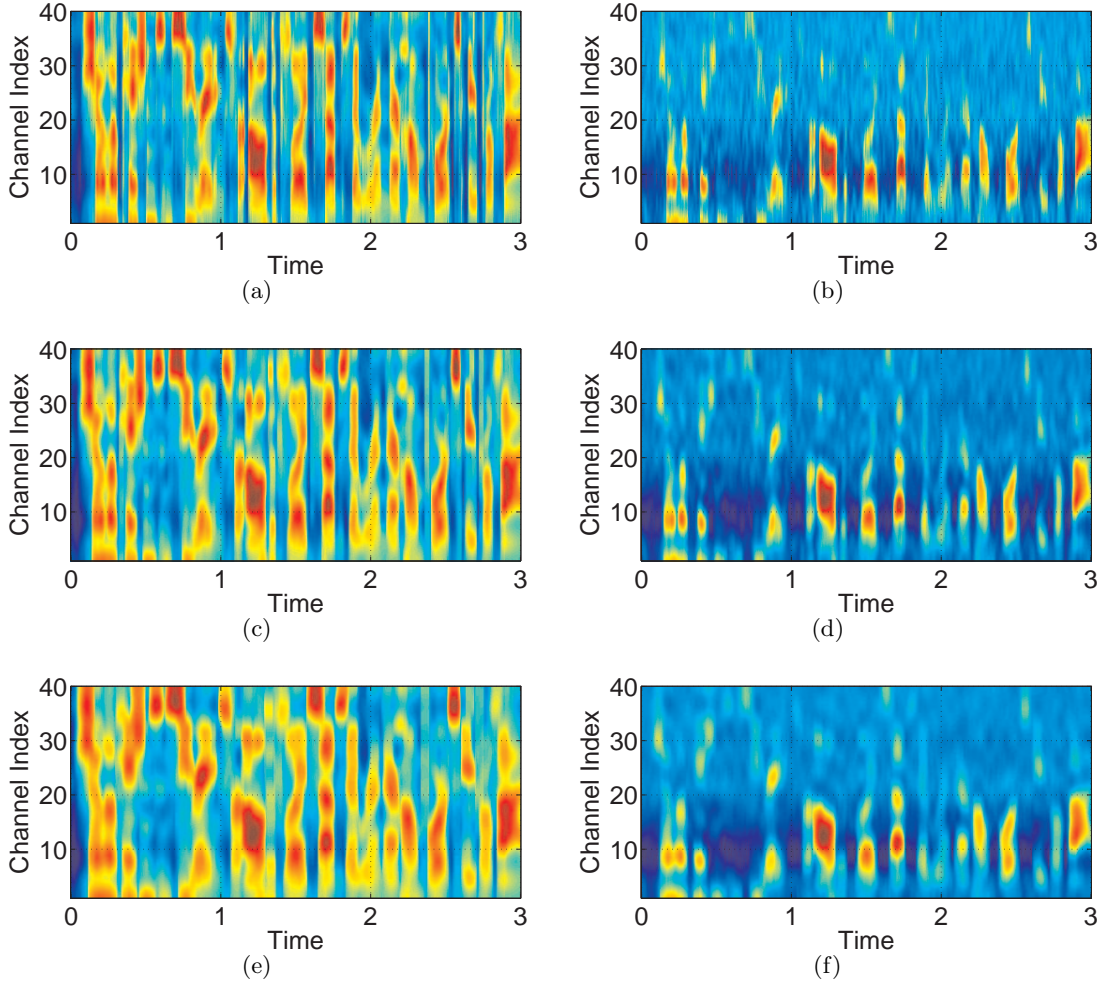


Fig. 3.4: (a) The spectrograms from clean speech with $M = 0$, (b) with $M = 2$, and (c) with $M = 4$
(d) The spectrograms from speech corrupted by 5 dB additive white noise with $M = 0$, (e)
with $M = 2$, and (f) with $M = 4$

we use a weighted average from adjacent channels, it is expected that we can obtain better performance.

Suppose that $\xi[m, k]$ is a parameter for the k -th frequency index at the m -th frame.

$$w[m, l] = \frac{\sum_{k=0}^{\frac{N-1}{2}} \xi[m, k] |X[m, e^{j\omega_k}] H_l(e^{j\omega_k})|}{\sum_{k=0}^{\frac{N-1}{2}} |X[m, e^{j\omega_k}] H_l(e^{j\omega_k})|} \quad (3.7)$$

where $X[m, e^{j\omega_k}]$ is the spectrum of the signal at this time-frequency bin and $H_l(e^{j\omega_k})$

is the frequency response of the i -th channel. Usually, the number of channels is much less than the FFT size. After obtaining the channel weighting coefficient $w[i, m]$ using (3.7), we obtain the smoothed weighting coefficient $\mu_g[k, m]$ using the following equation:

$$\mu_g[m, k] = \frac{\sum_{l=0}^{L-1} w[m, l] |H_l(e^{j\omega_k})|}{\sum_{l=0}^{L-1} |H_l(e^{j\omega_k})|} \quad (3.8)$$

Finally, the reconstructed spectrum is given by:

$$\tilde{X}[m, e^{j\omega_k}] = \max(\mu_g[m, k], \eta) X[m, e^{j\omega_k}] \quad (3.9)$$

where again η is a small constant used as a floor.

Using $\tilde{X}[k; m]$, we can re-synthesize speech using IFFT and OLA.

This approach has been used in Phase Difference Channel Weighting (PDCW) and the experimental results can be found in Chapter 8 of this thesis.

3.3.2 Weighting factor averaging across channels

In the previous section, we saw the channel weighting in the binary mask case. The same idea is applied for a continuous weighting case as well.

Suppose that we have a corrupt power $P[m, l]$ and enhanced power $\tilde{P}[m, l]$ for a certain time-frequency bin. As before, m is the frame index, and l is the channel index.

Instead of directly using $\tilde{P}[m, l]$ as the enhanced power, the weighting factor averaging scheme works as follows:

$$\hat{P}[m, l] = \left(1/(l_2 - l_1 + 1) \sum_{l'=l_1}^{l_2} \frac{\tilde{P}[m, l']}{P[m, l]} \right) P[m, l] \quad (3.10)$$

where $l_2 = \min(l + N, N_{ch} - 1)$ and $l_1 = \max(l - N, 0)$.

In the above equation, averaging is done using a rectangular window across frequencies. Instead of using the rectangular window, we can also consider the hamming or Bartlett windows. However, based on the actual speech recognition experiment, we could not observe substantial performance differences.

This approach has been used in the Power Normalized Cesptral Coefficient (PNCC) and Small Power Boosting (SPB). Experimental results can be found in Chapters 5 and 6.

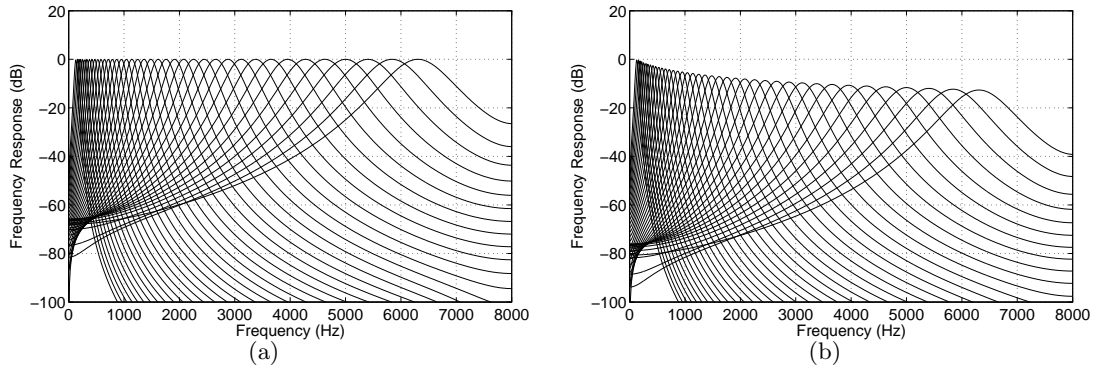


Fig. 3.5: (a) Gammatone Filterbank Frequency Response and (b) Normalized Gammatone Filterbank Frequency Response

3.3.3 Comparison between the triangular and the gammatone filter bank

In the previous subsection, we discussed obtaining performance improvement by using the channel-weighting scheme. Usually, in conventional speech feature extraction such as MFCC or PLP, frequency-domain integration has been already employed in the form of triangular or trapezoidal frequency response integration. In this section, we compare the triangular frequency integration and the gammatone frequency integration in terms of speech recognition accuracy. The gammatone frequency response is shown in Fig 3.5. This figure was obtained using Slaney’s auditory toolbox [44].

3.4 Proposed work

In this chapter, we discussed the effects of window length and channel weighting. We discussed performance improvement in terms of window length in several applications by repeating experiments using different window lengths. Thus, up until now, our discussion has been application-dependent and the optimal normalization window length has been selected empirically. As a proposed study, we will try to develop a more general theory on this topic by measuring modulation frequency of speech and some typical noise.

We will develop a missing feature reconstruction algorithm using longer window. In the current form of missing feature reconstruction algorithm, Gaussian Mixture Model (GMM) is obtained using short windows. Thus, reconstruction is also performed using short windows.

Based on the discussion in this chapter, we expect that if we use longer windows for reconstruction, then the result will be more reliable. Unlike conventional missing feature system, if we use longer windows, then we cannot directly obtain the feature. So, we will resynthesize speech and from this resynthesized speech, we will obtain features using conventional feature extraction systems.

4. AUDITORY NONLINEARITY

4.1 Introduction

In this chapter, we will discuss auditory nonlinearities and their role in robust speech recognition. The relation between the sound pressure level and the human perception has been studied for some time, and it is well explained in many literatures [45] [46]. These nonlinearity characteristics have been effectively used in many speech feature extraction systems. Inarguably, the most widely used features nowadays are either MFCC (Mel Frequency Cepstral Coefficient) or PLP (Perceptual Linear Prediction). In MFCC, we use logarithmic nonlinearity. PLP uses power-law nonlinearity, which is based on Steven's power law of hearing [26]. In this chapter, we will discuss the role of nonlinearity in feature extraction in terms of phone discrimination ability, noise robustness, and speech recognition accuracy in different noisy environments.

4.2 Human auditory nonlinearity

Human auditory nonlinearity has been investigated by many researchers. Due to the difficulty of conducting experiments on an actual human nerve, in many cases, researchers perform experiments on animals like cats [47], and the results were extrapolated to reflect human perception case [1]. Fig. 4.1 illustrates the simulation result of the relation between the average rate and the input SPL (Sound Pressure Level) for a pure sinusoidal input using the auditory model proposed by M. Heinz et al. [1]. In Fig. 4.1(a) and Fig. 4.1(b), we can see the intensity-rate relation at different frequencies obtained from the cat's nerve model and the human's nerve model. In this figure, especially in the human nerve model, this intensity-rate relation does not change significantly with respect to the frequency of the pure tone. Fig. 4.1(c) illustrates the relation averaged across frequencies in the human model. In Fig. 4.1(d),

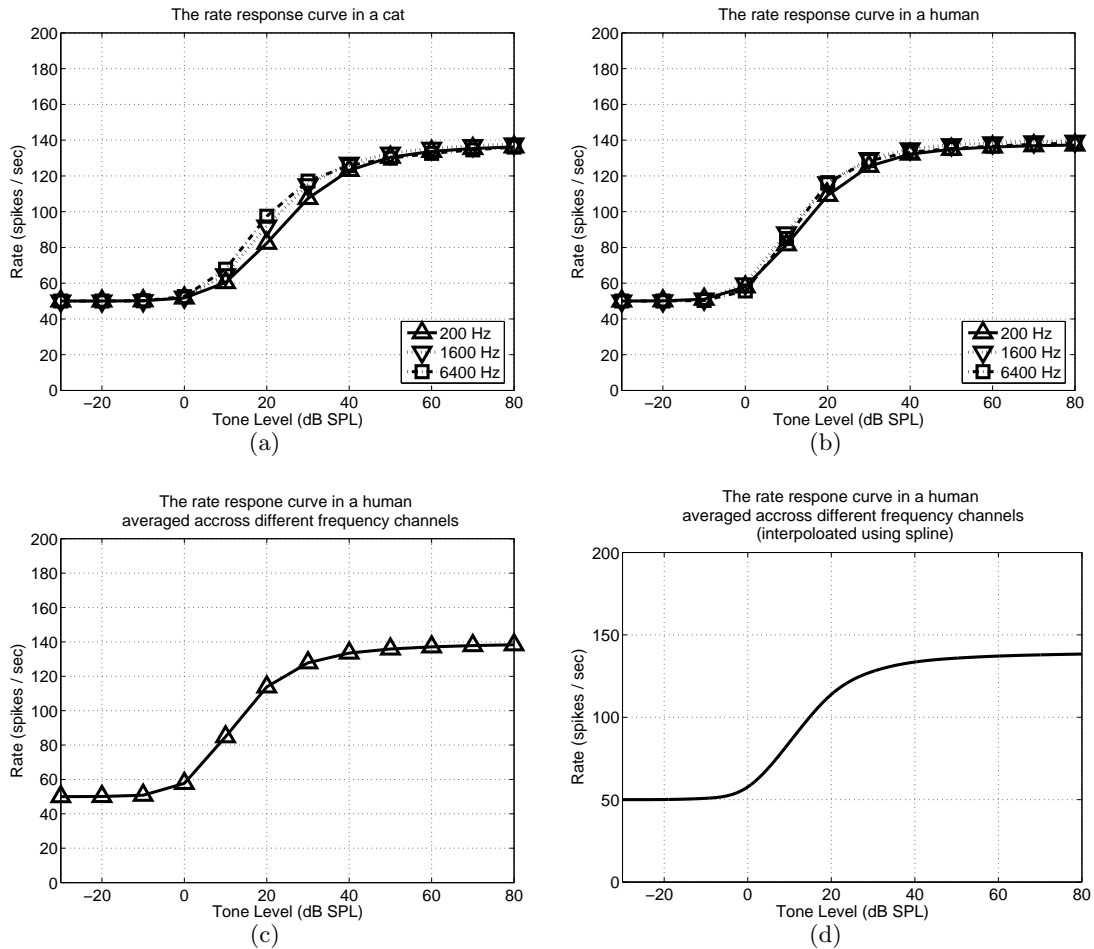


Fig. 4.1: The relation between the intensity and the rate. Simulation was done using the auditory model developed by Heinz. et al [4]: 4.1(a) shows the relation in a cat model at different frequencies. 4.1(b) shows the relation in a human model, and 4.1(c) shows the average across different channels, and 4.1(d) is the smoothed version of 4.1(c) using spline.

we can see the interpolated version of Fig. 4.1(c) using spline. In the discussion that follows, we will use the curve of Fig. 4.1(c) for a speech recognition experiment. As can be seen in Fig. 4.1(c) and Fig. 4.2, this curve can be divided into three distinct regions. If the input SPL (Sound Pressure Level) is less than 0 dB, then the rate is almost a constant, which is called a spontaneous rate. In the region between 0 dB and 20 dB, the rate linearly increases with respect to the input SPL. If the input SPL of the pure tone is more than 30 dB, then the rate curve is largely constant. The distance between the threshold and the saturation

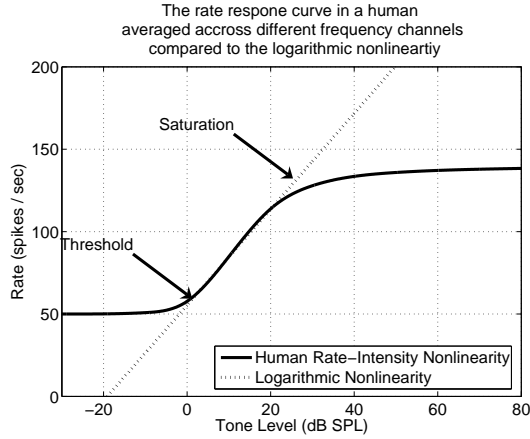


Fig. 4.2: The comparison between the intensity and rate response in the human auditory model [1] and the logarithmic curve used in MFCC. A linear transformation is applied to fit the logarithmic curve to the intensity-rate curve.

points are around 25 dB in SPL. As will be discussed later, this relatively short linear region causes problems in applying the original human rate-intensity curve to speech recognition systems.

In MFCC, we use logarithmic nonlinearity in each channel, which is given by the following equation

$$g(m, l) = \log_{10}(p(m, l)) \quad (4.1)$$

where $p(m, l)$ is the power for l -th channel index at time m and $g(m, l)$ is the nonlinearity output.

$$\eta(m, l) = 20 \log_{10} \left(\frac{p(m, l)}{p_{ref}} \right) \quad (4.2)$$

Thus, if we represent $g(m, l)$ in terms of $\eta(m, l)$, it appears as:

$$g(m, l) = \log_{10}(p_{ref}) + \frac{\eta(m, l)}{20} \quad (4.3)$$

From the above equation, we can see that the relation is just basically a linear function. In speech recognition, the coefficients of this linear equation are not important as long as we consistently use the same coefficient for the entire training and test utterances. If we

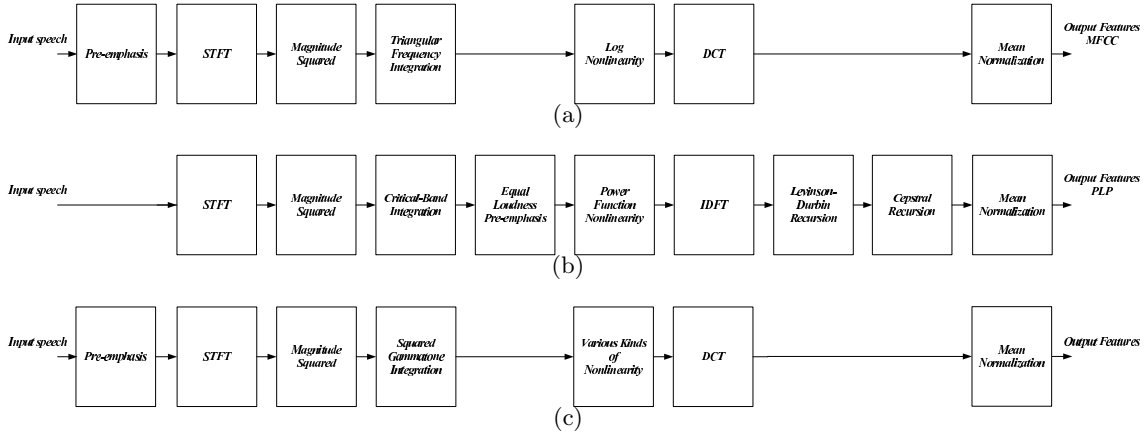


Fig. 4.3: The structure of the feature extraction system 4.3(a): MFCC, 4.3(b): PLP, and 4.3(c): General nonlinearity system

match this linear function to the linear region of Fig. 4.1(d), then we obtain Fig. 4.2. As is obvious from this figure, the biggest difference between logarithmic nonlinearity and the human auditory nonlinearity is that human auditory nonlinearity has threshold and saturation points. Because the logarithmic nonlinearity used in MFCC features does not exhibit threshold behavior, for speech segments of low power, the output of the logarithm nonlinearity can produce large output changes even if the changes in input are small. This characteristic, which can degrade speech recognition accuracy, becomes very obvious as the input approaches zero. If the power in a certain time-frequency bin is small, then even for a very small additive noise, the nonlinearity output will be very different. Hence, we can guess that the threshold point has a very important role for robust speech recognition.

In the following discussion, we will discuss the role of the threshold and the saturation points in actual speech recognition. Although the importance of auditory nonlinearity has been confirmed in several studies (*e.g.* [48]), there has been relatively little analysis concerning the effects of peripheral nonlinearities.

4.3 Speech recognition using different nonlinearities

In the following discussions, to test the effectiveness of different nonlinearities, we will use the feature extraction system shown in Fig 4.3(c) using different nonlinearities. For the

comparison test, we will also provide MFCC and PLP speech recognition results, which are shown in Fig. 4.3(a) and Fig. 4.3(b), respectively. Throughout this chapter, we will provide speech recognition experimental results by changing the nonlinearity in 4.3(c). For frequency domain integration, in MFCC, we use triangular frequency integration, and in PLP, we use critical band integration [49]. For the system in Fig 4.3(c), we use the gammatone frequency integration. In all of the following experiments, we used 40 channels. For the MFCC in Fig. 4.3(a) and the general feature extraction system in Fig. 4.3(c), a pre-emphasis filter of the form $H(z) = 1 - 0.97z^{-1}$ is applied first. The STFT analysis is performed using Hamming windows of duration 25.6 ms, with 10 ms between frames for a sampling frequency of 16 kHz. Both the MFCC and PLP procedures include intrinsic nonlinearities: PLP passes the amplitude-normalized short-time power of critical-band filters through a cube-root nonlinearity to approximate the power law of hearing [49, 50]. In contrast, the MFCC procedure passes its filter outputs through a logarithmic function.

4.4 Recognition results using human auditory nonlinearity and discussions

Using the structure shown in Fig. 4.3(c) and the nonlinearity shown in Fig. 4.2, we conducted speech recognition experiments using the CMU Sphinx 3.8 system with Sphinxbase 0.4.1. For training the acoustic model, we used SphinxTrain 1.0. For comparison purposes, we also obtained MFCC and PLP features using sphinx_fe and HTK 3.4, respectively. All experiments were conducted under the same condition, and delta and delta-delta components were appended to the original feature. For training and testing, we used subsets of 1600 utterances and 600 utterances, respectively, from the DARPA Resource Management (RM1) database. To evaluate the robustness of the feature extraction approaches, we digitally added three different types of noise: white noise, street noise, and background music. The background music was obtained from a musical segment of the DARPA Hub 4 Broadcast News database, while the street noise was recorded on a busy street. For reverberation simulation, we used the Room Impulse Response (RIR) software [51]. We assumed a room of dimensions $5 \times 4 \times 3$ m with a distance of 2m between the microphone and the speaker.

Since the rate-intensity curve is highly nonlinear, it is expected that if the speech power level is set to a different value, then the recognition result will also be different. Thus, we

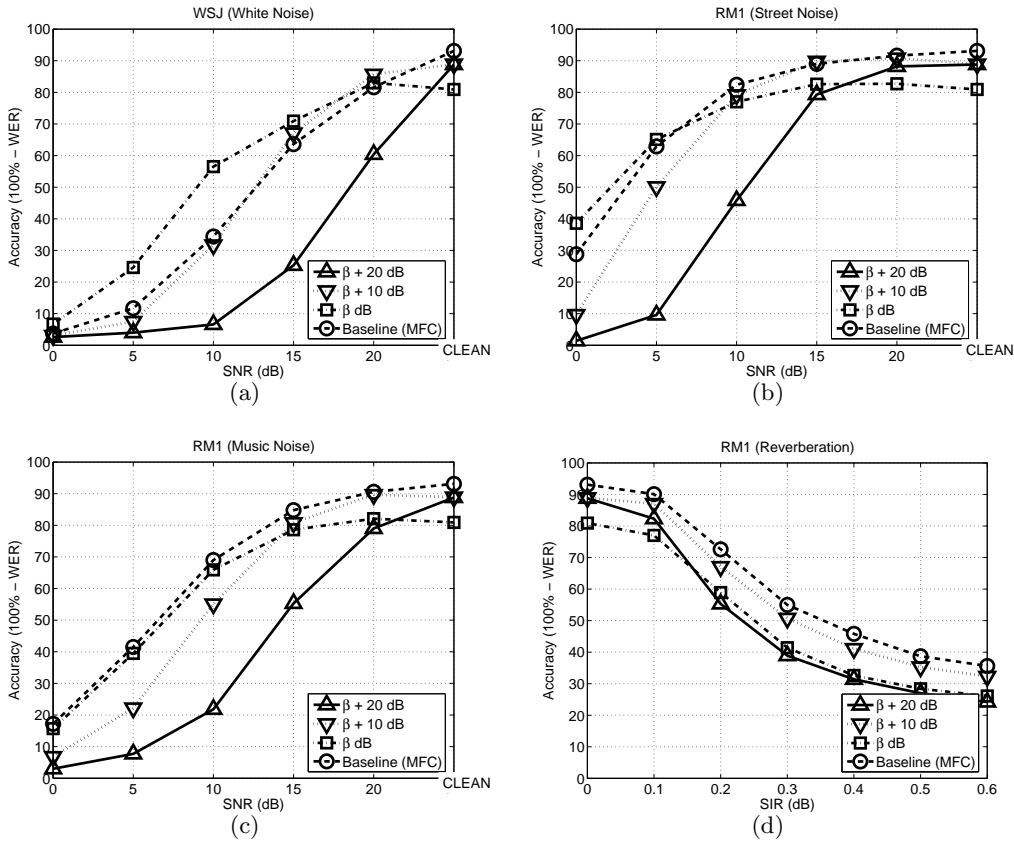


Fig. 4.4: Speech recognition accuracy obtained in different environments using the human auditory intensity-rate nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

conducted experiments at several different input SPL levels to check this effect. In Fig ??, β dB is the case where the average SPL falls slightly below the middle point of the linear region of the rate-intensity curve. By increasing the sound pressure level, we repeated experiments. For white noise, as shown in Fig. 4.4(a), if SPL is increased, then performance for noise is degraded, which is due to the fact that the portion benefited by the threshold part is reduced. For street noise, the performance improvement almost disappeared, and for music and reverberation, the performance is somewhat poorer than the baseline.

