SIGNAL PROCESSING FOR ROBUST SPEECH RECOGNITION

MOTIVATED BY AUDITORY PROCESSING

CHANWOO KIM

September 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 use mathematical representations that are motivated by human auditory processing to improve the accuracy of automatic speech recognition systems.

In our work we focus on five aspects of auditory processing. We first note that nonlinearities in the representation, and especially the nonlinear threshold effect, appear to play an important role in speech recognition. The second aspect of our work is a reconsideration of the impact of time-frequency resolution based on the observations that the best estimates of attributes of noise are obtained using relatively long observation windows, and that frequency smoothing provide significant improvements to robust recognition. Third, 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. We also consider the effects of temporal masking and the precedence effect for the processing of speech in reverberant environments and in the presence of a single interfering speaker. Finally, we exploit the excellent performance provided by the human binaural system in providing spatial analysis of incoming signals to develop signal separation systems using two microphones.

Throughout this work we propose a number of signal processing algorithms that are motivated by these observations and can be realized in a computationally efficient fashion using real-time online processing. We demonstrate that these approaches are effective in improving speech recognition accuracy in the presence of various types of noisy and reverberant environments.

CONTENTS

LIST OF FIGURES

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) [6]. 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) [6] [7] 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) [8] and Vector Taylor Series (VTS) [9] belong to these kinds of idea. Spectral subtraction [10] is subtracting the noise spectrum in the spectrum domain.

Even though a number of algorithms have shown improvements for stationary noise (*e.g.*[11, 12]), improvement in non-stationary noise remains a difficult issue (*e.g.* [13]). In these environments, auditory processing (*e.g.*[**?**]) and missing-feature-based approaches (*e.g.*[14]) are promising. In [**?**], 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 8, 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 9, 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 [5]. The popular Mel Frequency Cepstral Coefficients (MFCC) incorporate the MEL scale, which is represented by the following equation:

$$Mel(f) = 2595 \log(1 + f/700) \tag{2.1}$$

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]:

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

In PLP [22], the Bark-Frequency relation is based on the transformation given by Schroeder:

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

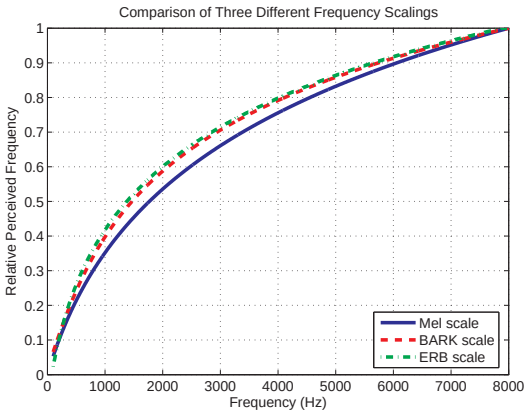


Fig. 2.1: The comparison between the MEL, Bark, and the ERB scales

Later on, Moore and Glasberg [5] 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:

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

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.* [23]). 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. [24].

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 [25]. This rule relates intensity and perceived loudness by fitting data from multiple observers using a power function:

$$L = (I/I_0)^3 \tag{2.5}$$

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:

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

The commonly used value of p_{ref} is 20μPa, 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

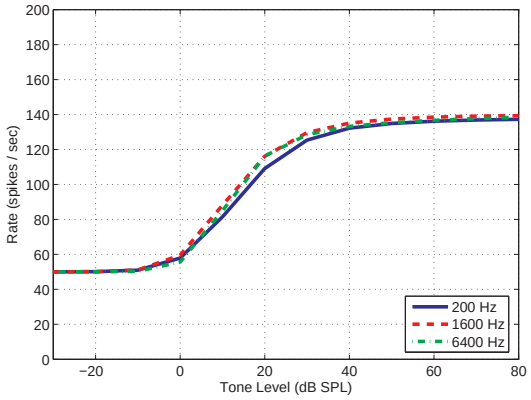


Fig. 2.2: The intensity-rate relation in the human auditory system simulated by the model proposed by M. Heinz. et. al. [1]

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) [22]. 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