Fig. ?? illustrates the speech recognition experiment results using the curve shown in Fig. 4.2.

Up until now, we discussed the characteristics of the human intensity-rate curve and com-

pared it with the log nonlinearity curve used in the MFCC. We observe both the advantages and disadvantages of the human intensity-rate curve. The biggest advantage of the human intensity-rate curve compared to log nonlinearity is that it uses the threshold point. The threshold point induces significant improvement in noise robustness in the speech recognition experiments. However, one clear disadvantage is that the speech recognition performance changes significantly depending on the input sound pressure level. Thus, the optimal input sound pressure level needs to be obtained by experiments. Also, if we use a different input sound pressure level for training and testing, then due to the environmental mismatch, the recognition system works poorly.

4.5 Shifted Log Function and Power Function Approach

In the previous section, we saw that the human auditory intensity-rate curve is more robust against stationary additive noise. However, at the same time, it shows critical problems. The first problem is that the performance heavily depends on the speech sound pressure level, which is not a desirable characteristic. The optimal input sound pressure level needs to be obtained by empirical experiments or some discrimination criterion. Additionally, if there are mismatches between the input sound pressure level between the training and testing utterances, then the performance will degrade significantly. Still another problem is that even though the feature extraction system with this human intensity-rate curve shows improvement for stationary noisy environments, the performance is poorer than the baseline for high SNR cases. For highly non-stationary noise like music, it does not show improvements.

In the previous section, we argued that the threshold portion provides benefits compared to logarithmic nonlinearity. Then, one natural question is how the performance will look if we ignore the saturation portion and use only the threshold portion of the human auditory intensity-rate curve. This nonlinearity can be modeled by the following shifted-log as shown in Fig. 4.5. The shifted log function is represented by the following equation:

$$g(m, l) = \log_{10}(p(m, l) + \alpha P_{max}) \quad (4.4)$$

where P_{max} is defined to be the 95-th percentile of all $p(m, l)$. Depending on the choice of α , the location of the threshold point is changed.

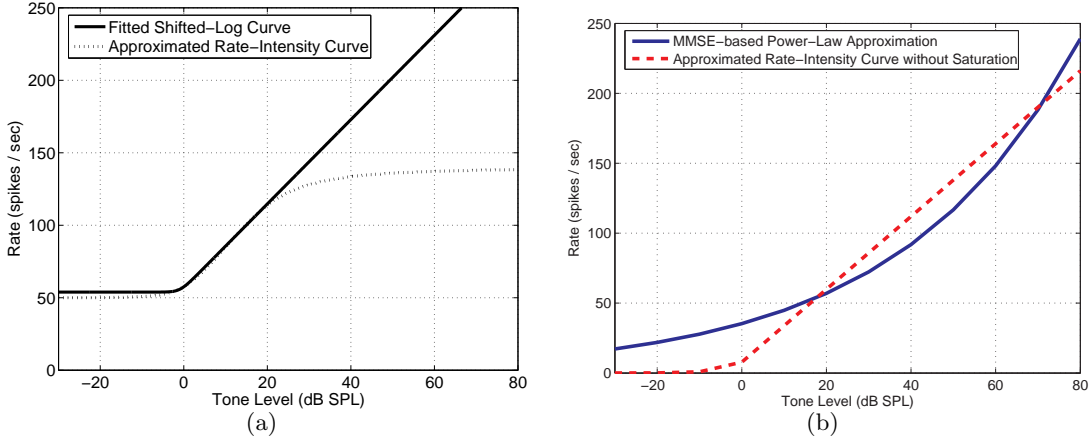


Fig. 4.5: 4.5(a) Rate-intensity curve and its stretched form in the form of shifted log 4.5(b) Power function approximation to the stretched form of the rate-intensity curve

The solid curve in Fig. 4.5(a) is basically a stretched version of the rate-intensity curve. The dotted curve in Fig. 4.5(b) is virtually identical to the solid curve in Fig. 4.5(a), but translated downward so that for small intensities the output is zero (rather than the physiologically-appropriate spontaneous rate of 50 spikes/s). The solid power function in that panel is the MMSE-based best-fit power function to the piecewise-linear dotted curve. The reason for choosing the power-law nonlinearity instead of the dotted curve in Fig. 4.5(b) is that the dynamic behavior of the output does not depend critically on the input amplitude. For greater input intensities, this solid curve is a linear approximation to the dynamic behavior of the rate-intensity curve between 0 and 20 dB. Hence, this solid curve exhibits threshold behavior but no saturation. We prefer to model the higher intensities with a curve that continues to increase linearly to avoid spectral distortion caused by the saturation seen in the dotted curve in the upper panel of Fig. 7.2. This nonlinearity, which is what is used in PNCC feature extraction, is described by the equation

$$y = x^{a_0} \quad (4.5)$$

with the best-fit value of the exponent observed to be between 1/10 and 1/15. We note that this exponent differs somewhat from the power-law exponent of 0.33 used for PLP features; this exponent is based on Steven's power law of hearing [50]. While our power-function nonlinearity may appear to be only a crude approximation to the physiological rate-intensity

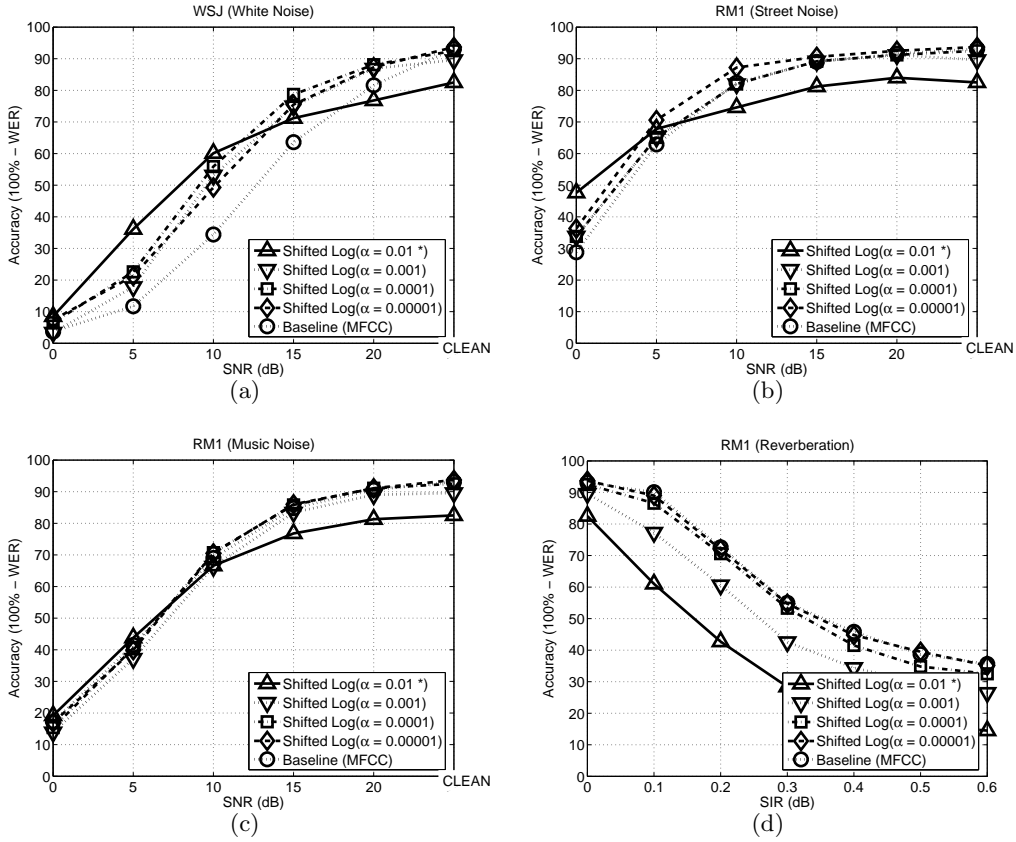


Fig. 4.6: Speech recognition accuracy obtained in different environments using the shifted log non-linearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

function, we will show in Sec. 7.3 that it provides a substantial improvement in recognition accuracy compared to the traditional log nonlinearity used in MFCC processing.

4.6 Speech Recognition Result Comparison of Several Different Nonlinearities

In this section, we will compare the performance of different nonlinearities explained in the previous sections. These nonlinearities include the human rate-auditory curve, its non-saturated model (shifted log), and the power function approach. As discussed earlier, the human intensity-rate curve depends on the sound pressure level of the utterance. On the other hand, the non-saturated model (shifted log) and power function model depend on their

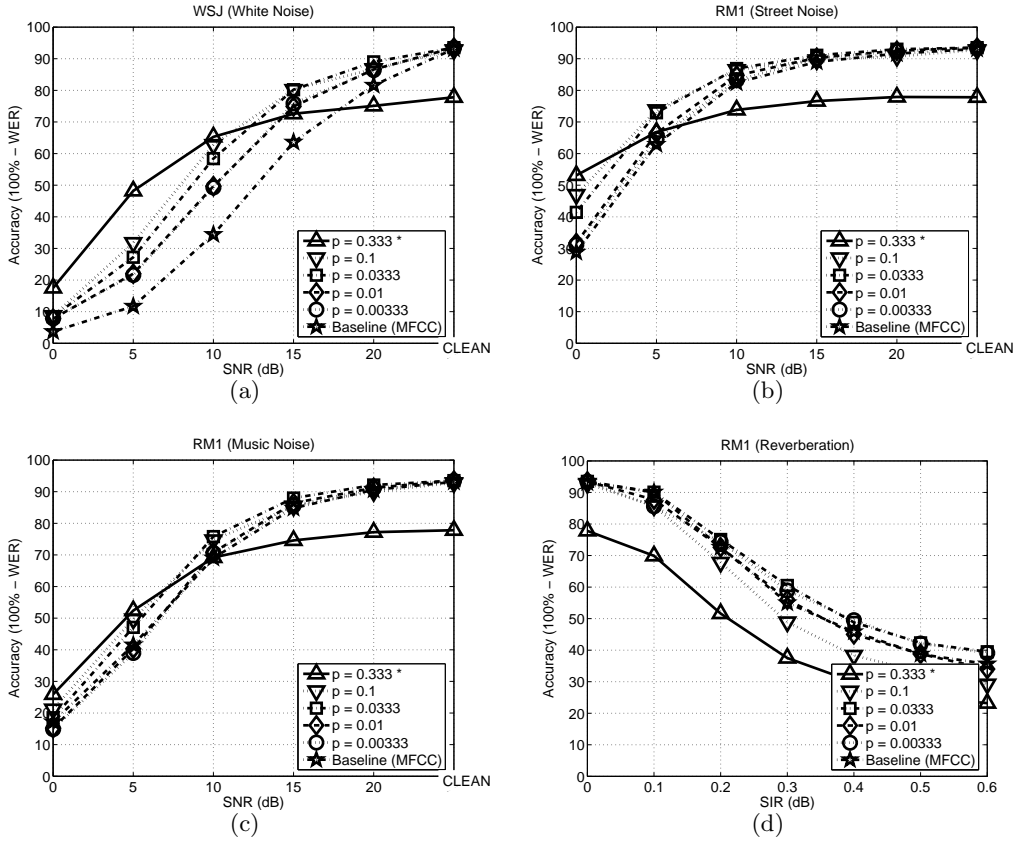


Fig. 4.7: Speech recognition accuracy obtained in different environments using the power function nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

intrinsic parameters. Thus, in comparing the performance of these algorithms, we selected those which showed reasonably good recognition performance in the previous results shown in Fig 4.4 , Fig 4.6, and Fig 4.7. Thus, in comparison, for the non-saturated model, we used

For white noise, as shown in Fig 4.8, there are not substantial differences in performance in terms of the threshold shift and the shift of around 5 dB is observed. Since the threshold point is the common characteristic of all of the three nonlinearities, we can infer that the threshold point plays an important role for additive noise. However, for high SNR cases, the human auditory intensity-rate nonlinearity falls behind other nonlinearities that do not use saturation, so we can see that the saturation point is actually harming the performance.

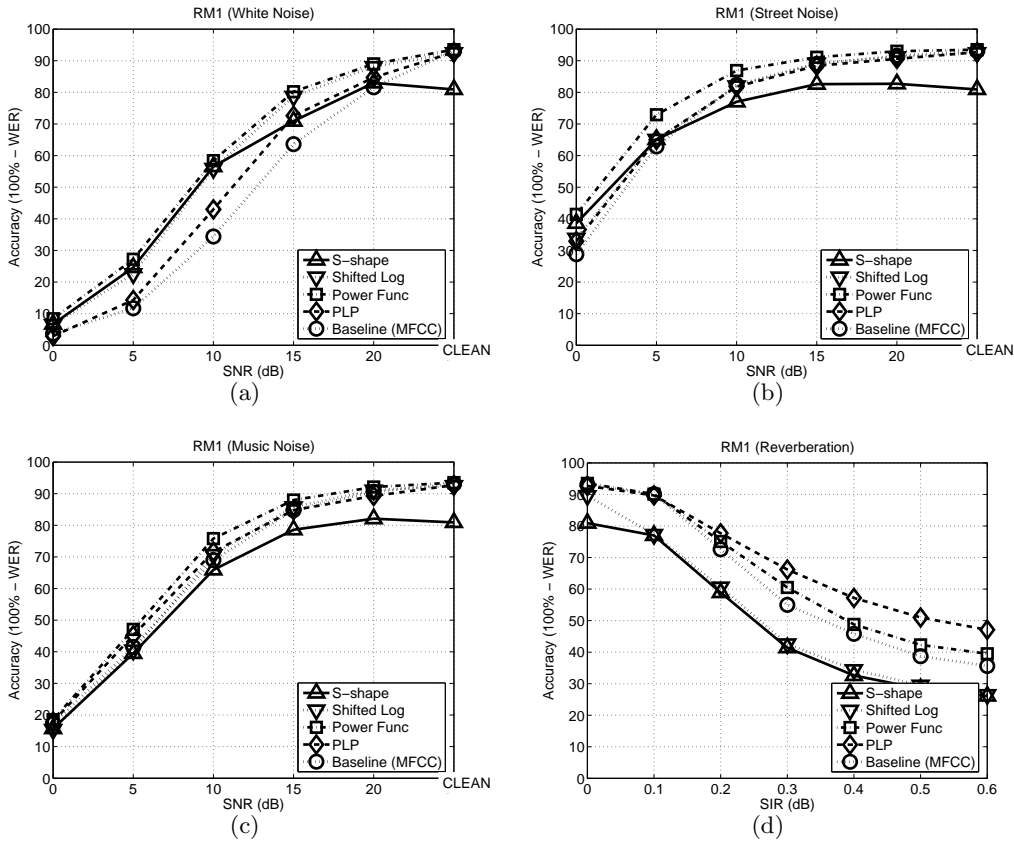


Fig. 4.8: Comparison of different nonlinearities (human rate-intensity curve, under different environments: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

This tendency of losing performance for high SNR is being observed in various kinds of noise shown in Fig 4.8. For the street and the music noise, the threshold shift is significantly reduced compared to the case of the white noise. The power function-based nonlinearity still shows some improvements compared to the baseline. In this figure, we can also note that even though PLP also uses the power function, it is not doing as well as the power function based feature extraction system described in this chapter. However, for reverberation, PLP shows better performance, as shown in Fig. 4.8(d).

4.7 *Proposed Work*

We will examine the trade-off between discrimination power and noise robustness of different nonlinearities. For discrimination power, we will use Fisher ratio obtained from spectra of each Context Independent (CI) phone. To measure the noise robustness, we will use the distortion after applying different nonlinearities. Using the statistical distribution information obtained from training database, we will measure the distortion.

5. SMALL POWER BOOSTING ALGORITHM

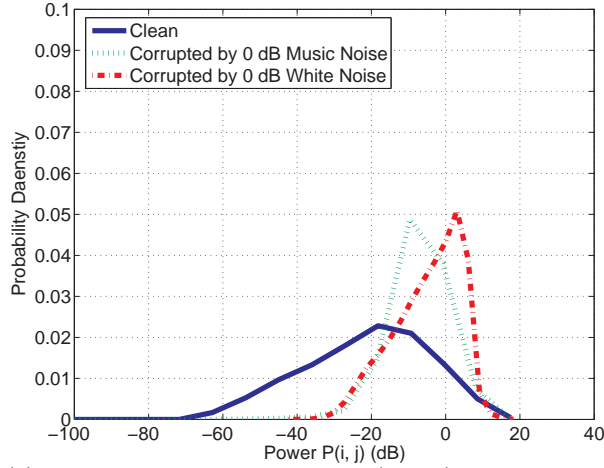
5.1 Introduction

Recent studies show that for non-stationary disturbances such as background music or background speech, algorithms based on missing features (*e.g.* [14, 52]) or auditory processing are more promising (*e.g.* [48, 53, 54, 23]). Still, the improvement in non-stationary noise remains less than the improvement that is observed in stationary noise. In previous work [53] and in the previous section, we also observed that the “threshold point” of the auditory nonlinearity plays an important role in improving performance in additive noise. Let us imagine a specific time-frequency bin with small power. Even if a relatively small distortion is applied to this time-frequency bin, due to the nature of compressive nonlinearity the distortion can become quite large.

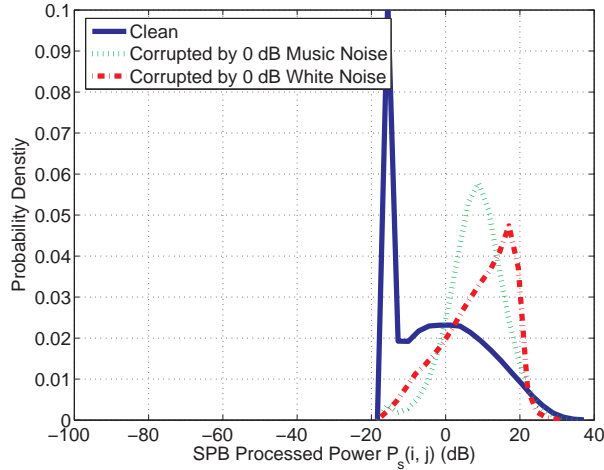
In this section, we explain the structure of the small boosting (SPB) algorithm in two different ways. In the first approach, we apply small power boosting to each time-frequency bin in the spectral domain, and then resynthesize speech (SPB-R). The resynthesized speech is fed to the feature extraction system. This approach is conceptually straightforward but less computationally efficient (because of the number of FFTs and IFFTs that must be performed). In the second approach, we use SPB to obtain feature values directly (SPB-D). This approach does not require IFFT operations and the system is consequently more compact. As we will discuss below, effective implementation of SPB-D requires smoothing in the spectral domain.

5.2 The Principle of Small Power Boosting

Before presenting the structure of the SPB algorithm, we first review how we obtain spectral power in our system, which is similar to the system in [43]. Pre-emphasis in the form of



(a) Probability Density Functions (PDFs) obtained with the conventional log nonlinearity



(b) Probability Density Functions (PDFs) obtained with the SPB with 0.02 power boosting coefficient in (5.2)

Fig. 5.1: Comparison of the Probability Density Functions (PDFs) obtained in three different environments : clean, 0-dB additive background music, and 0-dB additive white noise

$H(z) = 1 - 0.97z^{-1}$ is applied to an incoming speech signal sampled at 16 kHz. A short-time Fourier transform (STFT) is calculated using Hamming windows of a duration of 25.6 ms. Spectral power is obtained by integrating the magnitudes of the STFT coefficients over

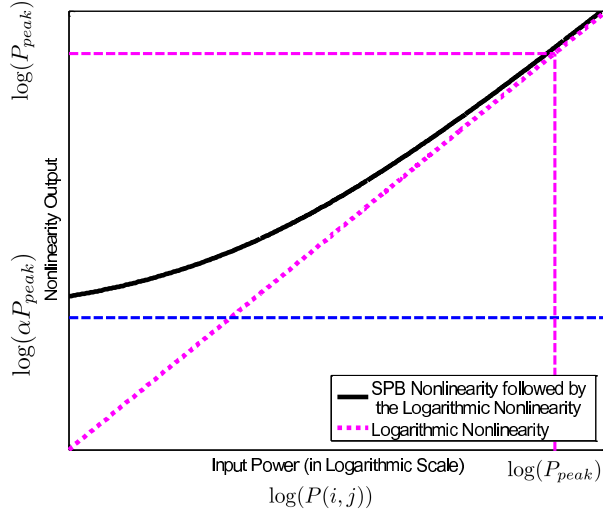


Fig. 5.2: The total nonlinearity consists of small power boosting and the subsequent logarithmic nonlinearity in the SPB algorithm

a series of weighting functions [55]. This procedure is represented by the following equation:

$$P(i, j) = \sum_{k=0}^{N-1} |X(e^{j\omega_k}; j)H_i(e^{j\omega_k})|^2 \quad (5.1)$$

In the above equation i and j represent the channel and frame indices respectively, N is the FFT size, and $H_i(e^{j\omega_k})$ is the frequency response of the i -th Gammatone channel. $X(e^{j\omega_k}; j)$ is the STFT for the j -th frame. w_k is defined by $\omega_k = \frac{2\pi k}{N}$, $0 \leq k \leq N - 1$.

In Fig. 5.1(a), we observe the distributions of $\log(P(i, j))$ for clean speech, speech in 0-dB music, and speech in 0-dB white noise. We used a subset of 50 utterances to obtain these distributions from the training portion of the DARPA Resource Management 1 (RM1) database. In plotting the distributions, we scaled each waveform to set the 95-th percentile of $P(i, j)$ to be 0 dB. We note in Fig. 5.1(a) that higher values of $P(i, j)$ are (unsurprisingly) less affected by the additive noise, but the values that are small in power are severely distorted by additive noise. While the conventional approach to this problem is spectral subtraction (*e.g.* [9]), this goal can also be achieved by intentionally boosting power for all utterances, thereby rendering the small-power regions less affected by the additive noise. We implement the SPB algorithm with the following nonlinearity:

$$P_s(i, j) = \sqrt{P(i, j)^2 + (\alpha P_{peak})^2} \quad (5.2)$$

We will call α the "small power boosting coefficient" or "SPB coefficient". P_{peak} is defined to be the 95-th percentile in the distribution of $P(i, j)$. In our algorithm, further explained in Subsection 5.3 and 5.3, after obtaining $P_s(i, j)$, either resynthesis or smoothing is performed. After that, the logarithmic nonlinearity follows. Thus, if we plot the entire nonlinearity defined by (5.2) and the subsequent logarithmic nonlinearity, then the total nonlinearity is represented by Fig. 5.2. Suppose that the power of clean speech at a specific time-frequency bin $P(i, j)$ is corrupted by additive noise ν . The log spectral distortion is represented by the following equation:

$$\begin{aligned} d(i, j) &= \log(P(i, j) + \nu) - \log(P(i, j)) \\ &= \log\left(1 + \frac{1}{\eta(i, j)}\right) \end{aligned} \quad (5.3)$$

where $\eta(i, j)$ is the Signal-to-Noise Ratio (SNR) for this time-frequency bin defined by:

$$\eta(i, j) = \frac{P(i, j)}{\nu} \quad (5.4)$$

Applying the nonlinearity of (5.2) and the logarithmic nonlinearity, the remaining distortion is represented by:

$$\begin{aligned} d_s(i, j) &= \log(P_s(i, j) + \nu) - \log(P_s(i, j)) \\ &= \log\left(1 + \frac{1}{\sqrt{\eta(i, j)^2 + \left(\frac{\alpha P_{peak}}{\nu}\right)^2}}\right) \end{aligned} \quad (5.5)$$

The largest difference between $d(i, j)$ and $d_s(i, j)$ occurs when $\eta(i, j)$ is relatively small. For small power regions even if ν is not large, $\eta(i, j)$ will become relatively large, and in (5.3), the distortion will diverge to infinity as $\eta(i, j)$ approaches zero. In contrast, in (5.5), even if $\eta(i, j)$ approaches zero, the distortion converges to $\log\left(1 + \frac{\nu}{\alpha P}\right)$.

Consider now the power distribution for SPB-processed powers. Fig. 5.1(b) compares the distributions for the same condition as Fig. 5.1(a). We can clearly see that the distortion is greatly reduced.

As can be seen, SPB reduces the spectral distortion and provides robustness to additive noise. However, as described in our previous paper [53], all nonlinearities motivated by human auditory processing, such as the "S"-shaped nonlinearity and the power-law nonlinearity

curves, also use this characteristic; however these approaches are less effective than the SPB approach described in the paper. The key difference, though, is that in other approaches, the nonlinearity is directly applied for each time-frequency bin. As will be discussed in Subsection 5.4, directly applying the non-linearity results in reduced variance for regions of small power, thus reducing the ability to discriminate small differences in power and finally, to differentiate speech sounds. We explain this issue in detail in Section 5.4.

5.3 Small Power Boosting with Re-synthesized Speech (SPB-R)

In this Subsection, we discuss the SPB system, which resynthesizes speech as an intermediate stage in feature extraction. The entire block-diagram for this approach is shown in Fig. 5.3. The blocks leading up to *Overlap-Addition* (OLA) are for small power boosting and resynthesizing speech, which is finally fed to conventional feature extraction. The only difference between the conventional MFCC features and our features is the use of the gammatone-shaped frequency integration with the equivalent rectangular bandwidth (ERB) scale [22] instead of the triangular integration with the MEL scale [20]. The advantages of gammatone-integration are described in [53], where gammatone-based integration was found to be more helpful in additive noise environments. In our system we use an ERB scale with 40 channels spaced between 130 Hz and 6800 Hz. From (5.2), the weighting coefficient $w(i, j)$ for each time-frequency bin is given by:

$$w(i, j) = \frac{P_s(i, j)}{P(i, j)} = \sqrt{1 + \left(\frac{\alpha P_{peak}}{P(i, j)}\right)^2} \quad (5.6)$$

Using $w(i, j)$, we apply the spectral reshaping expressed in [43]:

$$\mu_g(k, j) = \frac{\sum_{i=0}^{I-1} w(i, j) |H_i(e^{j\omega_k})|}{\sum_{i=0}^{I-1} |H_i(e^{j\omega_k})|} \quad (5.7)$$

where I is the total number of channels, and k is the discrete frequency index. The reconstructed spectrum is obtained from the original spectrum $X(e^{j\omega_k}; j)$ by using $\mu_g(k, j)$ in (5.7) as follows:

$$X_s(e^{j\omega_k}; j) = \mu_g(k, j)X(e^{j\omega_k}; j) \quad (5.8)$$

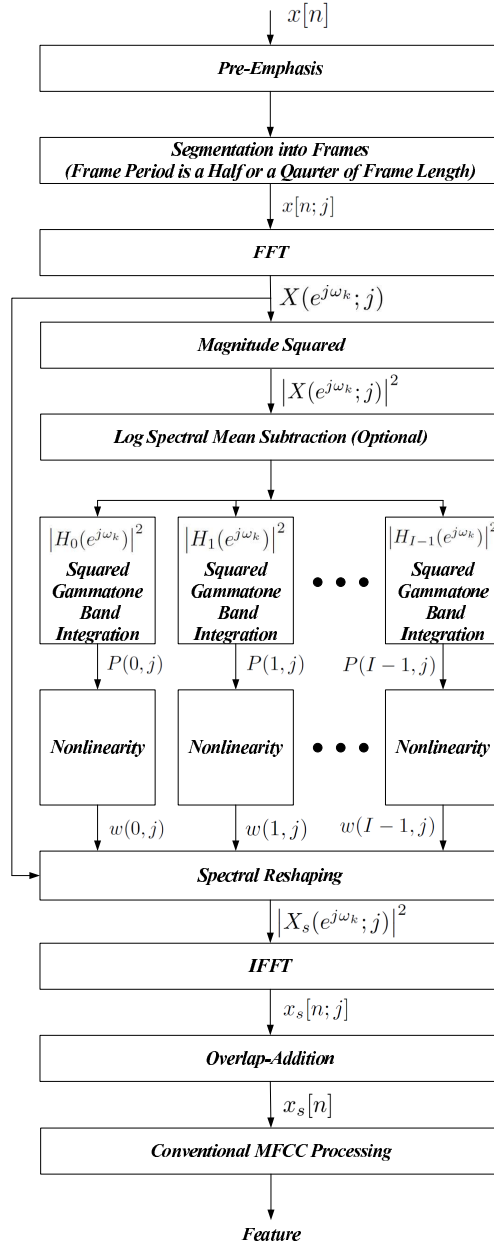


Fig. 5.3: Small power boosting algorithm which resynthesizes speech (SPB-R). Conventional MFCC processing is followed after resynthesizing the speech.

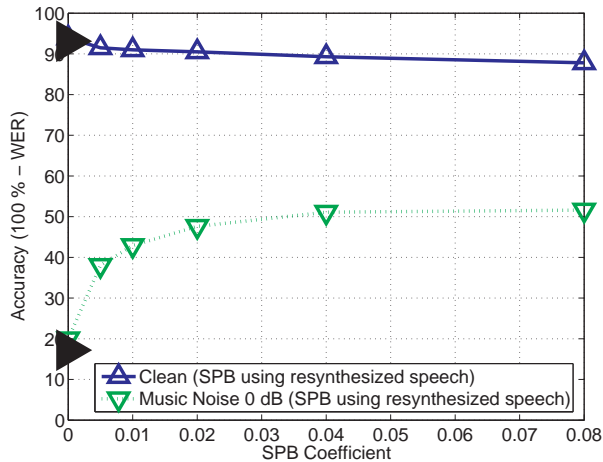


Fig. 5.4: Word error rates obtained using the SPB-R algorithm as a function of the value of the SPB Coefficient. The filled triangles at the y-axis represent the baseline MFCC performance for clean speech (upper triangle) and for additive background music noise at 0 dB SNR (lower triangle), respectively.

Speech is resynthesized using $X_s(e^{j\omega_k}; j)$ by performing IFFT and using OLA with hamming windows of 25 ms duration and 6.25 ms intervals between adjacent frames, which satisfy the OLA constraint for undistorted reconstruction. Fig. 5.4 plots the WER against the SPB coefficient α . The experimental configuration is as described in Subsection 5.6. As can be seen in that figure, increasing the boosting coefficient results in much better performance for highly non-stationary noise even at 0 dB SNR; while losing some performance for the clean environment. Based on that trade-off between the clean and noisy performance, we may select the SPB coefficient α in 0.01-0.02.

5.4 Small Power Boosting with Direct Feature Generation (SPB-D)

In the previous Subsection we discussed the SPB-R system which resynthesizes speech as an intermediate step. Because resynthesizing the speech is quite computationally costly, we discuss in this Subsection an alternate approach that generates SPB-processed features

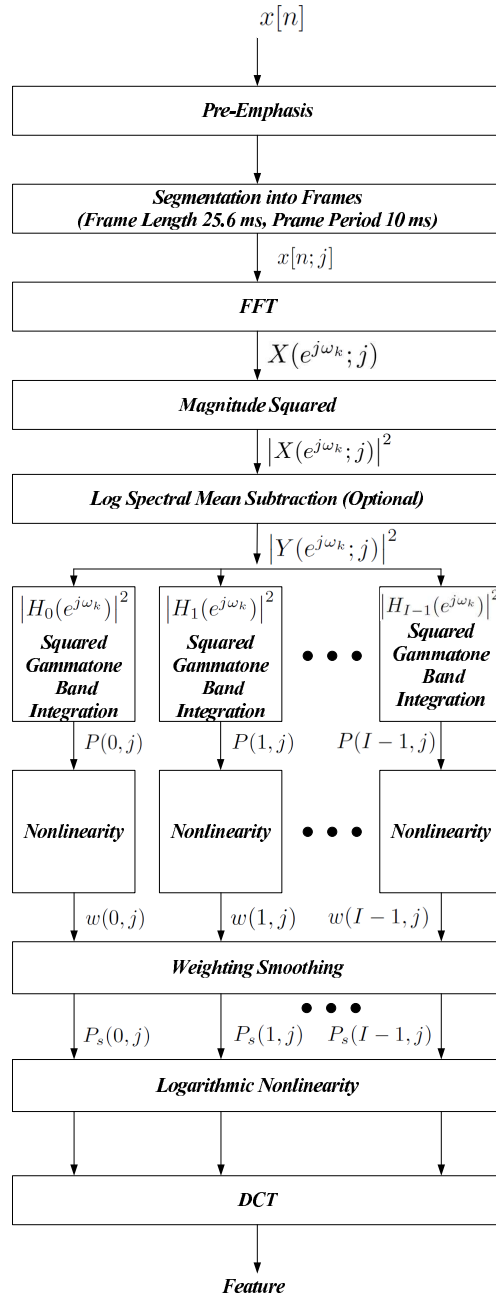


Fig. 5.5: Small power boosting algorithm with direct feature generation (SPB-D)

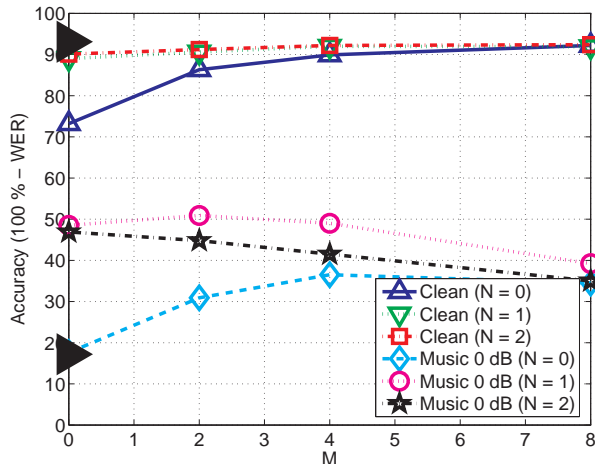


Fig. 5.6: The effects of weight smoothing on performance of the SPB-D algorithm for clean speech for speech corrupted by additive background music at 0 dB. The filled triangles at the y-axis represent the baseline MFCC performance for clean (upper triangle) and 0 dB additive background music (lower triangle) respectively. The SPB coefficient α was 0.02.

without the resynthesis step. A direct approach towards that end would be to simply apply the Discrete Cosine Transform (DCT) to the SPB-processed power $P_s(i, j)$ terms in (5.2). Since this direct approach is basically a feature extraction system itself, it will of course require that the window length and frame period used for segmentation into frames for SPB processing be the same values as are used in conventional feature extraction. Hence we use a window length of 25.6 ms with 10 ms between successive frames. We refer to this direct system as Small Power Boosting with Direct Feature Generation (SPB-D), and it is illustrated in Fig. 5.5.

Comparing the WER corresponding to $M = 0$ and $N = 0$ in Fig. 5.6 to the performance of SPB-R in Fig. 5.4), it is easily observed that SPB-D in the original form described above performs far worse than the SPB-R algorithm. These differences in performance are reflected in the corresponding spectrograms, as can be seen by comparing Fig. 5.7(c) to the SPB-R-derived spectrogram in Fig. 5.7(b)). In Fig. 5.7(c), the variance in small power regions is very small (concentrated at αP_{peak} in Fig. 5.2 and (5.2)), thus losing the power to discriminate sounds which have small power. Small variance is harmful in this context because PDFs in the

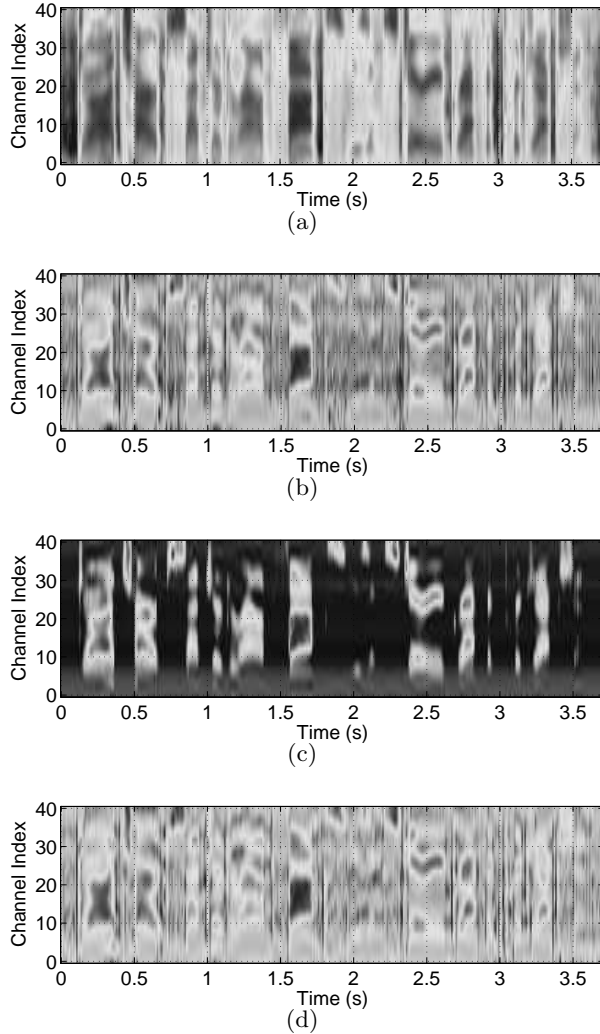


Fig. 5.7: Spectrograms obtained from a clean speech utterance using different processing: (a) conventional MFCC processing, (b) SPB-R processing, (c) SPB-D processing without any weight smoothing, and (d) SPB-D processing with weight smoothing $M = 4, N = 1$ in (5.9). A value of 0.02 was used for the SPB coefficient α . (5.2)

training data will be modeled by Gaussians with very narrow peaks. As a consequence small perturbation in the feature values from their means lead to large changes in log-likelihood scores. Hence we should avoid variances that are too small in magnitude.

We also note that there exist large overlaps in the shape of gammatone-like frequency

responses, as well as an overlap between successive frames. Thus, the gain in one time-frequency bin is correlated with that in an adjacent time-frequency bin. In the SPB-R approach, similar smoothing was achieved implicitly by the spectral reshaping from (5.7) and (8.10), and in the OLA process. With the SPB-D approach the spectral values must be smoothed explicitly.

Smoothing of the weights can be done horizontally (along time) as well as vertically (along frequency). The smoothed weight are obtained by:

$$\tilde{w}(i, j) = \exp \left(\frac{\sum_{j'=j-N}^{j+N} \sum_{i'=i-M}^{i+M} \log(w(i', j'))}{(2N+1)(2M+1)} \right) \quad (5.9)$$

where, M and N respectively indicate smoothing along the time and frequency axes. The averaging in (5.9) is performed in the logarithmic domain (equivalent to geometric averaging) since the dynamic range of $w(i, j)$ is very large. (If we had performed a normal arithmetic averaging instead of geometric averaging in (5.9), the resulting averages would be dominated inappropriately by the values of $w(i, j)$ of greatest magnitude.)

Results of speech recognition experiments using different values of M and N are reported in Fig. 5.6. The experimental configuration is the same as was used for the data shown in Fig. 5.4. We note that the smoothing operation is quite helpful, and that with suitable smoothing the SBP-D algorithm works as well as the SPB-R. In our subsequent experiments, we used values of $N = 1$ and $M = 4$ in the SPB-D algorithm with 40 gammatone channels. The corresponding spectrogram obtained with this smoothing is shown in Fig. 5.7(d), which is similar to that obtained using SPB-R in Fig. 5.7(b).

5.5 *log spectral mean subtraction*

In this Subsection, we discuss log spectral mean subtraction (LSMS) as an optional pre-processing step in the SPB approach and we compare the performance between LSMS computed for each frequency index and LSMS computed for each gammatone channel. LSMS is a standard technique which has been commonly applied for robustness to environmental mismatch, and this technique is mathematically equivalent to the well known cepstral mean normalization (CMN) procedure. Log spectral mean subtraction is commonly performed for

$\log(P(i, j))$ for each channel i as shown below.

$$\tilde{P}(i, j) = \frac{P(i, j)}{\exp(\frac{1}{2L+1} \sum_{j'=j-L}^{j+L} \log(P(i, j')))} \quad (5.10)$$

Hence, this normalization is performed between the squared gammatone integration in each band and the nonlinearity. It is also reasonable to apply LSMS for $X(e^{j\omega_k}; j)$ for each frequency index k before performing the gammatone frequency integration. This can be expressed as:

$$\tilde{X}(e^{j\omega_k}; j) = \frac{|X(e^{j\omega_k}; j)|}{\exp(\frac{1}{2L+1} \sum_{j'=j-L}^{j+L} \log(|X(e^{j\omega_k}; j')|))} \quad (5.11)$$

Fig. 5.8 depicts the results of speech recognition experiments using the two different approaches to LSMS (without including SPB). In that figure, the moving average window length indicates the length corresponding to $2L + 1$ in (5.10) and (5.11). We note that the approach in (5.10) provides slightly better performance for white noise, but that the performance difference diminishes as the window length increases. However, the LSMS based on (5.11) shows consistently better performance in the presence of background music, which is consistent across all window lengths. This may be explained due to the rich discrete harmonic components in music, which makes frequency-index-based LSMS more effective. In the next Subsection we examine the performance obtained when LSMS as described by (5.11) is used in combination with SPB.

5.6 Experimental results

In this Subsection we present experimental results using the SPB-R algorithm described in Subsection 5.3 and the SPB-D algorithm described in Section 5.4. We also examine the performance of SPB in combination with LSMS as described in Subsection 5.5. We conducted speech recognition experiments using the CMU `Sphinx 3.8` system with `Sphinxbase 0.4.1`. For training the acoustic model, we used `SphinxTrain 1.0`. For the baseline MFCC feature, we used `sphinx_fe` included in `Sphinxbase 0.4.1`. All experiments in this and previous Subsections were conducted under identical condition, with delta and delta-delta components appended to the original features. For training and testing we used subsets of 1600 utterances and 600 utterances respectively from the DARPA Resource Management (RM1) database.

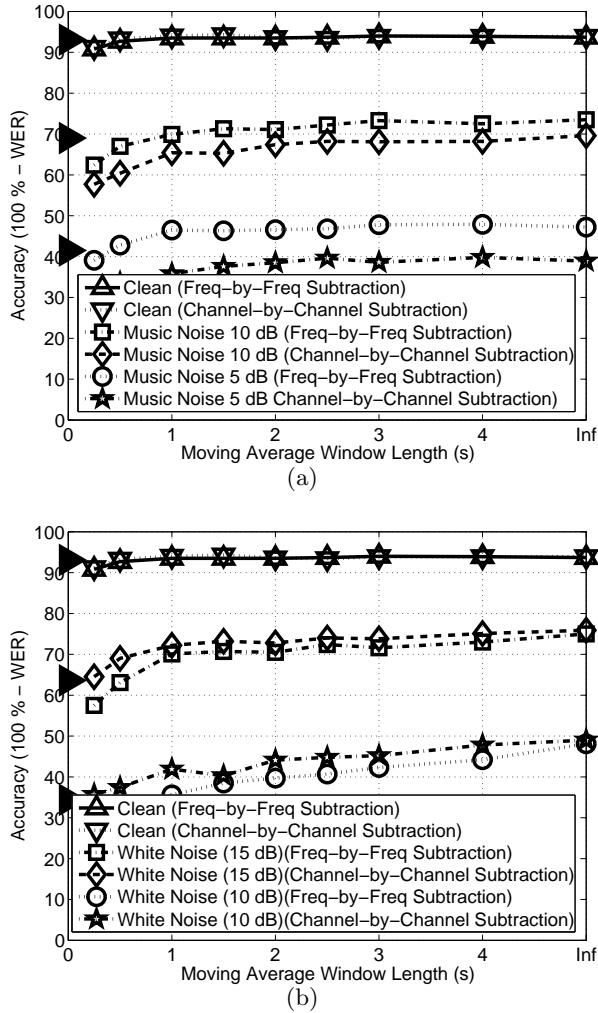


Fig. 5.8: The effect of Log Spectral Subtraction for (a) background music and (b) white noise as a function of the moving window length. The filled triangles at the y-axis represent baseline MFCC performance.

To evaluate the robustness of the feature extraction approaches we digitally added white Gaussian noise and background music noise. The background music was obtained from musical segments of the DARPA HUB 4 database.

In Fig. 5.9, SPB-D is the basic SPB system described in Subsection 5.4. While we noted in a previous paper [43] that gammatone frequency integration provides better performance than conventional triangular frequency integration the effect is minor in these results. Thus, the performance boost of SPB-D over the baseline MFCC is largely due to

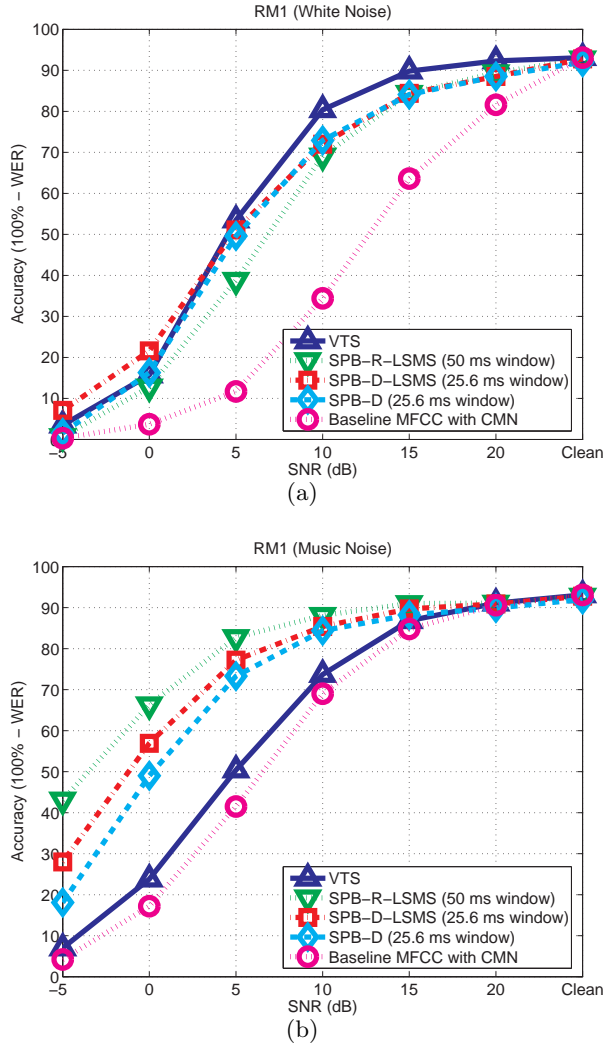


Fig. 5.9: Comparison of recognition accuracy between VTS, SPB-CW and MFCC processing: (a) additive white noise, (b) background music.

the SPB nonlinearity in (5.2) and subsequent gain smoothing. SPB-D-LSMS refers to the combination of the SPB-D and LSMS techniques. For both the SPB-D and SPB-D-LSMS systems we used a window length of 25.6 ms with 10ms between adjacent frames. Even though not explicitly plotted in this figure, SPB-R shows nearly the same performance as SPB-D as mentioned in 5.4 and shown in Fig. 5.4.

We prefer to characterize improvement in recognition accuracy by the amount of lateral threshold shift provided by the processing. For white noise, SPB-D and SPB-D-LSMS provides an improvement of about 7 dB to 8 dB compared to MFCC, as shown in Fig. 5.9.

SPB-R-LSMS results in slightly smaller threshold shift. For comparison, we also conduct experiments using the Vector Taylor Series (VTS) algorithm [8], as shown in Fig. 5.9. For white noise, the performance of SPB family is slightly worse than that obtained using VTS.

Compensation for the effects of music noise, on the other hand, is considered to be much more difficult (*e.g.* [41]). The SPB family of algorithms provides a very impressive improvement in performance with background music. An implementation of SPB-R-LSMS with window durations of 50 ms provides the greatest threshold shift (amounting to about 10 dB), and SPB-D provides a threshold shift of around 7 dB. VTS provides a performance improvement of about 1 dB for the same data.

Open Source MATLAB code for SPB-R and SPB-D can be found at [The code in this directory](#) was used for obtaining the results in this paper.

5.7 Conclusion

In this Subsection, we presented a robust speech recognition algorithm named Small Power Boosting (SPB), which is very helpful for difficult noise environment such as music noise. Our contribution is summarized in the following. First, we examine the PDFs obtained from clean and noisy environments, and observe that small power region is most vulnerable to noise. Based on the observation, we intentionally boost the small power region. We also noted that we should not boost power in each time-frequency bin independently as adjacent time-frequency bins are highly correlated. This can be achieved implicitly in SPB-R and by applying weighting smoothing in SPB-D. We also observed that directly applying nonlinearity results in too small variance for small power regions, which is harmful for robustness and speech sound discrimination. Finally, we also observe that for music noise LSMS for each frequency index is more helpful than doing this for each channel index.

5.8 Proposed Work

In the SPB in Section 5, we explored the effects of smoothing in combination of the non-linearity. We observed that if we apply the nonlinearity for direct feature generation, then the threshold point is helpful in reducing the spectral distance between the clean and the

noisy utterances, however, it has a negative effect of reducing the standard deviation. As a proposed work, we will analyze the effect of variance obtained from the time-frequency bins with small power more rigorously.

6. ENVIRONMENTAL COMPENSATION USING POWER DISTRIBUTION NORMALIZATION

In this chapter, we will discuss several power distribution normalization methods especially based on the power amplitude distributions at each frequency band.

One characteristic of speech signals is that its power level changes rapidly while the background noise power usually changes more slowly. In the case of stationary noise such as white or pink noise, the variation of power approaches zero if the window length is sufficiently large. Even in case of non-stationary noise like music noise, the noise power is not changing as fast as the speech power. Thus, if we measure the variation of the power, then it can be effectively used to see how much the current frame is affected by noise, and furthermore, this information can be used for equalization. One effective way of doing this is measuring the ratio of arithmetic mean to geometric mean, since if power values are not changing very fast, then both arithmetic and geometric mean will have similar values, but if they are changing fast, then arithmetic mean will be much larger than the geometric mean. This ratio is directly related to the shaping parameter of the gamma distribution [56], and it has also been used to estimate the signal-to-noise ratio [56].

In this chapter, we introduce new power distribution normalization algorithms based on this principle. We observe that the the ratio of arithmetic mean to geometric mean ratio of the power within each frequency band differs significantly from clean environments to noisy environment. Thus, by using the ratio obtained from the training DB of clean speech, several different ways of normalization can be considered. As one of such approaches, in Section 6.1, we discuss the Power Bias Subtraction (PBS) approach. In this approach, we subtract the unknown power bias level from the test speech to make the AM-to-GM ratio the same as that of clean training DB.

Another approach called Power-function-based Power Distribution Normalization (PPDN)

is based on application of the power nonlinearity. In this approach, input band power is applied to the power nonlinearity to make the AM-to-GM ratio after the nonlinearity the same as that of clean speech.

6.1 Medium-Duration Power bias subtraction

In this section, we discuss medium-duration power distribution normalization, which provides further decreases in WER. This operation is motivated by the fact that perceptual systems focus on changes in the target signal and largely ignore constant background levels. The algorithm presented in this section resembles conventional spectral subtraction in some ways, but instead of estimating noise power from non-speech segments of an utterance, we simply subtract a bias that is assumed to represent an unknown level of background stimulation.

6.1.1 Medium-duration power bias removal based on arithmetic-to-geometric mean ratios

In Section 3.2, we argued that noise compensation can be accomplished more effectively if we use the temporal analysis methods such as the running average method, and the medium-duration window method. In this subsection, we will introduce Power Bias Subtraction (PBS) using the medium-duration running average method explained in Section 3.2.1.

The first stage of the PBS is frequency analysis. Pre-emphasis of $H(z) = 1 - 0.97z^{-1}$ is performed, and Applying a short-time hamming window with 25.6 ms length is followed. Short-time Fourier Transform (STFT) is performed and the spectrum is squared. The squared spectrum is integrated using the squared gammatone frequency response. Using this procedure, we can obtain the channel-by-channel power $P[m, l]$ where m is the channel index and l is the frame index. In the equation form, it is represented as follows:

$$P[m, l] = \sum_{k=0}^{N-1} \left| X[m, e^{j\omega k}] G_l(e^{j\omega k}) \right|^2 \quad (6.1)$$

, where N is the FFT size. Since we are using 100-ms windows at 16-kHz sampling rate, $N = 2048$. $G_l[k]$ is the l -th channel gammatone filterbank, and $X[m, e^{j\omega k}]$ is the short-time spectrum of the speech signal for this m -th frame. We are using 40 gammatone channels to obtain the channel-by-channel power $P[m, l]$.

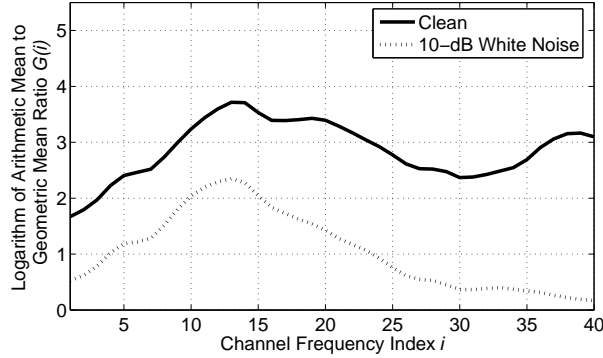


Fig. 6.1: Comparison between $G(l)$ coefficients for clean speech and speech in 10-dB white noise, using $M = 3$ in (7.2).

We estimate the medium-duration power of speech signal $Q[m, l]$ by computing the running average of $P[m, l]$, the power observed in a single analysis frame, according to the equation:

$$Q[m, l] = \frac{1}{2M + 1} \sum_{m'=m-M}^{m+M} P[m', l] \quad (6.2)$$

where l represents the channel index and m is the frame index. As mentioned before, we use a 25.6-ms Hamming window, and 10 ms between successive frames. We found that $M = 3$ is optimal for speech recognition performance, which corresponds to seven consecutive windows or 85.6 ms.

We find it convenient to use the ratio of arithmetic mean to geometric mean (the “AM-to-GM ratio”) to estimate the degree of speech corruption. Because addition is easier to handle than multiplication and exponentiation to the power of $1/M$, we use the logarithm of the ratio of arithmetic and geometric means in the l -th channel as the normalization statistic:

$$G(l) = \log \left[\sum_{m=0}^{M-1} \max(Q[m, l], \epsilon) \right] - \frac{1}{M} \sum_{m=0}^{M-1} \log [\max(Q[m, l], \epsilon)] \quad (6.3)$$

where the small positive constant ϵ is imposed to avoid evaluations of negative infinity. Fig. 7.3 illustrates typical values of the statistic $G(l)$ for clean speech and speech that is corrupted by additive white noise at an SNR of 10 dB. As can be seen, values of $G(l)$ tend to increase

with increasing SNR. $G(l)$ was estimated from 1,600 utterances of the DARPA resource management training set, with $M = 3$ as in (7.2).

6.1.2 Removing the power bias

Power bias removal consists of estimating $B[l]$, the unknown level of background excitation in each channel, and then computing the system output that would be obtained after it is removed. If we could assume a value for $B[l]$, the normalized power $\tilde{Q}[m, l|B(l)]$ is given by following equation:

$$\tilde{Q}[m, l|B(l)] = \max(Q[m, l] - B(l), d_0 Q[m, l]) \quad (6.4)$$

In the above equation d_0 is a small constant (currently 10^{-3}) that prevents $\tilde{Q}[m, l]$ from becoming negative. Using this normalized power $\tilde{Q}[m, l|B(l)]$, we can define the parameter $\tilde{G}(l|B(l))$ from (7.3) and (7.4):

$$\tilde{G}(l|B(l)) = \log \left[\sum_{m=0}^{M-1} \max \left(\tilde{Q}[m, l|B(l)], c_f(l) \right) \right] \quad (6.5)$$

$$- \frac{1}{M} \sum_{m=0}^{M-1} \log \left[\max \left(\tilde{Q}[m, l|B(l)], c_f(l) \right) \right] \quad (6.6)$$

The floor coefficient $c_f(l)$ is defined by:

$$c_f(l) = d_1 \left(\frac{1}{M} \sum_{m'=0}^{M-1} Q[m, l'] \right) \quad (6.7)$$

In our system, we use d_1 of 10^{-3} , causing d_1 to represent -30 dB of the channel average power. In our experiments, we observed that $c_f(l)$ plays a significant role in making the power bias estimate reliable, so its use is highly recommended. We noted previously that the $G(l)$ statistic is smaller for corrupt speech than it is for clean speech. From this observation, we can define the estimated power bias $B^*(l)$ as the smallest power which makes the AM-to-GM ratio the same as that of clean speech. This can be represented by the equation

$$B^*(l) = \min \left\{ B(l) \left| \tilde{G}(l|B(l)) \geq G_{cl}(l) \right. \right\} \quad (6.8)$$

where $G_{cl}(l)$ is the value of $G(l)$ observed for clean speech, as shown in Fig. 7.3 Hence we obtain $B^*(l)$ by increasing $B(l)$ in steps from -50 dB relative to the average power in

Channel l until $\tilde{G}(l|B(l))$ becomes greater than $G_{cl}(l)$ as in Eq. (7.7). Using this procedure for each channel, we can obtain $\tilde{Q}(m, l|B^*(l))$. Thus, for each time-frequency bin represented by $[m, l]$, the power normalization gain is given by:

$$w[m, l] = \frac{\tilde{Q}[m, l|B^*(l)]}{Q[m, l]} \quad (6.9)$$

For smoothing purposes, we average across channels from the $(l - N)$ -th channel up to the $(l + N)$ -th channel. Thus, the final power $\tilde{P}[m, l]$ is given by the following equation,

$$\tilde{P}[m, l] = \left(\frac{1}{2N + 1} \sum_{l'=\max(l-N, 1)}^{\min(l+N, C)} w[l', m] \right) P[m, l] \quad (6.10)$$

where C is total number of channels. In our algorithm, we use $N = 5$ and a total number of 40 gammatone channels. This normalized power $\tilde{P}[m, l]$ is applied to the power function nonlinearity as shown in the block diagram of Fig. 7.1.

6.1.3 Simulation results with Power Normalized Cepstral Coefficient

As of now, we haven't tested the performance of PBS as a separate system. Thus in this subsection, we present experimental results when it is used as a part of Power Normalized Cepstral Coefficient (PNCC). PNCC system will be explained in detail in Chapter 7 and the experimental results are presented in that chapter.

6.2 Bias estimation based on Maximizing the sharpness of the power distribution and power flooring

In this section we describe a power-bias subtraction that is based on maximization of the sharpness of the power distributions. This approach is different from the approach described in the previous section. First, instead of matching the sharpness of the distribution of power coefficients to a training database, we simply maximize this sharpness distribution. We continue to use the ratio of the arithmetic mean to the geometric mean of the power coefficients, which we refer to as the "AM-to-GM ratio", as this measure has proved to be a useful and easily-computed way to characterize the data. (*e.g.* [56]). Second, we apply a minimum

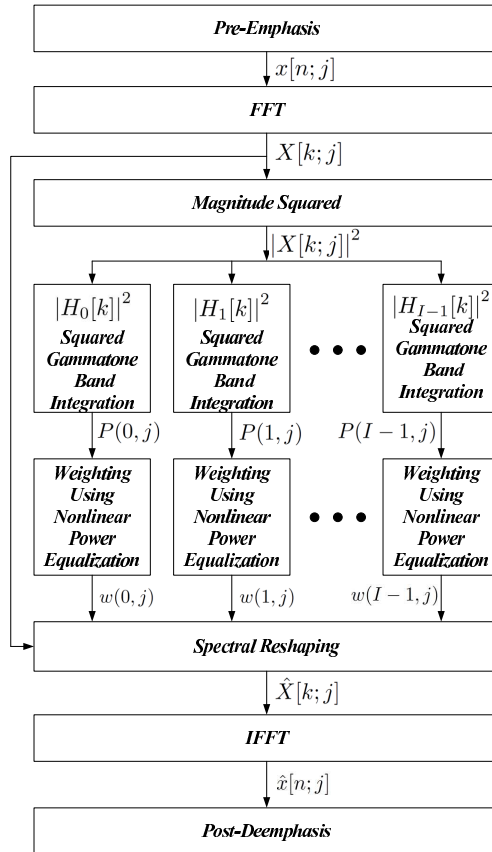


Fig. 6.2: The block diagram of the power function-based power equalization system

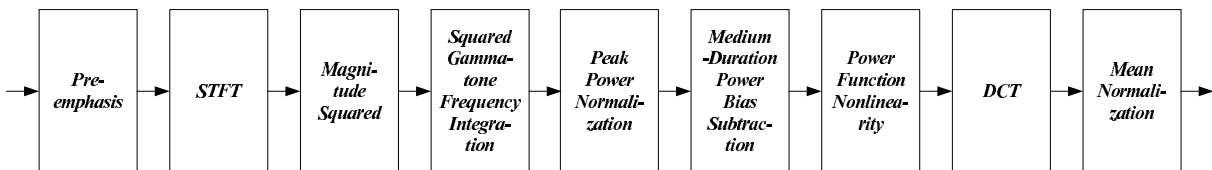


Fig. 6.3: The structure of PNCC feature extraction

threshold to these power values (which we call “power flooring,” because the spectrotemporal segments representing speech that exhibit the smallest power are also the most vulnerable to additive noise (*e.g.* [35])). Using power flooring, we can reduce spectral distortion between training and test sets for these regions. In this section, we will present experimental results when it is applied to the PNCC. The PNCC structure will be described in much more detail in Chapter 7.

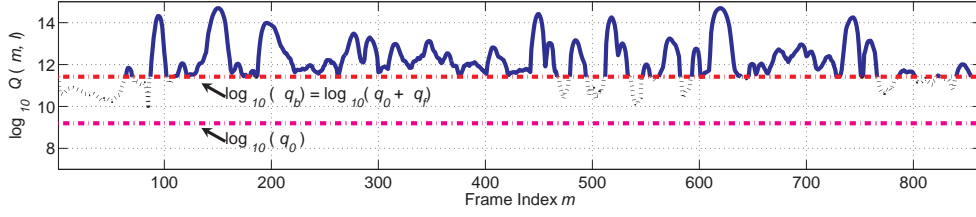


Fig. 6.4: Medium duration power $q[m, l]$ obtained from the 10th channel of a speech utterance corrupted by 10-dB additive background music. The bias power level (q_b) and subtraction power level (q_0) are represented as horizontal lines. Those power levels are the actual calculated levels calculated using the PBS algorithm. The logarithm of the AM-to-GM ratio is calculated only from the portions of the line that are solid.

6.2.1 Power bias subtraction

Notational conventions. We begin by defining some of the mathematical conventions used in the discussion below. Note that all operations are performed on a channel-by-channel basis.

Consider a set $\mathcal{Q}(l)$ as follows:

$$\mathcal{Q}(l) = \left\{ Q[m', l'] : 1 \leq m' \leq M, l' = l \right\} \quad (6.11)$$

where $Q[m, l]$ is the medium-duration power given by (7.2). We define the truncated set \mathcal{Q}_t with respect to the threshold t (which is a subset of $\mathcal{Q}(l)$ above) as follows:

$$\mathcal{Q}_t(l) = \left\{ Q[m, l] : Q[m, l] > t, 1 \leq m \leq M, l' = l \right\} \quad (6.12)$$

We use the symbol μ to represent the mean of $\mathcal{Q}(l)$:

$$\mu(\mathcal{Q}(l)) = \frac{1}{M} \sum_{m'=1}^M Q[m', l] \quad (6.13)$$

We define the max operation between a set and a constant c in the following way:

$$\max\{\mathcal{Q}(l), c\} = \left\{ \max\{q, c\} : q \in \mathcal{Q}(l) \right\} \quad (6.14)$$

Finally, the symbol ξ represents the logarithm of the AM-to-GM ratio for a set $\mathcal{Q}(l)$:

$$\xi(\mathcal{Q}(l)) = \log \left(\frac{1}{M} \sum_{m'=1}^M Q[m', l] \right) - \frac{1}{M} \sum_{m'=1}^M (\log Q[m', l]) \quad (6.15)$$

Implementation of PBS. The objective of PBS is to apply a bias to the power in each of the frequency channels that maximizes the sharpness of the power distribution. This procedure is motivated by the fact that the human auditory system is more sensitive to changes in power over frequency and time than to relatively constant background excitation.

The motivation of power flooring is twofold. First, we wish to limit the extent to which power values of small magnitude affect Eq. (6.15), specifically to avoid values of $\mathcal{Q}(l)$ that are close to zero which cause the log value to approach negative infinity. Second, as mentioned in our previous work (*e.g.* [53, 35]), because small power regions are the most vulnerable to additive noise, we can reduce the spectral distortion caused by additive noise by applying power flooring both to the training and to test data [35].

Let us consider the set $\mathcal{Q}(l)$ in (6.11). If we subtract q_0 from each element, we obtain the following set:

$$\mathcal{R}(l|q_0) = \left\{ R[m', l'] : \begin{array}{l} R[m', l'] = Q[m', l'] - q_0, \\ 1 \leq m' \leq M, l' = l \end{array} \right\} \quad (6.16)$$

Elements in $\mathcal{R}(l|q_0)$ that are larger than the threshold q_f are used in estimating the bias level; values smaller than q_f are replaced by q_f .

In selecting q_f we first obtain the following threshold:

$$q_t = c_0 \mu(\mathcal{R}_{(0)}(l|q_0)) \quad (6.17)$$

where c_0 is a small coefficient called the “power flooring coefficient”, and $\mathcal{R}_{(0)}(l|q_0)$ is the truncated set using the notation defined in (6.12) with the threshold of $t = 0$. For convenience this truncated set is shown below:

$$\mathcal{R}_{(0)}(l|q_0) = \left\{ R[m', l'] : R[m', l'] > 0, 1 \leq m' \leq M, l' = l \right\} \quad (6.18)$$

To prevent a long silence or a long period of constant power from affecting the mean value, we use the following threshold instead of q_t :

$$q_f = c_0 \mu(\mathcal{R}_{(q_t)}(l|q_0)) \quad (6.19)$$

Again, $\mathcal{R}_{(q_t)}(l|q_0)$ is the truncated set obtained from $\mathcal{R}(l|q_0)$ using a threshold of $t = q_t$ (using the definition of the truncated set in (6.12)). Next, the AM-to-GM ratio is calculated

using the above power floor level q_f . Even though q_t and q_f are actually different for each channel l , we drop the channel index for those variables for notational simplicity.

$$g(q_0) = \xi \left(\max \{ \mathcal{R}_{(q_t)}(l|q_0), q_f \} \right) \quad (6.20)$$

The statistic $g(q_0)$ in the above equation represents the logarithm of the AM-to-GM ratio of power values whose values are above q_t after being subtracted by q_0 ; and these values are floored to q_f . The value of q_0 is selected which maximizes Eq. (6.20):

$$\hat{q}_0 = \arg \max_{q_0} \left\{ \xi \left(\max \{ \mathcal{R}_{(q_t)}(l|q_0), q_f \} \right) \right\} \quad (6.21)$$

In searching for q_0 using (6.21), we used the following range:

$$\left\{ q_0 : q_0 = 0 \text{ or } \frac{p_0}{10^{-n/10} + 1}, -70 \leq n \leq 10, n \in \mathcal{Z} \right\} \quad (6.22)$$

where p_0 is the peak power value after peak power normalization. After estimating q_0 , the normalized power $\tilde{Q}[m, l]$ is given by:

$$\tilde{Q}[m, l] = \max \{ Q[m, l] - q_0, q_f \} \quad (6.23)$$

As noted above, q_f provides power flooring. Fig. 6.5 demonstrates that the power flooring coefficient c_0 has a significant effect on recognition accuracy. Based on these results we use a value of 0.01 for c_0 to maintain good recognition accuracy both in clean and noisy environments.

Recall that the weighting factor for a specific time-frequency bin is given by the ratio $\tilde{Q}[m, l]/Q[m, l]$. Since smoothing across channels is known to be helpful (*e.g.* [35], [43]) the weight for channel l is smoothed by computing the average from the $(l - N)^{th}$ channel up to the $(l + N)^{th}$ channel. Hence, the final power $\tilde{P}[m, l]$ is given by:

$$\tilde{P}[m, l] = \left(\frac{1}{l_2 - l_1 + 1} \sum_{l'=l_1}^{l_2} \frac{\tilde{Q}[m, l']}{Q[m, l']} \right) P[m, l] \quad (6.24)$$

where $l_1 = \min(l - N, L)$ and $l_2 = \max(l + N, 1)$, and L is the total number of channels. Fig. 6.6 shows how recognition accuracy depends on the value of the smoothing parameter N . From this figure we can see that performance is best for $N = 3$ or $N = 4$. In the present implementation of PNCC we use $N = 4$ and a total number of $L = 40$ gammatone channels.

6.2.2 Experimental results and conclusions

The implementation of PNCC described in this paper was evaluated by comparing the recognition accuracy obtained with PNCC introduced in this paper with that of conventional MFCC processing implemented as `sphinx_fe` in `sphinxbase 0.4.1`, and with PLP processing using `HCopy` included in `HTK 3.4`. In all cases decoding was performed using the CMU `Sphinx 3.8` system, and training was performed using `SphinxTrain 1.0`. A bigram language model was used in all experiments. For experiments using the DARPA Resource Management (RM1) database we used subsets of 1600 utterances for training and 600 utterances for testing. In other experiments we used WSJ0 SI-84 training set and WSJ0 5k test set. To evaluate the robustness of the feature extraction approaches we digitally added three different types of noise: white noise, street noise, and background music. The background music was obtained from a musical segment of the DARPA Hub 4 Broadcast News database, while the street noise was recorded by us on a busy street. We prefer to characterize improvement in recognition accuracy by the amount of lateral threshold shift provided by the processing. For white noise, PNCC provides an improvement of about 13 dB compared to MFCC, as shown in Fig. 6.7. For street noise and background music, PNCC provides improvements in effective SNR of about 9.5 dB and 5.5 dB, respectively. In the WSJ0 experiment, PNCC improves the effective SNRs by about 10 dB, 8 dB, and 2.5 dB for the three types of noise. These improvements are greater than improvements obtained with algorithms such as Vector Taylor Series (VTS) [8] and significantly better than the standard PLP implementation, as shown in Fig. 6.7. For clean environments, all four approaches (MFCC, PLP, VTS, PNCC) provided similar performance, but PNCC provided the best performance for both the RM1 and WSJ0 5k test set. The results described in this paper are also somewhat better than the previous results described in [53], which were obtained under exactly the same conditions. Improvements compared to the original implementation of PNCC were greatest at lowest SNRs and with background music. The improved PNCC algorithm is conceptually and computationally simpler, and it provides better recognition accuracy.

Open Source MATLAB code for PNCC can be found at http://www.cs.cmu.edu/~robust/archive/algorithms/PNCC_ICASSP2010. The code in this directory was used for

obtaining the results in this paper.

6.3 Power-function-based power distribution normalization algorithm

6.3.1 Structure of the system

FFig. 6.2 shows the entire structure of our power distribution normalization algorithm. The first step is doing pre-emphasis on the input speech signal. Next, medium duration (100 ms) signal is obtained by applying the hamming window. In our system, we are using 10 ms frame period and 100 ms window length. The reason for using rather longer window (medium duration window) will be explained later. After doing this, FFT and gammatone integration are done to obtain the band power $P[m, l]$ which is shown below:

$$P[m, l] = \sum_{k=0}^{N-1} |X[m; j\omega k] H_l(e^{j\omega k})|^2 \quad (6.25)$$

where l and m represent the channel and frame indices respectively, k is the discrete frequency index, and N is the FFT size. Since we are using 100 ms window, for 16 kHz-sampled audio samples, N is 2048. $H_l[k]$ is the spectrum of the gammatone filter bank for the l -th channel and $X[m; k]$ is the short-time spectrum of the speech signal for this m -th frame. We are using 40 gammatone channels for obtaining the bandpower. After power equalization which will be explained in the following subsections, we do spectral reshaping and do the IFFT using OLA to get enhanced speech.

6.3.2 Arithmetic mean to geometric mean ratio of powers in each channel and its normalization

In this subsection, we will examine how the arithmetic mean to geometric mean ratio looks like in each channel. The ratio of the arithmetic mean to the geometric mean of $P[m, l]$ for each channel is given by the following equation:

$$g(l) = \frac{\frac{1}{M} \sum_{m=0}^{M-1} P[m, l]}{\left(\prod_{m=0}^{M-1} P[m, l] \right)^{\frac{1}{M}}} \quad (6.26)$$

Since, addition is easier to handle than multiplication and power to $1/M$, we will use the following logarithm of the above ratio in the following discussion.

$$G(l) = \log \left(\sum_{m=0}^{M-1} P[m, l] \right) - \frac{1}{M} \sum_{m=0}^{M-1} \log P[m, l] \quad (6.27)$$

Fig. 6.8 illustrates $G(l)$ for clean and noisy speech corrupted by 10-dB additive white noise. Thus, we can see that in noisy condition, the values are very different. From now on, let's represent $G(l)$ obtained from clean training database as $G_{cl}(l)$. Now, we will see how we can normalize this difference using a power function.

$$\tilde{P}_{cl}[m, l] = k_l P[m, l]^{a_l} \quad (6.28)$$

In the above equation, $P[m, l]$ is the corrupt medium duration power, and $\tilde{P}_{cl}[m, l]$ is normalized medium duration power. We want the AM-to-GM ratio from normalized power to have the same value from the clean database. Now, our objective is estimating both k_l and a_l under this criterion.

Putting $\tilde{P}_{cl}[m, l]$ in (7.3) and canceling out k_l , the ratio $\tilde{G}_{cl}(l|a_l)$ from this transformed variable $\tilde{P}_{cl}[m, l]$ can be represented by the following equation:

$$\begin{aligned} \tilde{G}_{cl}(l|a_l) = \log \left(\frac{1}{M} \sum_{m=0}^{M-1} P[m, l]^{a_l} \right) \\ - \frac{1}{M} \sum_{m=0}^{M-1} \log P[m, l]^{a_l} \end{aligned} \quad (6.29)$$

For a specific channel l , we see that a_l is the only unknown variable in $\tilde{G}_{cl}(l|a_l)$. Now, from the following equation:

$$\tilde{G}_{cl}(l|a_l) = G_{cl}(l) \quad (6.30)$$

we can obtain a_l value. To obtain the solution, we can use the Newton-Raphson method. Now, we need to obtain k_l in (6.28). By assuming that the derivative of $\tilde{P}_{cl}[m, l]$ with respect to $P[m, l]$ is the unity at $\max_m P[m, l]$ for this channel l , we can set up the following constraint.

$$\left. \frac{d\tilde{P}_{cl}[m, l]}{dP[m, l]} \right|_{\max_m P[m, l]} = 1 \quad (6.31)$$

The above constraint is illustrated in Fig 6.9. The meaning of the above equation is that the slope of the nonlinearity is the unity for the largest power of the l -th channel. This constraint might look arbitrary, but it makes sense for additive noise case, since the following equation will hold:

$$P[m, l] = P_{cl}[m, l] + N[m, l] \quad (6.32)$$

where $P_{cl}[m, l]$ is the true clean speech power, and $N[m, l]$ is the noise power. By differentiating the above equation with respect to $P[m, l]$, we obtain:

$$\frac{dP_{cl}[m, l]}{dP[m, l]} = 1 - \frac{dN[m, l]}{dP[m, l]} \quad (6.33)$$

For the peak $P[m, l]$ value, for a variation of $P[m, l]$, the variation of $N[m, l]$ will be much small, which means variation of $P[m, l]$ around its largest value would be mainly due to variation of speech power not due to the noise power. Thus, the second term on the right hand side in (6.33) would be very small, thus it yields (6.31). By arranging (6.31) with (6.28), we can obtain k_l value, as follows:

$$k_l = \frac{1}{a_l} \max_m P[m, l]^{1-a_l} \quad (6.34)$$

Using the above equation with (6.28), we see that the weight for $P(l, m)$ is given by:

$$\begin{aligned} w[m, l] &= \frac{\tilde{P}_{cl}[m, l]}{P[m, l]} \\ &= \frac{1}{a_l} \left(\frac{P[m, l]}{\max_m P[m, l]} \right)^{a_l-1} \end{aligned} \quad (6.35)$$

After obtaining the weight $w[m, l]$ for each gammatone channel, we reshape the original spectrum $X[m; e^{j\omega k}]$ using the following equation for the m -th frame:

$$\hat{X}[m; e^{j\omega k}] = \sqrt{\sum_{l=0}^{L-1} (w[m, l] |H_l(e^{j\omega k})|)^2} X[m; e^{j\omega k}] \quad (6.36)$$

As mentioned before, $H_l(e^{j\omega k})$ is the spectrum of the l -th channel of the gammatone filter bank. $\hat{X}[m; e^{j\omega k}]$ is the resultant enhanced spectrum. After doing this, we do the IFFT of $\hat{X}[m; e^{j\omega k}]$ to retrieve the time-domain signal and do the post-deemphasis to compensate the effect of the previous pre-emphasis. Speech waveform is resynthesized using the OLA.

6.3.3 Medium duration window

As mentioned in Chapter 3, even though short time windows of 20 ~ 30 ms duration are suitable for feature extraction for speech signals, in many applications, we observe that windows longer than this are better for normalization purpose [53] [43]. The reason is because noise power is changing more slowly than the fast-varying speech signal. Thus, to model the speech part, we need to use short windows, but if we want to measure the noise power and compensate it, then longer windows might be better. Fig. illustrates the accuracy as a function of window length. As can be seen in this figure, if we use the normal window length of 25 ms, then it's doing significantly poorer than longer window. Based on this figure, we see that a window of length between 75 ms and 100 ms is optimal for performance. We will call a window of this duration "medium duration window".

6.3.4 On-line implementation

In many applications, we want an on-line algorithm for speech recognition and speech enhancement. In this case, we cannot use (6.29) for obtaining the coefficient a_l , since this equation requires the knowledge about the entire speech signal. Thus, in this section we will discuss how on-line algorithm version of the power equalization algorithm can be implemented. To resolve this problem, we define two terms $S_1[m, l]$ and $S_2[m, l]$ with the forgetting factor λ of 0.9 as follows.

$$S_1[m, l|a_l] = \lambda S_1[m-1, l] + (1-\lambda)Q_l(m)^{a_l} \quad (6.37)$$

$$S_2[m, l|a_l] = \lambda S_2[m-1, l] + (1-\lambda)\ln Q_l(m)^{a_l} \quad (6.38)$$

$$a_l = 1, 2, \dots, 10$$

In our on-line algorithm, we calculate $S_1(m, l|a_l)$ and $S_2(m, l|a_l)$ for integers value of a_l in $1 \leq a_l \leq 10$ for each frame. From (6.29), we can define the on-line version of $G(l)$ using $S_1[m, l]$ and $S_2[m, l]$.

$$\tilde{G}_{cl}(m, l|a_l) = \log(S_1(m, l|a_l)) - S_2(m, l|a_l), a_l = 1, 2, \dots, 10 \quad (6.39)$$

Now, the $\hat{a}[m, l]$ is defined as the solution to the following equation:

$$\tilde{G}_{cl}(m, l|\hat{a}[m, l]) = G_{cl}(l) \quad (6.40)$$

Since we are updating $G_{cl}(m, l|a_l)$ for each frame using integer values of a_l in $1 \leq a_l \leq 10$, we use linear interpolation of $\tilde{G}_{cl}(m, l|a_l)$ with respect to a_l to obtain the solution to (6.40). For estimating k_l using (6.34), we need to obtain the peak power. In the on-line version, we define the following on-line peak power $M[m, l]$.

$$M[m, l] = \max(\lambda M[m-1, l], P[m, l]) \quad (6.41)$$

$$Q[m, l] = \lambda Q[m-1, l] + (1 - \lambda)M[m, l] \quad (6.42)$$

Instead of directly using $M[m, l]$, we use the smoothed online peak $Q[m, l]$. Using $Q[m, l]$ and $\hat{a}[m, l]$ with (6.35), we obtain:

$$w[m, l] = \frac{1}{\hat{a}[m, l]} \left(\frac{P[m, l]}{Q[m, l]} \right)^{\hat{a}[m, l]-1} \quad (6.43)$$

Now, using $w[m, l]$ in (6.36), we can normalize spectrum and can do resynthesize speech using IFFT and OLA. In (6.41) and (6.42), we use the same λ of 0.9 as those in (6.37) and (6.38). In our implementation, we use the first 10 frames for estimating the initial values of the $\hat{a}[m, l]$ and $Q[m, l]$, but after doing this initialization, no lookahead buffer is used in processing the speech.

Fig. 6.11 shows spectrograms of original speech corrupted by various types of additive noise, and corresponding spectrograms of processed speech using the on-line PPDN explained in this section. As shown in 6.11(b), for additive Gaussian white noise, improvement is observable even at 0-dB Signal-to-Noise Ratio (SNR) level. For the 10-dB SNR music and 5-dB SNR street noise which are more realistic noise, as shown in 6.11(d) and 6.11(f), we can clearly observe that processing gives us improvements. In the next section, we present speech recognition experimental results using the on-line PPDN.

6.3.5 Simulation results of the on-line power equalization algorithm

In this section, we will see the experimental results obtained on the DARPA Resource Management (RM) database using the on-line version explained in Section 6.3.4. First, perceptually, we could observe that this algorithm has significant effects in enhancing the quality of speech. Thus, this algorithm can be used for speech enhancement. In the RM Database, we used 1,600 utterances for training and 600 utterances for testing. We used SphinxTrain 1.0

for training the acoustic model, and Sphinx 3.8 for decoding. For feature extraction, we used sphinx_fe which is included in sphinxbase 0.4.1. In Fig. 4 (a), we used the test utterances corrupted by the white noise, and in Fig. 4 (b), we used the test utterances corrupted by musical segments of DARPA Hub 4 Broadcast News database. We prefer to characterize improvement as the threshold shift provided by the processing. As shown in these figures, this waveform normalization showed 10 dB threshold shifts for white noise, and 3.5 dB shifts for background music noise. Note that obtaining improvements for background music noise is not so easy. For white noise, as shown in the figure, Power Equalization algorithm showed 10 dB threshold shift. For comparison, we also did experiments using the state of the art noise compensation algorithm VTS (Vector Taylor Series) [8]. For PPE, MVN showed slightly better performance than CMN, but for VTS, we could not observe significant performance improvement using MVN, so we compared MVN version of PPE and CMN version of VTS. If the SNR is equal to or less than 5 dB, PPE algorithm is doing better than VTS and the threshold shift is also larger, but if the SNR is equal to or higher than 10 dB, then VTS is doing somewhat better. In the street noise, both of them showed similar performances. Music noise is considered to be more difficult than white or street noise [12]. For music noise, PPE algorithm showed around 3.5 dB threshold shift, and it is showing better performance than the VTS for all SNR ranges. Matlab version of demo package used for this experiment is available at <http://www.cs.cmu.edu/~chanwook/Algorithms/OnlinePPE/DemoPackage.zip>. This package is used in obtaining the recognition experiments shown in this section.

6.4 Conclusions

In this chapter, we proposed a new power equalization algorithm based on power function and the ratio of arithmetic to geometric mean of band-power. The contribution of our work is as follows. First, we proposed a new algorithm which is very simple and easy to implement compared to other normalization algorithms. At the same time, This algorithm turned out to be quite effective against additive noise and it is showing comparable or somewhat better performance than current state of art techniques like VTS. Second, we developed an efficient algorithm which can re-synthesize enhanced speech. So, unlike compensation algorithm in

feature domain, this algorithm can be effectively used for speech enhancement, and it can also be used as a pre-processing stage with other algorithms working in cepstral domain. Third, this algorithm can be effectively implemented as an online algorithm without any lookahead buffer. This characteristic makes this algorithm quite useful for applications like real-time speech recognition or real-time speech enhancement. Besides the above mentioned things, in our work, we could observe that for normalization, windows longer than those used in feature are better for normalization purpose, so we used 100 ms window length in this normalization scheme.

6.5 *Proposed Work*

In this chapter, we discussed two different ways of power normalization using the AM-to-GM ratio as a statistic. One problem of AM-to-GM ratio is that if we just use the off-line algorithm, then it is not a very good measure about the rate of change if the input utterance is sufficiently long. So, we will handle the problem in two different ways.

First, we can do analysis in the on-line way using exponential way. In PPDN algorithm, this is already done, but we will try to apply this idea to PBS and also we want to find some general ways of finding the optimal forgetting factor.

Secondly, we will evaluate the system by combining this idea with a modulation filter scheme.

As another proposed work, we will evaluate PBS system in isolation. Currently, PBS system has been tested only as a part of Power Normalized Cepstral Coefficient (PNCC).

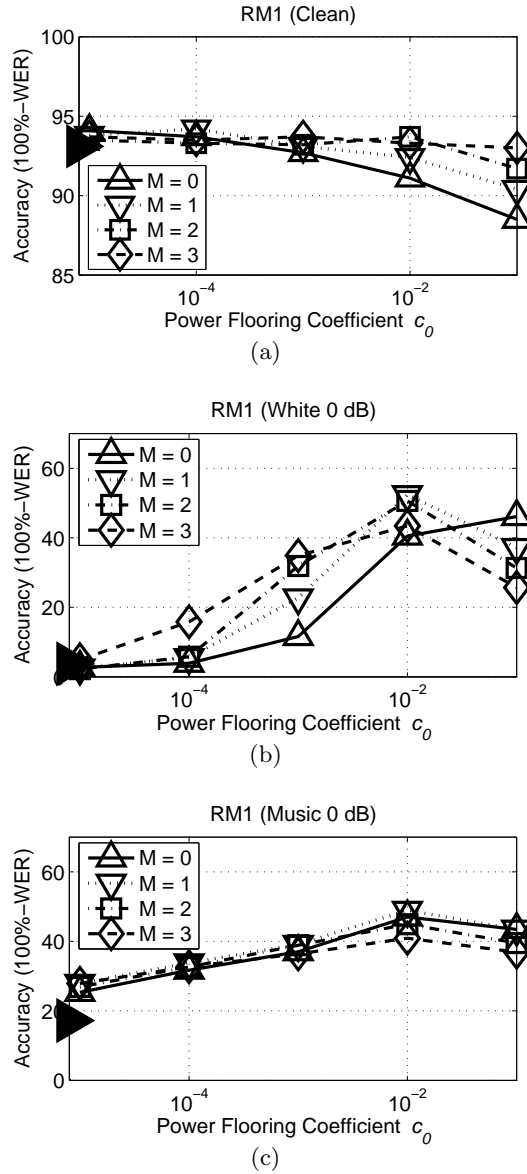
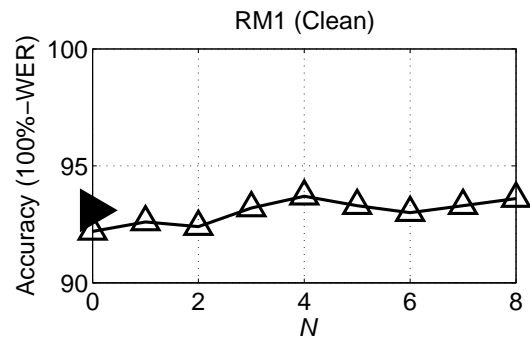
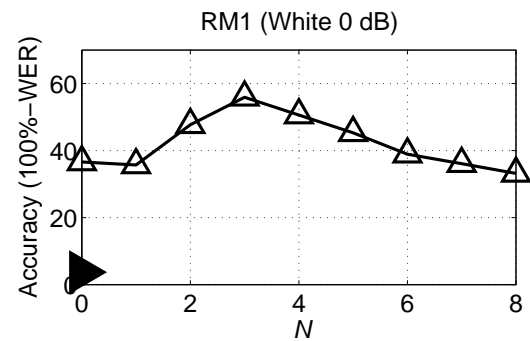


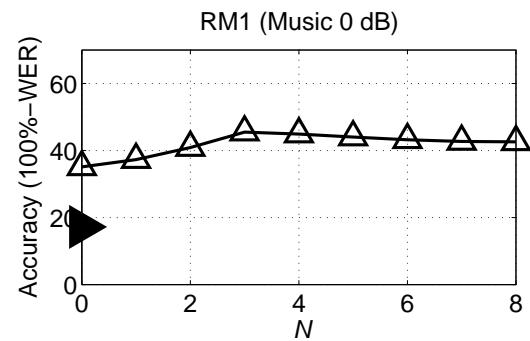
Fig. 6.5: The dependence of speech recognition accuracy obtained using PNCC on the medium-duration window factor M and the power flooring coefficient c_0 . Results were obtained for (a) the clean RM1 test data (b) the RM1 test set corrupted by 0-dB white noise, and (c) the RM1 test set corrupted by 0-dB background music. The filled triangle on the y-axis represents the baseline MFCC result for the same test set.



(a)



(b)



(c)

Fig. 6.6: The corresponding dependence of speech recognition accuracy on the value of the weight smoothing factor N . The filled triangle on the y-axis represents the baseline MFCC result for the same test set. For c_0 and M , we used 0.01 and 2 respectively.

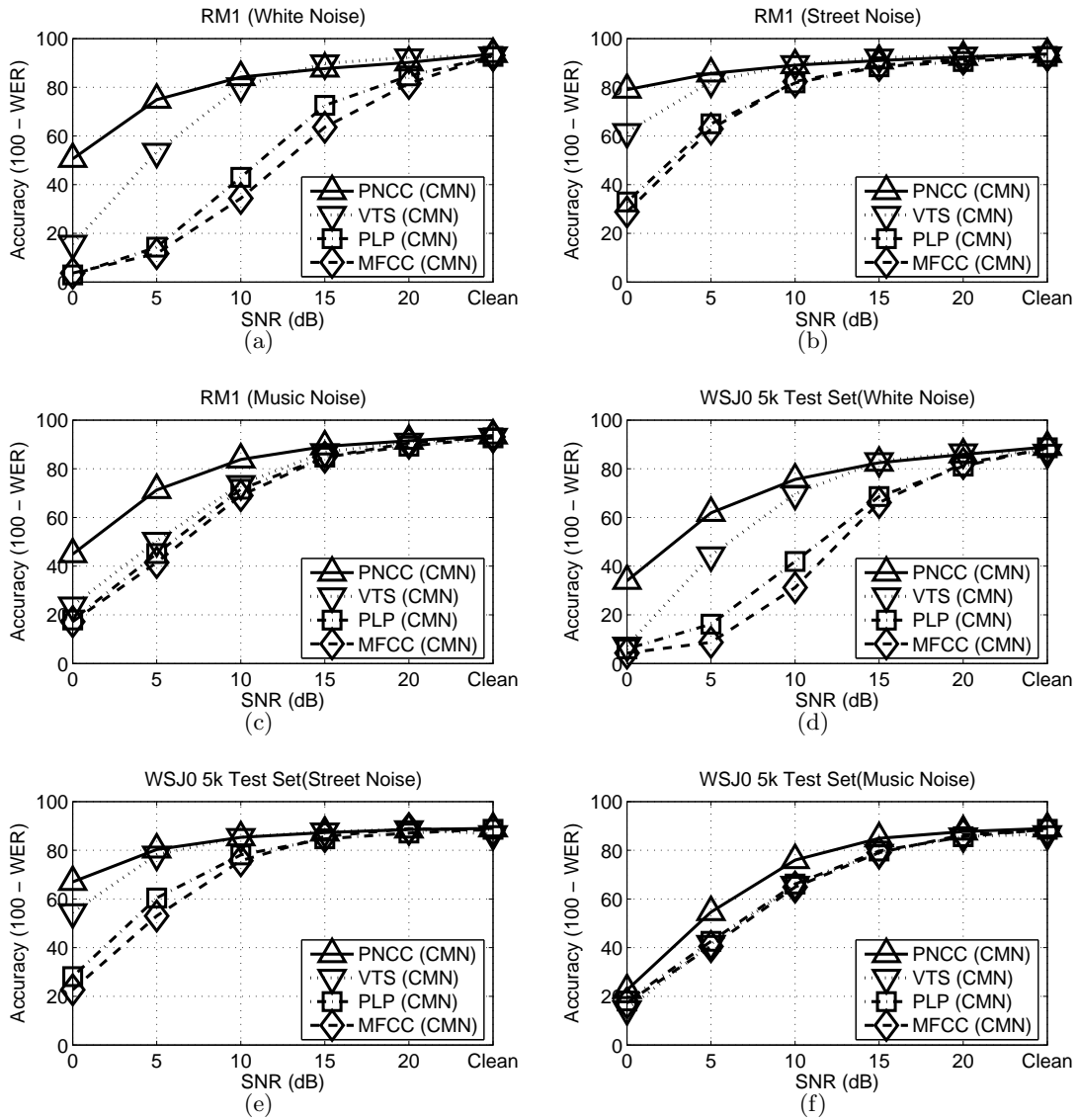


Fig. 6.7: Speech recognition accuracy obtained in different environments for different training and test sets. The RM1 database was used to produce the data in (a), (b), and (c), and the WSJ0 SI-84 training set and WSJ0 5k test set were used for the data of panels (d), (e), and (f).

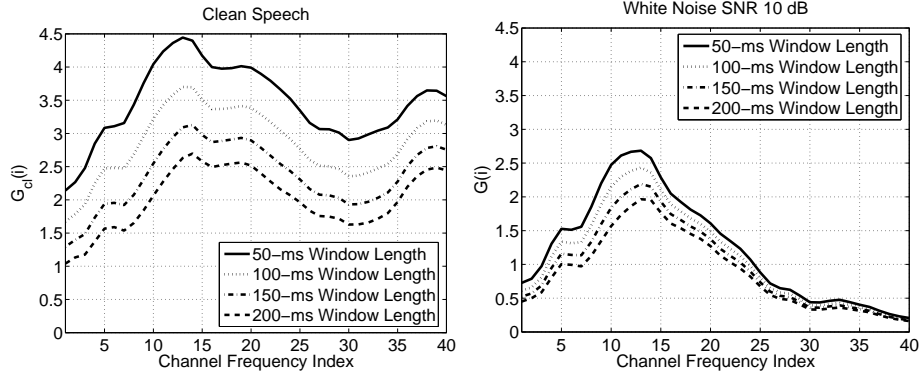


Fig. 6.8: The logarithm of the ratio of arithmetic mean to geometric mean of power from clean (a) and noise speech corrupted by 10 dB white noise (b). Data is collected from 1,600 training utterances of the resource management DB

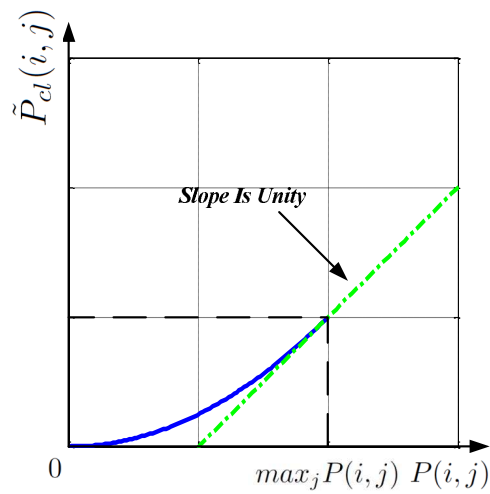


Fig. 6.9: The assumption about the relationship between $P_{cl}[m, l]$ and $P[m, l]$

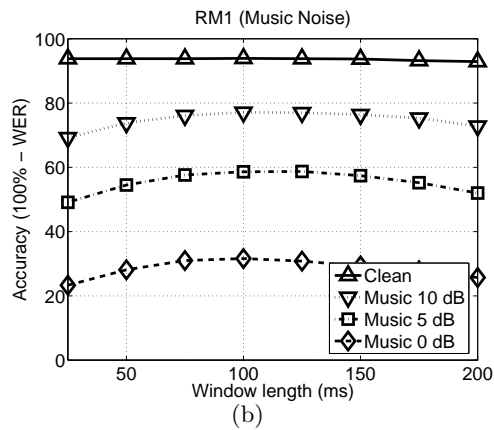
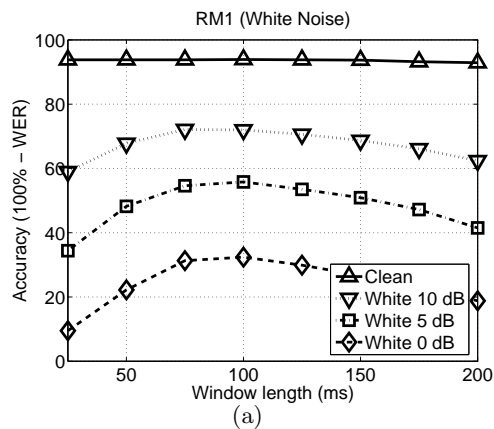


Fig. 6.10: Speech recognition accuracy as a function of the window length for the DARPA RM database corrupted by (a) white noise and (b) background music noise.

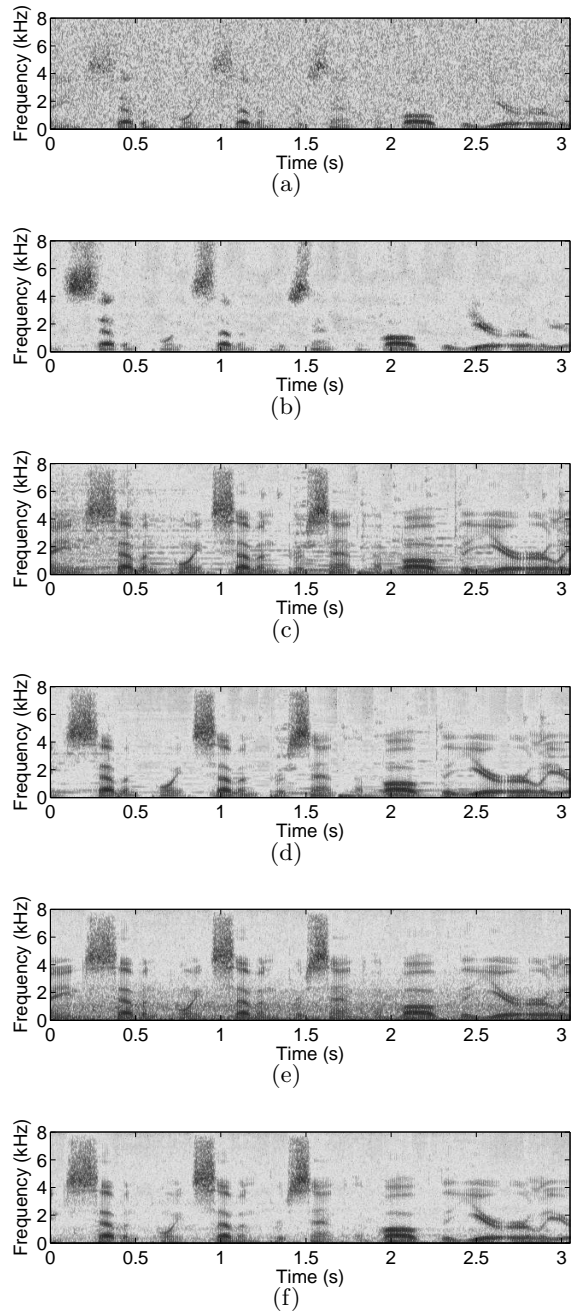


Fig. 6.11: Sample spectrograms illustrating the effects of on-line PPDN processing. (a) original speech corrupted by 0-dB additive white noise, (b) processed speech corrupted by 0-dB additive white noise (c) original speech corrupted by 10-dB additive music noise (d) processed speech corrupted by 10-dB additive music noise (e) original speech corrupted by 5-dB street noise (f) processed speech corrupted by 5-dB street noise

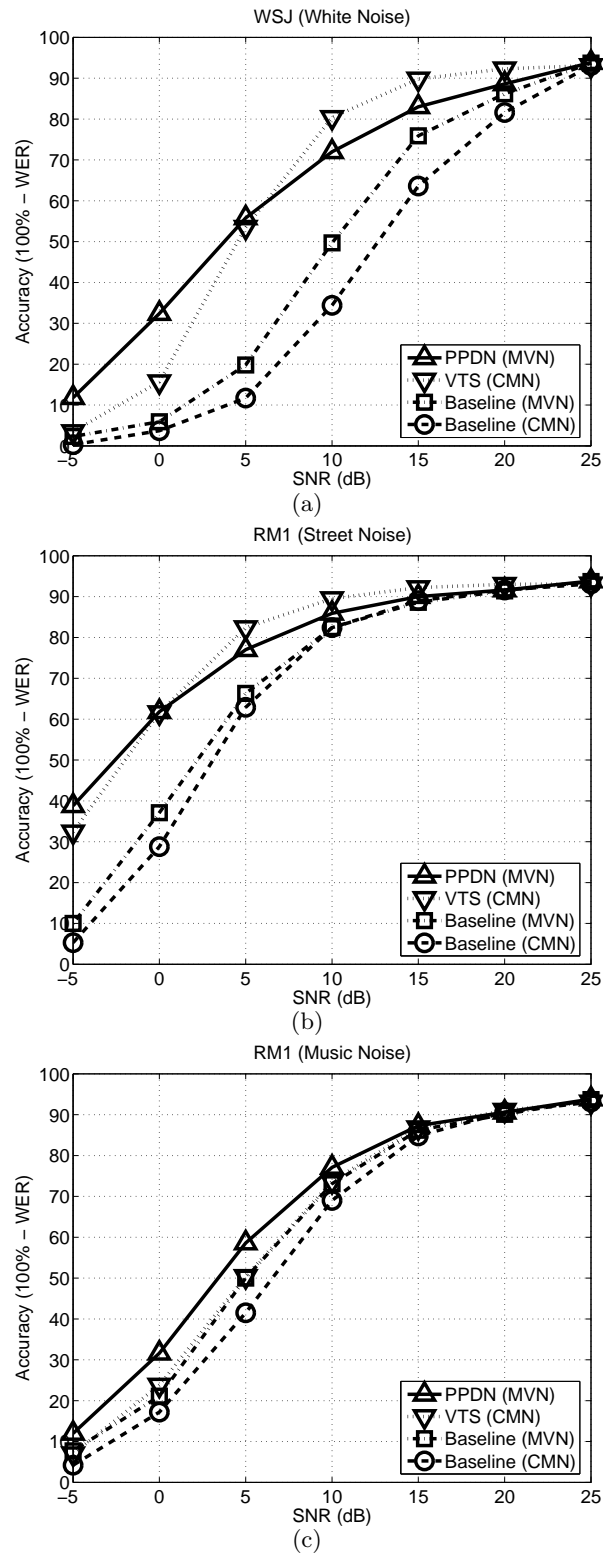


Fig. 6.12: Performance comparison for the DARPA RM database corrupted by (a) white noise, (b) street noise, and (c) music noise.

7. POWER NORMALIZED CEPSTRAL COEFFICIENT

In this chapter, we introduce the (Power Normalized Cepstral Coefficient) PNCC feature extraction algorithm. The structure of PNCC is similar to MFCC, but this feature extraction system is more faithful in representing physiological observations.

Many of the discussion we made in previous chapters has been employed in the PNCC. For example, motivated by the discussion in Chapter 5, we use the power law nonlinearity with a power coefficient between $1/10$ and $1/15$. As mentioned in Chapter 3, we use the gammatone frequency weighting instead of the conventional triangular shape frequency weighting employed in MFCC.

As discussed in Chapter 3, a longer window is better for estimating the noisy component. So, in PNCC, we use a medium-duration window for normalization purpose. We are using the Medium-duration-window Running Average (MRA) approach discussed in that chapter. As will be explained in more detail in this chapter, averaging the weighting coefficient across frequency channel also has a significant impact in speech recognition performance.

In Chapter 2, we reviewed several techniques which try to remove a constant or a slowly varying component of the signal, which are likely to coming from the noisy source. In the PNCC structure, we propose several new techniques of achieving this objective.

Figure 7.1 compares the structure of conventional MFCC processing, PLP processing [49], and the new approach described in this paper, which will be called Power-Normalized Cepstral Coefficients (PNCC). As can be seen from Fig. 7.1, the major innovations in this algorithm are the use of a well-motivated power function that replaces the log function, and the use of a novel approach to the blind removal of background excitation based on medium-duration power estimation. This normalization makes use of the ratio of the arithmetic mean to the geometric mean, which has proved to be a useful measure in determining the extent to which speech is corrupted by noise [56]. In addition, PNCC uses frequency weighting

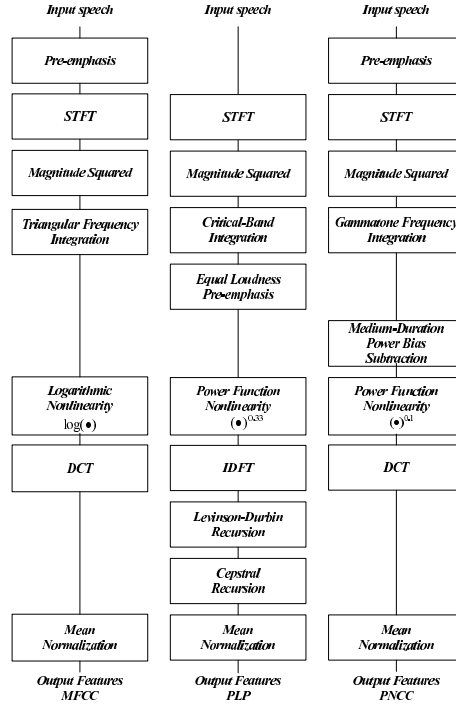


Fig. 7.1: Comparison of the PNCC feature extraction discussed in this paper with MFCC and PLP feature extraction.

based on the gammatone filter shape [55] rather than the triangular frequency weighting or the trapezoidal frequency weighting associated with the MFCC and PLP computation, respectively. A pre-emphasis filter of the form $H(z) = 1 - 0.97z^{-1}$ is applied first. The STFT analysis is performed using Hamming windows of duration 25.6 ms, with 10 ms between frames for a sampling frequency of 16 kHz, 40 gammatone channels. After passing through the gammatone channel, the power is normalized using peak power (*i.e.* the 95th percentile of short-time power).

7.1 Derivation of the power function nonlinearity

Currently the most widely used feature extraction algorithms are Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP). Both the MFCC and PLP

procedures include intrinsic nonlinearities: PLP passes the amplitude-normalized short-time power of critical-band filters through a cube-root nonlinearity to approximate the power law of hearing [49, 50] while the MFCC procedure passes its filter outputs through a logarithmic function. Even though the importance of auditory nonlinearity has been confirmed in several studies (*e.g.* [48]), there has been relatively little analysis concerning the effects of peripheral nonlinearities. In sophisticated auditory models such as [57], the curve relating input level in decibels to the auditory-nerve firing rate is usually S-shaped. For example, the dotted line in the upper panel of Fig. 7.2 shows the relation between the intensity of a tone in dB and the rate of the auditory-nerve response, averaged across frequency, based on predictions by the model of [57] with the spontaneous rate of firing assumed to be 50 spikes/second. This curve is an abstract of results from many studies that observe that the firing rate is almost constant if the input SPL is smaller than a threshold intensity (-10 dB in this case), that the rate increases approximately linearly between 0 and 20 dB, and that it saturates at higher input levels. Because the logarithmic nonlinearity used in MFCC features does not exhibit threshold behavior, for speech segments of low power the output of the logarithm nonlinearity can produce large output changes even if the changes in input are small. This characteristic, which can degrade speech recognition accuracy, becomes very obvious as the input approaches zero. With a power-function nonlinearity, the output is close to zero if the input is very small, which is what is observed in human auditory processing.

The solid curve in the upper panel of Fig. 7.2 is a piecewise-linear approximation to the dotted curve in the same panel for intensities below 0 dB. For greater input intensities this solid curve is a linear approximation to the dynamic behavior of the rate-intensity curve between 0 and 20 dB. Hence, this solid curve exhibits threshold behavior but no saturation. We prefer to model the higher intensities with a curve that continues to increase linearly to avoid spectral distortion caused by the saturation seen in the dotted curve in the upper panel of Fig. 7.2.

The solid curve of the lower panel of Fig. 7.2 reprises the solid curve in the upper panel of the same figure, but translated downward so that for small intensities the output is zero (rather than the physiologically-appropriate spontaneous rate of 50 spikes/s). The dotted power function in that panel is the MMSE-based best-fit power function to the piecewise-linear solid curve. The reason for choosing the power-law nonlinearity instead of the solid

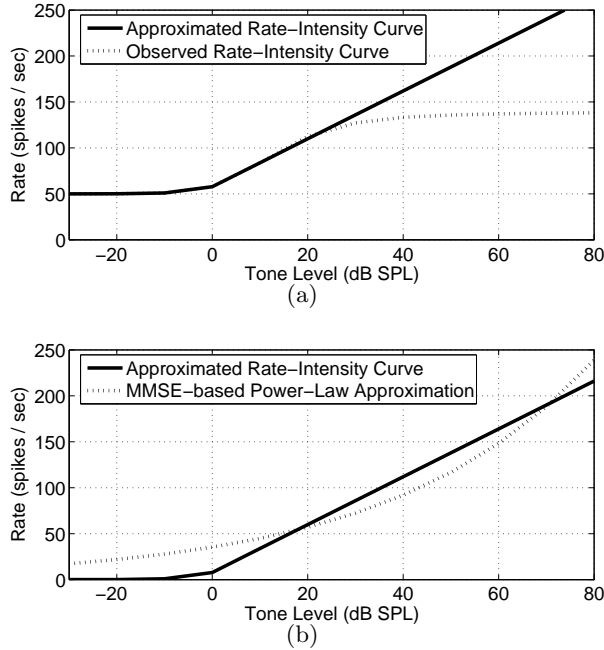


Fig. 7.2: Upper panel: Observed frequency-averaged mean rate of auditory-nerve firings versus intensity (dotted curve) and its piece-wise linear approximation (solid curve). Lower panel: Piece-wise linear rate-level curve with no saturation (solid curve) and best-fit power function approximation (dotted curve).

curve in Fig. 7.2 is that the dynamic behavior of the output does not depend critically on the input amplitude. This nonlinearity, which is what is used in PNCC feature extraction, is described by the equation

$$y = x^{a_0} \tag{7.1}$$

with the best-fit value of the exponent observed to be $a_0 = 0.1$. We note that this exponent differs somewhat from the power-law exponent of 0.33 used for PLP features, which is based on Steven’s power law of hearing [50]. While our power-function nonlinearity may appear to be only a crude approximation to the physiological rate-intensity function, we will show in Sec. 7.3 that it provides substantial improvement in recognition accuracy compared to the traditional log nonlinearity used in MFCC processing.

7.2 Medium-duration power bias removal

In this section, we discuss medium-duration power normalization, which provides further decreases in WER. This operation is motivated by the fact that perceptual systems focus on target signal changes and largely ignore constant background levels. The algorithm presented in this section resembles conventional spectral subtraction in some ways, but instead of estimating noise power from non-speech segments of an utterance, we simply subtract a bias that is assumed to represent an unknown level of background stimulation.

7.2.1 Medium-duration power bias removal based on arithmetic-to-geometric mean ratios

Most speech recognition and speech coding systems use analysis frames of duration between 20 ms and 30 ms. Nevertheless, it is frequently observed that longer analysis windows provide better performance for noise modeling and/or environmental normalization, presumably because noise power changes more slowly than speech power. In PNCC processing we estimate the medium-duration power of speech signal $Q(i, j)$ by computing the running average of $P(i, j)$, the power observed in a single analysis frame, according to the equation:

$$Q(i, j) = \frac{1}{2M+1} \sum_{j'=j-M}^{j+M} P(i, j') \quad (7.2)$$

where i represents the channel index and j is the frame index. As mentioned before, we use a 25.6-ms Hamming window, and 10 ms between successive frames. We found that $M = 3$ is optimal for speech recognition performance, which corresponds to seven consecutive windows or 85.6 ms.

We find it convenient to use the ratio of arithmetic mean to geometric mean (the “AM-to-GM ratio”) to estimate the degree of speech corruption. Because addition is easier to handle than multiplication and exponentiation to the power of $1/J$, we use the logarithm of the ratio of arithmetic and geometric means in the i -th channel as the normalization statistic:

$$G(i) = \log \left[\sum_{j=0}^{J-1} \max(Q(i, j), \epsilon) \right] - \frac{1}{J} \sum_{j=0}^{J-1} \log [\max(Q(i, j), \epsilon)] \quad (7.3)$$

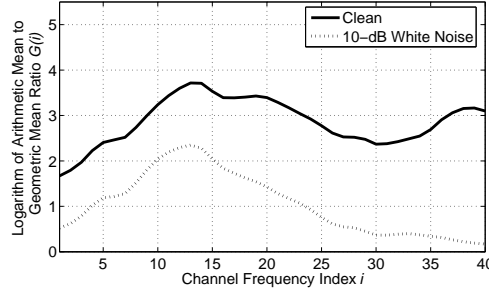


Fig. 7.3: Comparison between $G(i)$ coefficients for clean speech and speech in 10-dB white noise, using $M = 3$ in (7.2).

The ϵ term in the above equation is imposed to avoid evaluations of negative infinity. Fig. 7.3 illustrates typical values of the statistic $G(i)$ for clean speech and speech that is corrupted by additive white noise at an SNR of 10 dB. As can be seen, values of $G(i)$ tend to decrease with SNR. $G(i)$ was estimated from the 1,600 utterances of the DARPA resource management training set, with $M = 3$ as in (7.2).

7.2.2 Removing the power bias

Power bias removal consists of estimating $B(i)$, the unknown level of background excitation in each channel, and then computing the system output that would be obtained after it is removed. If we could assume a value for $B(i)$, the normalized power $\tilde{Q}(i, j|B(i))$ is given by following equation:

$$\tilde{Q}(i, j|B(i)) = \max(Q(i, j) - B(i), d_0 Q(i, j)) \quad (7.4)$$

In the above equation d_0 is a small constant (currently 10^{-3} that prevents $\tilde{Q}(i, j)$ from becoming negative. Using this normalized power $\tilde{Q}(i, j|B(i))$, we can define the parameter $\tilde{G}(i|B(i))$ from (7.3) and (7.4):

$$\begin{aligned} \tilde{G}(i|B(i)) = \log & \left[\sum_{j=0}^{J-1} \max \left(\tilde{Q}(i, j|B(i)), c_f(i) \right) \right] \\ & - \frac{1}{J} \sum_{j=0}^{J-1} \log \left[\max \left(\tilde{Q}(i, j|B(i)), c_f(i) \right) \right] \end{aligned} \quad (7.5)$$

The floor coefficient $c_f(i)$ is defined by:

$$c_f(i) = d_1 \left(\frac{1}{J} \sum_{j'=0}^{J-1} Q(i, j') \right) \quad (7.6)$$

In our system, we use d_1 of 10^{-3} , causing d_1 to represent -30 dB of the channel average power. In our experiments, we observed that $c_f(i)$ plays a significant role in making the power bias estimate reliable, so its use is highly recommended. We noted previously that the $G(i)$ statistic is smaller for corrupt speech than it is for clean speech. From this observation, we can define the estimated power bias $B^*(i)$ as the smallest power which makes the AM-to-GM ratio the same as that of clean speech. This can be represented by the equation

$$B^*(i) = \min \left\{ B(i) \left| \tilde{G}(i|B(i)) \geq G_{cl}(i) \right. \right\} \quad (7.7)$$

where $G_{cl}(i)$ is the value of $G(i)$ observed for clean speech, as shown in Fig. 7.3 Hence we obtain $B^*(i)$ by increasing $B(i)$ in steps from -50 dB relative to the average power in Channel i until $\tilde{G}(i|B(i))$ becomes greater than $G_{cl}(i)$ as in Eq. (7.7). Using this procedure for each channel, we can obtain $\tilde{Q}(i, j|B^*(i))$. Thus, for each time-frequency bin represented by (i, j) , the power normalization gain is given by:

$$w(i, j) = \frac{\tilde{Q}(i, j|B^*(i))}{Q(i, j)} \quad (7.8)$$

For smoothing purposes, we average across channels from the $i - N$ th channel up to the $i + N$ th channel. Thus, the final power $\tilde{P}(i, j)$ is given by the following equation,

$$\tilde{P}(i, j) = \left(\frac{1}{2N + 1} \sum_{i'=\max(i-N,1)}^{\min(i+N,C)} w(i', j) \right) P(i, j) \quad (7.9)$$

where C is total number of channels. In our algorithm, we use $N = 5$ and a total number of 40 gammatone channels. This normalized power $\tilde{P}(i, j)$ is applied to the power function nonlinearity as shown in the block diagram of Fig. 7.1.

7.3 Experimental results and conclusions

The PNCC system described in Secs. 7.1 and 7.2 was evaluated by comparing the recognition accuracy obtained using the CMU Sphinx 3.8 system with Sphinxbase 0.4.1, with

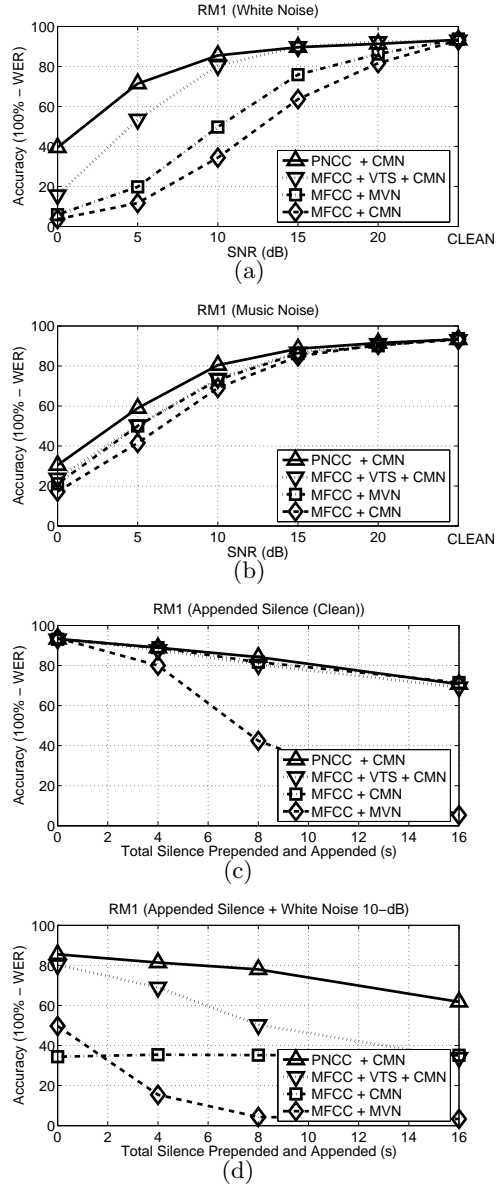


Fig. 7.4: Speech recognition accuracy obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of white Gaussian noise added to the data used in panel (c).

PNCC introduced in this paper, with that of conventional MFCC processing, and with PLP processing as included in HCopy of HTK 3.4. For training and testing, we used subsets of 1600 utterances and 600 utterances respectively from the DARPA Resource Management (RM1) database and trained using SphinxTrain 1.0. To evaluate the robustness of the feature extraction approaches we digitally added three different types of noise: white noise, street noise, and background music. The background music was obtained from a musical segment of the DARPA Hub 4 Broadcast News database, while the street noise was recorded by us on a busy street. We prefer to characterize improvement in recognition accuracy by the amount of lateral threshold shift provided by the processing. For white noise, PNCC provides an improvement of about 12 dB to 13 dB compared to MFCC, as shown in Fig. ???. For the street noise and the music noise, PNCC provides 8 dB and 3.5 dB shifts, respectively. These improvements are greater than improvements obtained with other current state-of-the-art algorithms such as Vector Taylor Series (VTS) [8], as shown in Fig. 7.4. We observe that if silence is added to the beginning and ends of the utterances, performance using some algorithms like mean-variance normalization (MVN) suffers if a good voice activity detector (VAD) is not included, as shown in Fig. 7.4. PNCC, on the other hand, degrades only slightly under the same conditions without VADs.

PNCC requires only slightly more computation than MFCC and much less computation than VTS. We also note that the use of the power nonlinearity and gammatone weighting with the DCT (dels in Fig. ??) still performs significantly better than PLP.

Open Source MATLAB code for PNCC can be found at <http://www.cs.cmu.edu/~robust/archive/algorithms>

The code in this directory was used for obtaining the results in this paper.

8. COMPENSATION WITH 2 MICS

In this chapter, we present a new two-microphone approach that improves speech recognition accuracy when speech is masked by other speech or ambient noise. There have been many attempts to suppress noise signals coming from different directions from the target direction using either Interaural Time Delay (ITD), Interaural Phase Difference (IPD), or Interaural Intensity Difference (IID) (*e.g.* [16] [58]). The algorithm improves on previous systems that have been successful in separating signals based on differences in arrival time of signal components from two microphones. The present algorithm differs from these efforts in that the signal selection takes place in the frequency domain with longer window and smoothing. We observe that smoothing of the phase estimates over time and frequency is needed to support adequate speech recognition performance. We demonstrate that the algorithm described in this paper chapter provides better recognition accuracy than time-domain-based signal separation algorithms, and at less than 10 percent of the computation cost.

8.1 Introduction

Speech recognition systems have significantly improved in the past decades but noise robustness and computational complexity remain critical issues. A number of algorithms have shown improvements for stationary noise (*e.g.* [10, 11]). Nevertheless, improvement in non-stationary noise remains a difficult issue (*e.g.* [12]). In these environments, auditory processing [13] and missing-feature-based approaches [?] are promising. An alternative approach is signal separation based on analysis of differences in arrival time (*e.g.* [15, 16, 17]). It is well documented that the human binaural system bears remarkable ability in speech separation (*e.g.* [17]). Many models have been developed that describe various binaural phenomena (*e.g.* [18, 19]), typically based on interaural time difference (ITD), interaural phase difference (IPD), interaural intensity difference (IID), or changes of interaural correlation.

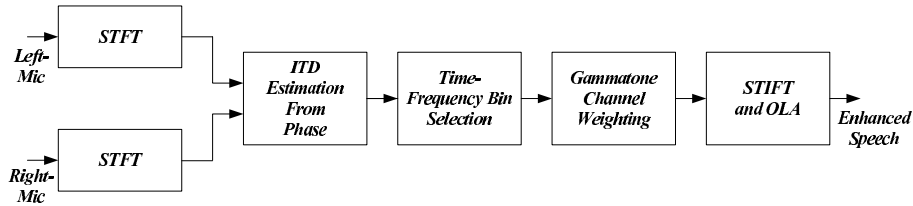


Fig. 8.1: The block diagram of the Phase Difference Channel Weighting (PDCW) algorithm

The Zero Crossing Amplitude Estimation (ZCAE) algorithm was recently introduced by Park [16] which is similar in some respects to work by Srinivasan *et al.* [15]. These algorithms (and similar ones by other researchers) typically analyze incoming speech in bandpass channels and attempt to identify the subset of time-frequency components for which the ITD is close to the nominal ITD of the desired sound source (which is presumed to be known *a priori*). The signal to be recognized is reconstructed from only the subset of “good” time-frequency components. This selection of “good” components is frequently treated in the computational auditory scene analysis (CASA) literature as a multiplication of all components by a binary mask that is nonzero for only the desired signal components. Although ZCAE provides impressive performance even at low SNRs, it is very computationally intensive, which makes it unsuitable for hand-held devices.

The goals of this work are twofold. First, we would like to obtain improvements in word error rate (WER) for speech recognition systems that operate in real world environments that include noise and reverberation. We also would like to develop a computationally efficient algorithm than can run in real time in embedded systems. In the present ZCAE algorithm much of the computation is taken up in the bandpass filtering operations. We found that computational cost could be significantly reduced by estimating the ITD through examination of the phase difference between the two sensors in the frequency sensors. We describe in the sections below how the binary mask is obtained using frequency information.

8.2 Phase-difference-based binary time-frequency mask estimation

Our work on signal separation is motivated by binaural speech processing. Sound sources are localized and separated by the human binaural system primarily through the use of ITD

information at low frequencies and IID information at higher frequencies, with the crossover point between these two mechanisms considered to be based on the physical distance between the two ears and the need to avoid spatial aliasing (which would occur when the ITD between two signals exceeds half a wavelength). In our work we focus on the use of ITD cues and avoid spatial aliasing by placing the two microphones closer together than occurs anatomically. When multiple sound sources are presented, it is generally assumed that humans attend to the desired signal by attending only to information at the ITD corresponding to the desired sound source.

Our processing approach, which we refer to as Phase Difference Channel Weighting (PDCW), crudely emulates human binaural processing, and is summarized in Fig. 8.1. Briefly, the system first performs a short-time Fourier transform (STFT) which decomposes the two input signals in time and in frequency. ITD is estimated indirectly by comparing the phase information from the two microphones at each frequency, and the time-frequency mask identifying the subset of ITDs that are “close” to the ITD of the target speaker is identified. A set of channels is developed by weighting this subset of time-frequency components using a series of Gammatone functions, and the time domain signal is obtained by the overlap-add method. As noted above, the principal novel feature in this paper chapter is the use of interaural phase information in the frequency domain rather than ITD, IPD, or IID information in the time domain to obtain the binary mask.

Consider the two signals that are input to the system which we refer to as $x_L[n]$ and $x_R[n]$. We assume that the location of the desired target signal is known and without loss of generality we assume its ITD to be equal to zero. For mathematical convenience, we refer to the number of interfering sources as L , with $\delta(l)$ being their respective ITDs. Note that both L and $\delta(l)$ are unknown. With the above formulations, the signals at the microphones are

$$x_L[n] = \sum_{l=0}^L x_l[n], \quad x_R[n] = \sum_{l=0}^L x_l[n - \delta(l)] \quad (8.1)$$

with $x_0[n]$ representing the target signal, $x_l(l \neq 0)$ representing interfering signals, x_L and x_R , respectively, representing the signals at the left and right microphones. The corresponding

short-time Fourier transforms can be represented as

$$X(k, m) = \sum_{n=-\infty}^{\infty} x[n]w[m-n]e^{-j2\pi kn/N} \quad (8.2)$$

$$X_L(k, m) = \sum_{i=0}^L X_i(k, m) \quad (8.3)$$

$$X_R(k, m) = \sum_{i=0}^L e^{-jw_k d_i(k, m)} X_i(k, m) \quad (8.4)$$

where $w[n]$ is a finite-duration Hamming window, k indicates one of N frequency bins, with positive frequency samples corresponding to $w_k = 2\pi k/N$ for $0 \leq k \leq N/2 - 1$. In our work N equals 512 for 26.5-ms windows and 2048 for 75-ms windows. Note that even though (8.1) indicates that signals at the microphones are identical except for a time delay, it is more appropriate that we consider the time delays associated with each frequency component of the signal. Correspondingly, we replace the frequency-independent ITD parameter δ in (8.1) by the frequency-dependent ITD parameter $d(k, m)$ in (4). Next, we assume that a specific time-frequency bin (k_0, m_0) , is dominated by a single sound source l . This leads to

$$X_L(k_0; m_0) \approx X_{l^*}(k_0, m_0) \quad (8.5)$$

$$X_R(k_0; m_0) \approx e^{-jw_{k_0} d(k_0, m_0)} X_{l^*}(k_0, m_0) \quad (8.6)$$

where the source l^* dominates the time-frequency bin (k_0, m_0) . This leads to a simple binary decision concerning whether the time-frequency bin (k_0, m_0) belongs to the target speaker or not. The frequency-dependent ITD $d(k, m)$ for a particular time-frequency bin (k_0, m_0) is

$$|d(k_0, m_0)| \approx \quad (8.7)$$

$$\frac{1}{|w_{k_0}|} \min_r |\angle X_R(k_0, m_0) - \angle X_L(k_0, m_0) - 2\pi r|$$

for positive values of w_n of positive value, as discussed above, from which we derive the binary masking criterion

$$\mu(k_0, m_0) = \begin{cases} 1, & \text{if } |d(k_0, m_0)| \leq \tau \\ \eta, & \text{otherwise} \end{cases} \quad (8.8)$$

In other words, only time-frequency bins for which $|d(k_0, m_0)| < \tau$ are presumed to belong to the target speaker. We are presently using a value of 0.01 for the floor constant η . The mask $\mu(k, m)$ in (8.11b) is applied to $\bar{X}(k, m)$, the averaged signal spectrogram from the two channels, and speech is reconstructed from the $\tilde{X}(k, m)$ where

$$\bar{X}(k, m) = \frac{1}{2}\{X_L(k, m) + X_R(k, m)\} \quad (8.9)$$

$$\tilde{X}(k, m) = \mu(k, m) \bar{X}(k, m) \quad (8.10)$$

In Figure 2 we plot typical example of spectra from a signal that is corrupted by an interfering speaker with a signal-to-interference ratio (SIR) of 5 dB. We discuss two extensions to the basic PDCW algorithm in the next section.

8.2.1 *The effect of the window length and channel weighting*

In conventional speech coding and speech recognition systems, we generally use a length of approximately 20 to 30 ms for the Hamming window $w[n]$ in order to capture effectively the temporal fluctuations of speech signals. Nevertheless, longer observation durations are usually better for estimating environmental parameters. Using the procedures described below in Sec. 8.3, we considered the effect of window length on recognition accuracy. These results, summarized in Fig. 8.3, indicate that best performance is achieved with window length of about 75 ms. In the experiments described below we use Hamming windows of duration 75 ms with 37.5 ms between successive frames.

As explained in Subsection 3.7, we can significantly enhance performance using the channel weighting approach. Instead of using the estimates produced by (8.11b), we use the procedures described in (3.7), (5.7), and (3.9). The enhanced spectrum is obtained using (3.9).

8.3 *Experimental Results*

In this section, we present experimental results for two different environmental conditions. In the first condition, we simulate different reverberant environments, where the target is masked by an interfering speaker. We used the Room Impulse Response (RIR) software

[51] for simulating the effects of room reverberation. We assumed a room of dimensions $5 \times 4 \times 3$ m, a distance between the microphone and the speaker of 2 m, with the microphone located at the center of the room. We assumed that the target source is located along the perpendicular bisector of the line between two microphones, and that the masker is 45 degrees to one side. The target and noise signals are digitally added after simulating the reverberation effects. The two microphones are placed 4 cm apart from one another. We used `sphinx_fe` included in `Sphinxbase 0.4.1` for speech feature extraction, `SphinxTrain 1.0` for speech recognition training, and `Sphinx3.8` for decoding, all of which are readily available in Open Source form. We used subsets of 1600 utterances and 600 utterances, respectively, from the DARPA Resource Management (RM1) database for training and testing.

Fig. 8.4 compares word recognition accuracy for several of the algorithms discussed in the chapter. ZCAE refers to the time-domain algorithm described in [16] with binary masking, as the better-performing continuous-masking does not work in environments with reverberation or more than one masking source. PD refers to the algorithm described in Secs. 2 and 3 of this chapter with the 75-ms analysis window but without the gammatone frequency weighting, and PDCW refers to the complete algorithm including the gammatone channel weighting (CW) described in Sec. 3.1. As can be seen, the PDCW (and to a lesser extent the PD) algorithm provides lower WER than ZCAE, and the superiority of PDCW over ZCAE increases as the amount of reverberation increases.

In our second set of experiments, we still assume that the distance between the two microphones is the same, but we added noise recorded in real environments with real two-microphone hardware in locations such as a public market, a food court, a city street and a bus stop with background speech. Fig. 8.4(d) illustrates these experimental results. Again we observe that PDCW (and to a lesser extent PD) provides much better performance than ZCAE for all conditions.

We also profiled the run times of implementations in C of the PDCW and ZCAE algorithms on two machines. The PDCW algorithms ran in only 9.03% of the time required to run the ZCAE algorithm on an 8-CPU Xeon E5450 3-GHz system, and in only 9.68% of the time to run the ZCAE algorithm on an embedded system with an ARM11 667-Mhz processor using a vector floating point unit. The major reason for the speedup is that in ZCAE the signal must be passed through a bank of 40 filters while PDCW requires only two FFTs and

one IFFT for each feature frame. A MATLAB version of PDCW with sample audio files is available at

http://www.cs.cmu.edu/~robust/archive/algorithms/PDCW_IS2009. The code in this directory was used to obtain the results described in this chapter.

8.4 *Obtaining the ITD threshold*

In the case of binary masking using the ITD threshold, we usually select the appropriate ITD threshold from the development set. However, the optimal ITD threshold itself will depend on the number of noise sources and their locations, and both of which may be time-varying. For example, if the direction of the noise source is very different from that of the target source direction, a wider ITD threshold might be more helpful. On the contrary, if the noise source is very close to the target and if we use a wide ITD threshold, then it will also pass a large portion of interference source signals as well as the target signals. If there are more than one noise sources or if the noise sources are moving, then the problem becomes even more complicated.

Thus, in our approach, we construct two complementary masks using a binary threshold. Using these two complementary masks, we obtain two different spectra: one for the target and one for the interference. From these spectra, we obtain the short-time power for the target and the interference. These power sequences are applied to nonlinearity. We compute the correlation coefficient from these power sequences. We obtain the ITD threshold by minimizing the correlation coefficient.

8.4.1 *Complementary mask generation*

In this algorithm, we obtain complementary binary masks. One mask is for selecting the target signal and the other mask is for selecting the interference signal. Thus, we can construct two different spectra. From these spectra, we obtain the power sequence from the target and interdependence.

In the case where a set T consists of a finite number of possible ITD candidates, we will determine which element of this set will be the most appropriate ITD threshold. Let us consider one element of this set τ_0 . Using this τ_0 , we obtain the target mask and the

complementary mask:

$$\mu_T(m, k) = \begin{cases} 1, & \text{if } |d(m, k)| \leq \tau_0 \\ \delta, & \text{otherwise} \end{cases} \quad (8.11a)$$

$$\mu_I(m, k) = \begin{cases} \delta, & \text{if } |d(m, k)| > \tau_0 \\ 1, & \text{otherwise} \end{cases} \quad (8.11b)$$

In other words, we assume that time-frequency bins for which $|d(m, k)| < \tau_0$ are presumed to belong to the target speaker, and time-frequency bins for which $|d(m, k)| > \tau_0$ belong to the noise source. We are presently using a value of 0.01 for the floor constant η . The masks $\mu_T(m, k)$ and $\mu_I(m, k)$ in (8.11) are applied to $\bar{X}(k, m)$, the averaged signal spectrogram from the two channels.

$$\bar{X}(k, m) = \frac{1}{2}\{X_L(k, m) + X_R(k, m)\} \quad (8.12)$$

Using this procedure, we obtain the target spectra $X_T(m, e^{j\omega_k}|\tau_0)$ and the interference spectra $X_I(m, e^{j\omega_k}|\tau_0)$ as shown below:

$$X_T(m, e^{j\omega_k}|\tau_0) = \bar{X}(m, e^{j\omega_k})\tilde{\mu}_T(m, k) \quad (8.13a)$$

$$X_I(m, e^{j\omega_k}|\tau_0) = \bar{X}(m, e^{j\omega_k})\tilde{\mu}_I(m, k) \quad (8.13b)$$

In the above equation, we explicitly include τ_0 to show that the masked spectrum will depend upon the ITD threshold. Using these spectra $X_T(m, e^{j\omega_k})$ and $X_I(m, e^{j\omega_k})$, we obtain the power:

$$P_T(m|\tau_0) = \sum_{k=0}^{N-1} \left| X_T(m, e^{j\omega_k}) \right|^2 \quad (8.14a)$$

$$P_I(m|\tau_0) = \sum_{k=0}^{N-1} \left| X_I(m, e^{j\omega_k}) \right|^2 \quad (8.14b)$$

In the next subsection, we will discuss how to obtain the optimal τ_0 from the above equations.

8.4.2 Obtaining the ITD threshold using the minimum correlation criterion

It is well known that the perceived loudness of a sound source is not proportional to the intensity of that sound source (*e.g.* [59]). To represent the relationship between the intensity

and the perceived loudness, many nonlinearity models have been proposed. The most widely used form of these nonlinearities are the logarithmic nonlinearity and the power-law nonlinearity (*e.g.* [43]). The importance of auditory threshold in speech recognition has been confirmed in our previous research works (*e.g.*[53][35]). Thus, we use the following power-law nonlinearity:

$$R_T(m|\tau_0) = P_T(m|\tau_0)^{a_0} \quad (8.15a)$$

$$R_I(m|\tau_0) = P_I(m|\tau_0)^{a_0} \quad (8.15b)$$

where we use $a_0 = 1/15$ as the power coefficient as in [33].

From (8.15), the correlation coefficient using (8.15) is obtained as follows:

$$\rho_{T,I}(\tau_0) = \frac{\frac{1}{N} \sum_{m=1}^M R_T(m|\tau_0)R_I(m|\tau_0) - \mu_{R_T}\mu_{R_I}}{\sigma_{R_T}\sigma_{R_I}} \quad (8.16)$$

where σ_{R_T} and σ_{R_I} are standard deviations of $R_T(m|\tau_0)$ and $R_I(m|\tau_0)$ respectively, and μ_{R_I} and μ_{R_2} are means of $R_T(m|\tau_0)$ and $R_I(m|\tau_0)$ respectively.

Thus, the threshold τ_0 is selected to minimize the absolute value of the crosscorrelation.

$$\hat{\tau}_0 = \arg \min_{\tau_0} |\rho_{T,I}(\tau_0)| \quad (8.17)$$

8.4.3 Experimental Results

In this section, we present experimental results using the ITD threshold selection algorithm proposed in this paper. We compare the PD (Phase Difference) binary masking system using a fixed ITD threshold with another PD system that uses the ITD threshold algorithm proposed in this paper. In all the following experiments, we assumed a room of dimensions 5 x 4 x 3 m, and the microphone is located at the center of the room. The target is 2 m away from the microphone along the perpendicular bisector of the line between two microphones. The target and noise signals are digitally added after simulating the reverberation effects. The two microphones are placed 4 cm apart from one another. We used sphinx fe included in Sphinxbase 0.4.1 for speech feature extraction, SphinxTrain 1.0 for speech recognition training, and Sphinx3.8 for decoding, all of which are readily available in Open Source form. We used subsets of 1600 utterances and 600 utterances, respectively, from the DARPA Resource Management (RM1) database for training and testing.

For the fixed ITD threshold system, we obtained the optimal threshold by conducting an experiment in a specific environment: We located the interfering speaker along a 45 degree line to the side of the perpendicular bisector of the line between two microphones, and the interfering speaker generating a speech noise of 0 dB Signal-to-Interference Ratio (SIR). We further assumed that there was no reverberation in this room.

We conducted two different sets of experiments. In the first set of the experiments, we kept the geometrical configuration the same as the above, but we only change the Signal-to-Interference Ratio (SIR) and the reverberation time. To simulate the reverberation effects, we used the Room Impulse Response (RIR) software [51]. As shown in Fig. 8.6, in no reverberation at 0-dB SIR environment, both the fixed ITD PD and the automatic ITD PD systems show comparable performance. However, if the reverberation occurs, then the automatic ITD system shows substantially better performance than the fixed ITD PD system.

In the second set of the experiments, we changed the location of the interfering speaker while maintaining the SIR level at 0 dB. As shown in Fig. 8.7, even if the SIR is the same as the calibration environment, the fixed ITD threshold PD system shows significantly degraded performance if the actual interfering speaker location is different from the calibration environment. However, the automatic ITD threshold selection system shows much more robust recognition results.

8.4.4 Conclusion

In this section we present a new algorithm which selects an ITD threshold by minimizing the correlation of nonlinearity power from the masked and non-masked spectral regions. Experimental results show while the conventional fixed ITD threshold system shows degraded performance in unmatched conditions, this automatic ITD threshold selection algorithm makes the binary mask system much more reliable.

8.5 *PROPOSED WORK*

8.6 *Threshold selection algorithm*

In the previous chapter, we discussed the threshold selection algorithm for the PD system. We will apply the same idea to the PDCW system as well. We will also investigate the online threshold selection algorithm.

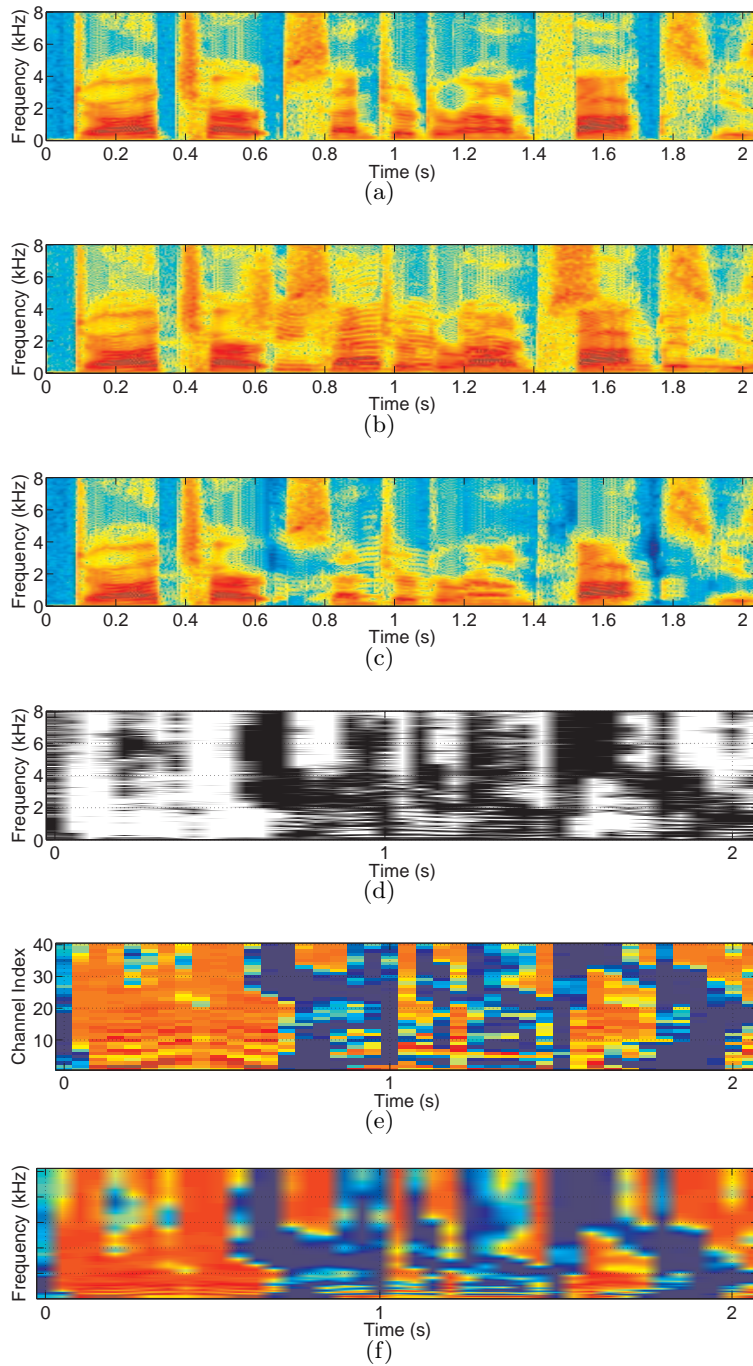


Fig. 8.2: Sample spectrograms illustrating the effects of PDCW processing. (a) original clean speech, (b) noise-corrupted speech, (c) reconstructed (enhanced) speech (d) the time-frequency mask obtained with (8.11b) (e) gammatone channel weighting obtained from the time-frequency mask in (3.7) (e) final frequency weighting shown in (5.7) (f) enhanced speech spectrogram using the entire PDCW algorithm

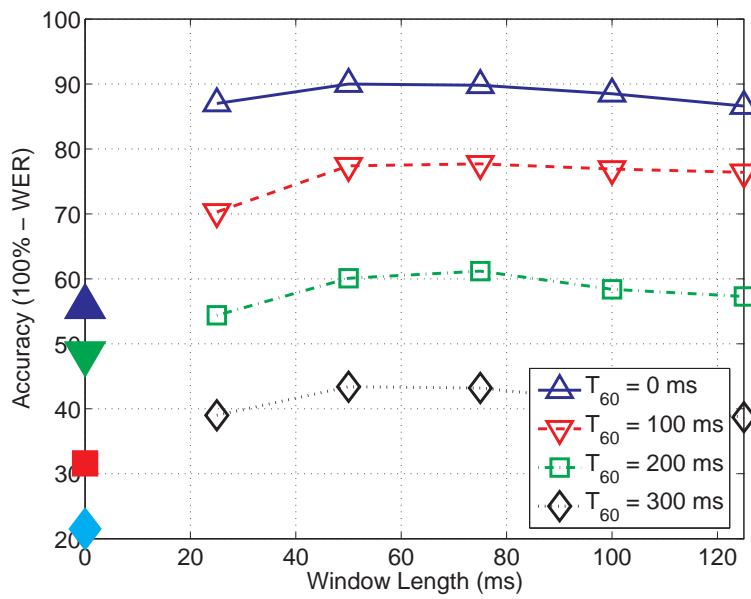


Fig. 8.3: The dependence of word recognition accuracy (100% - WER) on the window length, using an SIR of 10 dB and various reverberation times. The filled symbols at 0 ms represent baseline results obtained with a single microphone.

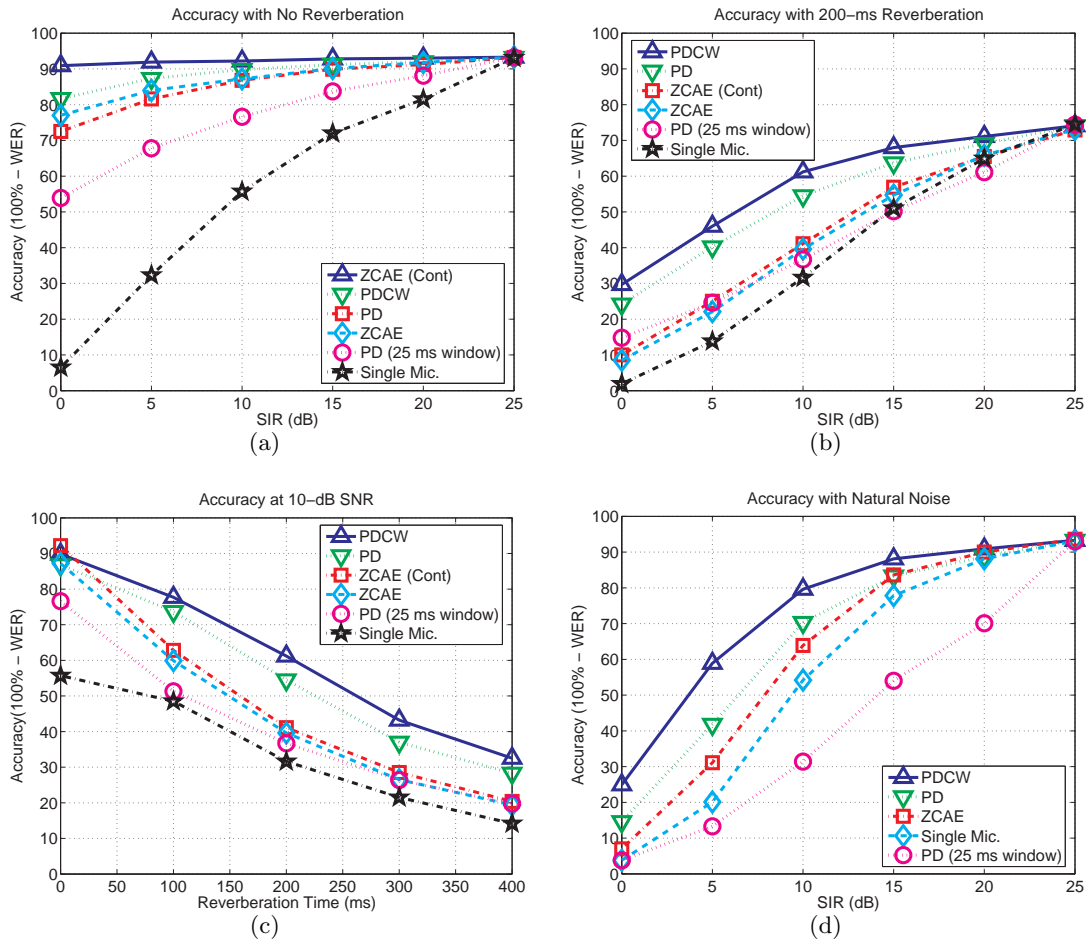


Fig. 8.4: Speech recognition accuracy using different algorithms (a) in the presence of an interfering speech source as a function of SNR in the absence of reverberation, (b,c) in the presence of reverberation and speech interference, as indicated, and (d) in the presence of natural real-world noise.

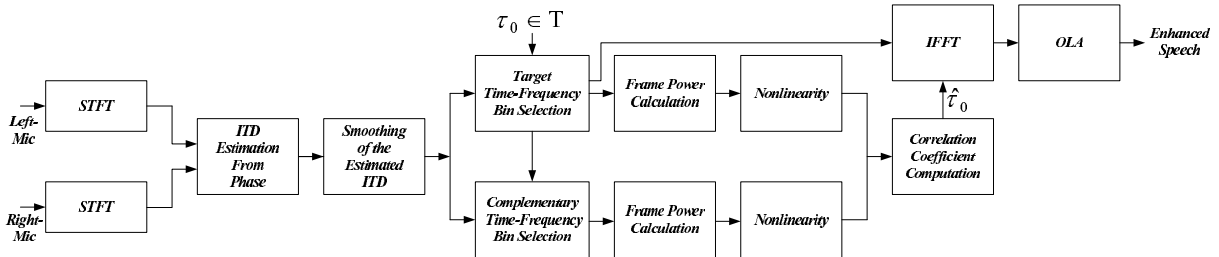


Fig. 8.5: The block diagram of the optimal ITD selection algorithm for sound source separation

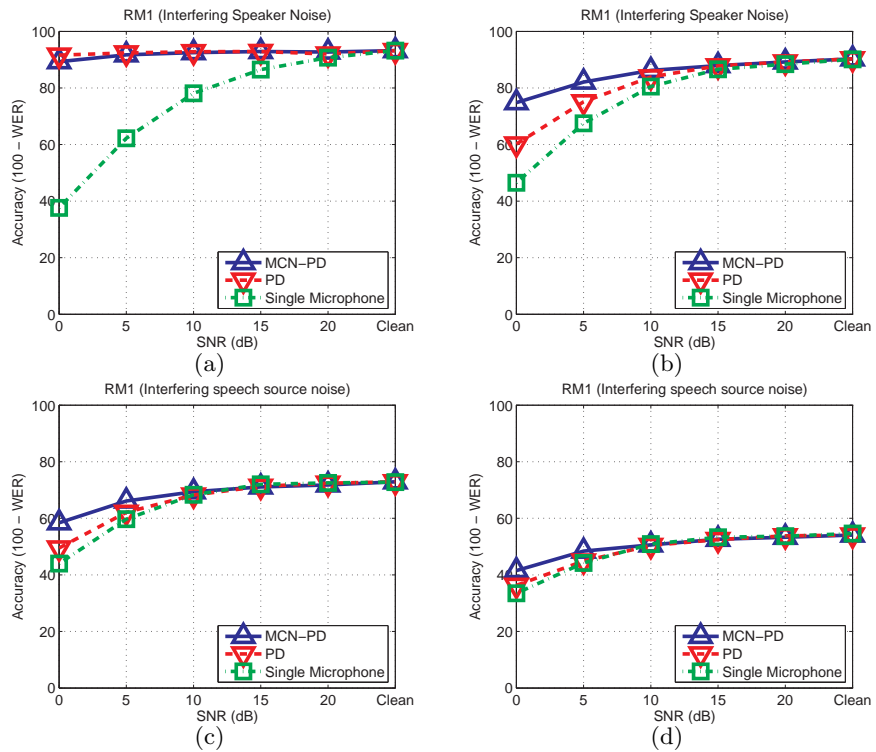


Fig. 8.6: Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at 45 degrees at different reverberation times (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms.

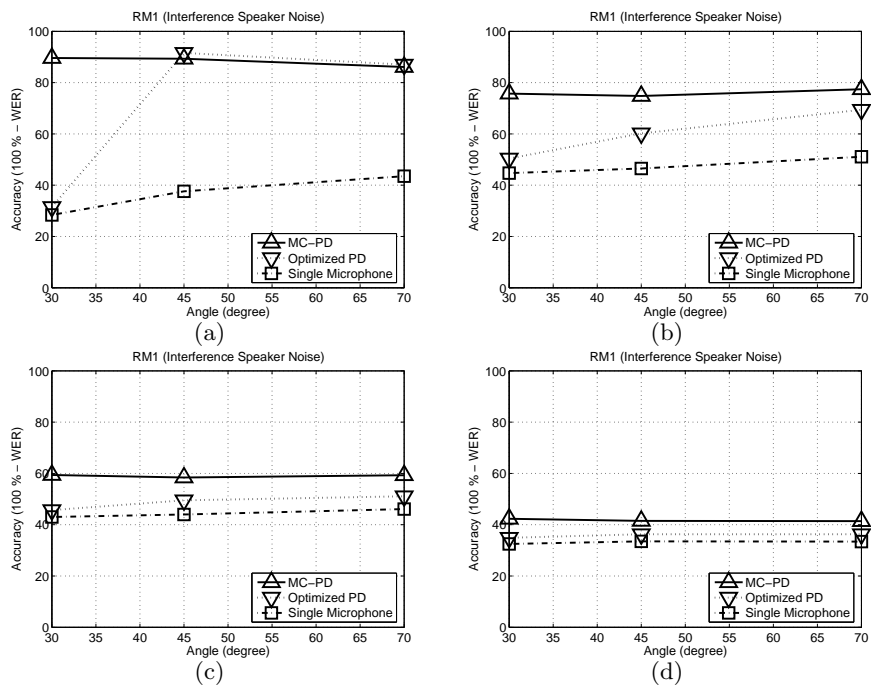


Fig. 8.7: Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at different locations at different reverberation time (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms.

9. THESIS GOAL AND TIME TABLE

9.1 Deliverables

We will provide the following deliverables:

- 1) A tool which can be used to test different auditory nonlinearities.

We tested the effects of various nonlinearities. While we were conducting experiments for our 2006 INTERSPEECH paper [43], we first noticed that auditory nonlinearity has a very important role in robust speech recognition. More importantly, we observed that the benefit of auditory nonlinearity comes from the threshold part of the rate-nonlinearity curve while the saturation part of the curve does not provide any substantial benefit. The effect of auditory threshold can also be achieved using the power function. We compared different nonlinearities and compared the speech recognition performance.

- 2) Small Power Boosting Algorithm

We observed that if we use compressive nonlinearity, then time-frequency bins with smaller power are more susceptible to additive noise. In the conventional approach, we enhance robustness by performing spectral subtraction on the test set. In our approach, if these time-frequencies are more susceptible, then we can intentionally boost power for these time frequency bins to make them more robust. Suitable smoothing is also critically important for obtaining good performance. Experimental results show that this idea is especially helpful for difficult environments like music noise.

- 3) Power Function Based Power Distribution (PPDN)

The current implementation of PPDN is working reasonably well for some databases like Resource Management 1, but the performance is degraded if the utterance includes some extra silences. We will incorporate a VAD to tackle this problem.

- 4) Missing feature speech enhancement system using longer windows

As an example of a temporal resolution study, we will conduct experiments on speech enhancement using a missing feature algorithm with a longer window. The Gaussian mixture model (GMM) is obtained from log spectral density with longer windows. Noisy speech is compensated using this GMM. However, we cannot directly use a longer window for feature generation since it loses temporal resolution. Thus, we re-synthesize speech and apply a feature extraction algorithm with a shorter window. This system has not been finished yet, but we believe it is a valid idea.

5) Signal-to-Noise Ratio Estimation Algorithm

We observe that the power distribution is highly correlated with the corruptness of speech. Qualitatively speaking, this is because speech has usually sharp distribution while noise has flat distribution. We found AM-to-GM ratio is a good way of quantitative measuring the sharpness of distribution. Especially, if a distribution obeys the gamma distribution, then the shaping parameter of the distribution is a function of the AM-to-GM ratio. From this idea, we developed a new noise estimation algorithm.

6) A VAD algorithm

Using the power distribution information, we developed a VAD algorithm using a hypothesis testing skill. While very simple in structure, we found it shows robustness in difficult environments. We will refine the algorithm.

7) Phase Difference Channel Weighting

We proposed an algorithm for binary source separation using the phase difference information. We observed that suitable smoothing is very important for obtaining good performance. This approach has shown better performance than ZCAE in omni-directional noise and in reverberant environments while using smaller computational amount.

8) Automatic ITD threshold selection algorithm

Conventionally, an ITD threshold has been obtained empirically using a certain development set. However, we observe that the optimal ITD threshold depends on the environment. In this algorithm, we consider two complementary masks: one from the target and the other from everything except the target. Thus, if the ITD threshold is an optimal one, the cross correlation between these power signals should be minimized. Because power has a very large dynamic range, we observe that before computing the cross-correlation, applying nonlinearity is important.

9) A study on the effect of variance in the HMM model

We found that increasing the variance of the HMM model with a small c_0 mean has a huge impact in difficult environments. The effect of variance flooring has been known for a long time, but we will intentionally boost the variance of “vulnerable” HMM models. We can distinguish more vulnerable HMM models by examining their c_0 mean.

10) Online implementation of the PNCC

Our objective with PNCC is to provide desirable performance for all kinds of noise while obtaining good performance for clean speech as well. PNCC is motivated by many of the psycho-acoustical observations: Gammatone weighting, power-law nonlinearity obtained from a physiological model, band-pass filtering for non-speech portions, high-pass filtering for speech portions, and temporal masking effect.

11) De-reverberation

We observe that the falling edge of power contours is vulnerable under reverberation environments. However, for human listeners, due to the precedence effect, the rising edge is more important. Motivated by this, we developed an efficient de-reverberation technique.

9.2 Timetable

Task	Start Date	End Date	Duration
Improving Power-Function Based Power Equalization Algorithm	June 2010	July 2010	1 month
Doing experiments on VAD	July 2010	Aug 2010	1 month
Refining nonlinearity study and refining SPB (small power boosting)	Aug 2010	Nov 2010	3 months
Investigation into the effect of window length for missing feature	Nov 2010	Dec 2011	2 months
Investigation into the on-line SNR estimation algorithm and incorporating into missing feature	Jan 2011	Feb 2011	2 months
Implementation and incorporation into Sphinx	Feb 2011	March 2011	2 months
Dissertation write-up	March 2011	Apr 2011	2 months

BIBLIOGRAPHY

- [1] M. G. Heinz, X. Zhang, I. C. Bruce, and, L. H. Carney, “Auditory nerve model for predicting performance limits of normal and impaired listeners,” *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 91–96, July 2001.
- [2] H. G. Hirsch, P. Meyer , and H. W. Ruehl, “Improved speech recognition using high-pass filtering of subband envelopes,” in *EUROSPEECH '91*, Sept. 1991, pp. 413–416.
- [3] H. Hermansky and N. Morgan, “RASTA processing of speech,” *IEEE. Trans. Speech Audio Process.*, vol. 2, no. 4, Oct. 1994.
- [4] M. G. Heinz, X. Zhang, I. C. Bruce, and, L. H. Carney, “Auditory nerve model for predicting performance limits of normal and impaired listeners,” *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 91–96, July 2001.
- [5] X. Huang, A. Acero, H-W Won, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [6] P. Pujol, D. Macho, and C. Nadeu, “On real-time mean-and-variance normalization of speech recognition features,” in *IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 1, May 2006, pp. 773–776.
- [7] A. Acero, and R. M. Stern, “Environmental Robustness in Automatic Speech Recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (Albuquerque, NM)*, vol. 2, Apr. 1990, pp. 849–852.
- [8] P. J. Moreno, B. Raj, and R. M. Stern, “A vector Taylor series approach for environment-independent speech recognition,” in *IEEE Int. Conf. Acoust., Speech and Signal Processing*, May. 1996, pp. 733–736.
- [9] S. F. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Trans. Acoust., Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.
- [10] R. Singh, R. M. Stern, and B. Raj, “Signal and feature compensation methods for robust speech recognition,” in *Noise Reduction in Speech Applications*, G. M. Davis, Ed. CRC Press, 2002, pp. 219–244.
- [11] R. Singh, B. Raj, and R. M. Stern, “Model compensation and matched condition methods for robust speech recognition,” in *Noise Reduction in Speech Applications*, G. M. Davis, Ed. CRC Press, 2002, pp. 245–275.
- [12] B. Raj, V. N. Parikh, and R. M. Stern, “The effects of background music on speech recognition accuracy,” in *IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 2, Apr. 1997, pp. 851–854.
- [13] C. Kim, Y.-H. Chiu, and R. M. Stern, “Physiologically-motivated synchrony-based processing for robust automatic speech recognition,” in *INTERSPEECH-2006*, Sept. 2006, pp. 1975–1978.
- [14] B. Raj and R. M. Stern, “Missing-Feature Methods for Robust Automatic Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 101–116, Sept. 2005.

- [15] S. Srinivasan, M. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Comm.*, vol. 48, pp. 1486–1501, 2006.
- [16] H. Park, and R. M. Stern, "Spatial separation of speech signals using amplitude estimation based on interaural comparisons of zero crossings," *Speech Communication*, vol. 51, no. 1, pp. 15–25, Jan. 2009.
- [17] R. M. Stern, E. Gouvea, C. Kim, K. Kumar, and H. Park, "Binaural and multiple-microphone signal processing motivated by auditory perception," in *Hands-Free Speech Communication and Microphone Arrays, 2008*, May. 2008, pp. 98–103.
- [18] R. M. Stern and C. Trahiotis, "Models of binaural interaction," in *Hearing*, B. C. J. Moore, Ed. Academic Press, 2002, pp. 347–386.
- [19] H. S. Colburn and A. Kulkarni, "Models of sound localization," in *Sound Source Localization*, A. N. Popper and R. R. Fay, Eds. Springer-Verlag, 2005, pp. 272–316.
- [20] J. Volkman, S. S. Stevens, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch (A)," *J. Acoust. Soc. Am.*, vol. 8, no. 3, pp. 208–208, Jan 1937.
- [21] E. Zwicker, "Subdivision of the audible frequency range into critical bands," *J. Acoust. Soc. Am.*, vol. 33, no. 2, pp. 248–248, Feb 1961.
- [22] B. C. J. Moore and B. R. Glasberg, "A revision of Zwicker's loudness model," *Acustica - Acta Acustica*, vol. 82, pp. 335–345, 1996.
- [23] H. Hermansky, "Perceptual linear prediction analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.
- [24] A. V. Oppenheim and R. W. Scafer, with J. R. Buck, *Discrete-time Signal Processing*. Englewood-Cliffs, NJ: Prentice-Hall, 1999.
- [25] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood-Cliffs, NJ: Prentice-Hall, 1978.
- [26] S. S. Stevens, "On the psychophysical law," *Psychological Review*, vol. 64, no. 3, pp. 153–181, 1957.
- [27] B. G. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. New York: John Wiley & Sons, Inc., 2000.
- [28] D. Ellis. (2006) Plp and rasta (and mfcc, and inversion) in matlab using melfcc.m and invmelfcc.m. [Online]. Available: <http://labrosa.ee.columbia.edu/matlab/rastamat/>
- [29] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 1995, pp. 153–156.
- [30] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, vol. 25, no. 1.
- [31] R. Drullman, J. M. Festen and R. Plomp, "Effect of temporal envelope smearing on speech recognition," *J. Acoust. Soc. Am.*, vol. 95, no. 2, pp. 1053–1064, Feb. 1994.
- [32] —, "Effect of reducing slow temporal modulations on speech recognition," *J. Acoust. Soc. Am.*, vol. 95, no. 5, pp. 2670–2680, May 1994.
- [33] C. Kim and R. M. Stern, "Feature extraction for robust speech recognition based on maximizing the sharpness of the power distribution and on power flooring," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2010, pp. 4574–4577.

- [34] —, “Power function-based power distribution normalization algorithm for robust speech recognition,” in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 188–193.
- [35] C. Kim, K. Kumar and R. M. Stern, “Robust speech recognition using small power boosting algorithm,” in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 243–248.
- [36] X. Huang, A. Acero, H-W Won, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [37] O. Vikki, and K. Laurila, “Cepstral domain segmental feature vector normalization for noise robust speech recognition,” *Speech Communication*, vol. 25, pp. 133–147, Aug. 1998.
- [38] M. C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajari, N. Morgan, S. Sivasdas, “Robust asr front-end using spectral-based and discriminant features: experiments on the aurora tasks,” in *EUROSPEECH-2001*, Sept. 2001, pp. 429–432.
- [39] Y. Obuchi, N. Hataoka, and R. M. Stern, “Normalization of time-derivative parameters for robust speech recognition in small devices,” *IEICE Transactions on Information and Systems*, vol. 87-D, no. 4, pp. 1004–1011, Apr. 2004.
- [40] R. M. Stern, B. Raj, and P. J. Moreno, “Compensation for environmental degradation in automatic speech recognition,” in *Proc. of the ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, Apr. 1997, pp. 33–42.
- [41] B. Raj, V. N. Parikh, and R. M. Stern, “The effects of background music on speech recognition accuracy,” in *IEEE Int. Conf. Acoust., Speech and Signal Processing*, vol. 2, Apr. 1997, pp. 851–854.
- [42] J. W. Strutt (Lord Rayleigh), “On our perception of sound direction,” *Philosophical Magazine*, vol. 13, pp. 214–232, 1907.
- [43] C. Kim, K. Kumar, B. Raj, and R. M. Stern, “Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain,” in *INTERSPEECH-2009*, Sept. 2009.
- [44] M. Slaney, “Auditory Toolbox Version 2,” *Interval Research Corporation Technical Report*, vol. 1998, no. 010, 1998.
- [45] D. M. Green, *An Introduction to Hearing, 6th edition*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., Publishers, 1976.
- [46] B. G. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. New York: John Wiley & Sons, Inc., 2000.
- [47] X. Zhang, M. G. Heing, I. C. Bruce, and, L. H. Carney, “A phenomenological model for the responses of auditory-nerve fibers: I. Nonlinear tuning with compression and suppression,” *J. Acoust. Soc. Am.*, vol. 109, no. 2, pp. 648–670, Feb 2001.
- [48] C. Kim, Y.-H. Chiu, and R. M. Stern, “Physiologically-motivated synchrony-based processing for robust automatic speech recognition,” in *INTERSPEECH-2006*, Sept. 2006, pp. 1975–1978.
- [49] H. Hermansky, “Perceptual linear prediction analysis of speech,” *J. Acoust. Soc. Am.*, vol. 87(4), no. 4, pp. 1738–1752, Apr. 1990.
- [50] S. S. Stevens, “On the psychophysical law,” *Psychological Review*, vol. 64, no. 3, pp. 153–181, 1957.
- [51] S. G. McGovern, “A model for room acoustics,” <http://2pi.us/rir.html>.

- [52] B. Raj, M. L. Seltzer, and R. M. Stern, “Reconstruction of Missing Features for Robust Speech Recognition,” *Speech Communication*, vol. 43, no. 4, pp. 275–296, Sept. 2004.
- [53] C. Kim and R. M. Stern, “Feature extraction for robust speech recognition using a power-law nonlinearity and power-bias subtraction,” in *INTERSPEECH-2009*, Sept. 2009.
- [54] D. Kim, “Auditory processing of speech signals for robust speech recognition in real-world noisy environments,” *IEEE Trans. Speech and Audio Processing*, vol. 7, no. 1, pp. 55–69, Jan. 1999.
- [55] P. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. H. Allerhand, “Complex sounds and auditory images,” in *Auditory and Perception*. Oxford, UK: Y. Cazals, L. Demany, and K. Horner, (Eds), Pergamon Press, 1992, pp. 429–446.
- [56] C. Kim and R. M. Stern, “Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis,” in *INTERSPEECH-2008*, Sept. 2008, pp. 2598–2601.
- [57] X. Zhang, M. G. Heing, I. C. Bruce, and L. H. Carney, “A phenomenological model for the responses of auditory-nerve fibers: I. Nonlinear tuning with compression and suppression,” *J. Acoust. Soc. Am.*, vol. 109, no. 2, pp. 648–670, Feb 2001.
- [58] P. Arabi and G. Shi, “Phase-based dual-microphone robust speech enhancement,” *IEEE Tran. Systems, Man, and Cybernetics-Part B*, vol. 34, no. 4, pp. 1763–1773, Aug. 2004.
- [59] D. M. Green, *An Introduction to Hearing, 6th edition*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., Publishers, 1976.