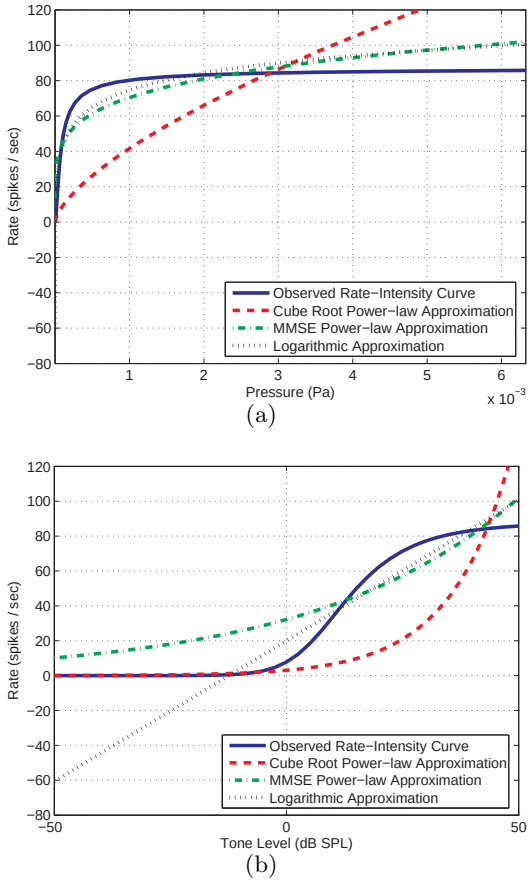


Fig. 2.3: Cube-root power law nonlinearity, MMSE power-law nonlinearity, and logarithmic nonlinearity are compared. Plots are shown on two different scales: 2.3(a) in Pa and 2.3(b) in dB Sound Pressure Level (SPL).

[22]. 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 [26].

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 `sphinxe_fe` included in `sphinxbase 0.4.1`. For PLP

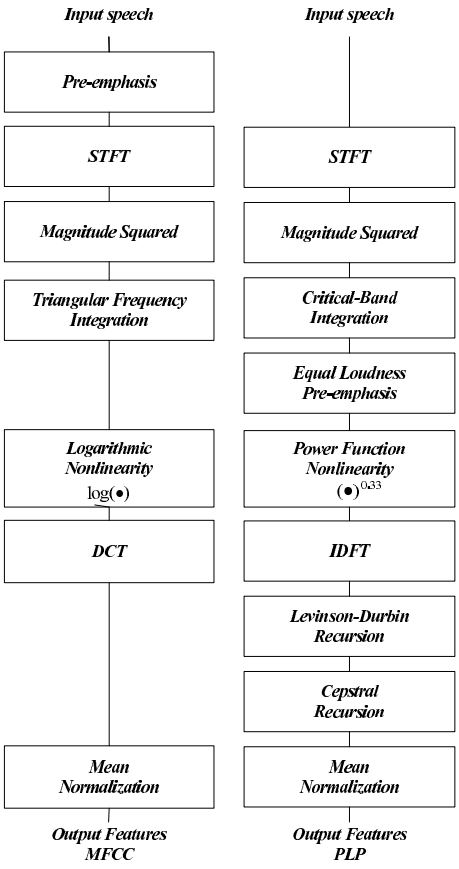


Fig. 2.4: The block diagram of MFCC and PLP

processing, we used both `HTK 3.4` and the matlab package provided by the D. Ellis group [27]. 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 [10]. 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:

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

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

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

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.* [29] [30] [31]). On the other hand, very slowly changing components (*e.g.* less than 5 Hz) are usually related to noisy sources (*e.g.*[32] [33] [34]). 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:

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

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:

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

where $Y(m,l)$

$$Y(m,l) = P(m,l) - \mu_P(m,l) \tag{2.11}$$

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 [6].

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.* [30] [31]). 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.

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

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.* [6]), 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,

$$y = \log(1 + Jx) \tag{2.13}$$

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 then the window length [35]. 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:

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

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:

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

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.* [7] [36] [37]). In the on-line CMN, the mean of the cepstral vector is updated recursively.

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

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.* [6] [7]). 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.* [7]). 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.

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

In the above equation, $F_{c_i^{te}}$ denotes the CDF of the current test utterance and $F^{-1}_{c_i^{tr}}$ 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, $F^{-1}_{c_i^{tr}}$ 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 [38]. This approach is called DCN (delta cepstrum normalization) [38].

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 [39]. This assumption can be represented by the following equation:

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

Noise compensation can be done either in the log spectral domain [9] or in the cepstral domain [8]. 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:

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

This equation can be rewritten in the form of

$$z = x + q + r(x, n, q) = x + f(x, n, q) \tag{2.20}$$

where $f(x, n, q)$ is called the "environment function" [39].

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.

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

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

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

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 [40].

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.

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

$$f(n) = N(\mu_n, |\Sigma_n) \tag{2.24}$$

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.

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

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

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

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.

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

$$= \int x p(x|z) dx \tag{2.28}$$

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.* [41]) 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.* [42] [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 [42], 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][42]. Another problem is that this algorithm requires large computation[42], 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.

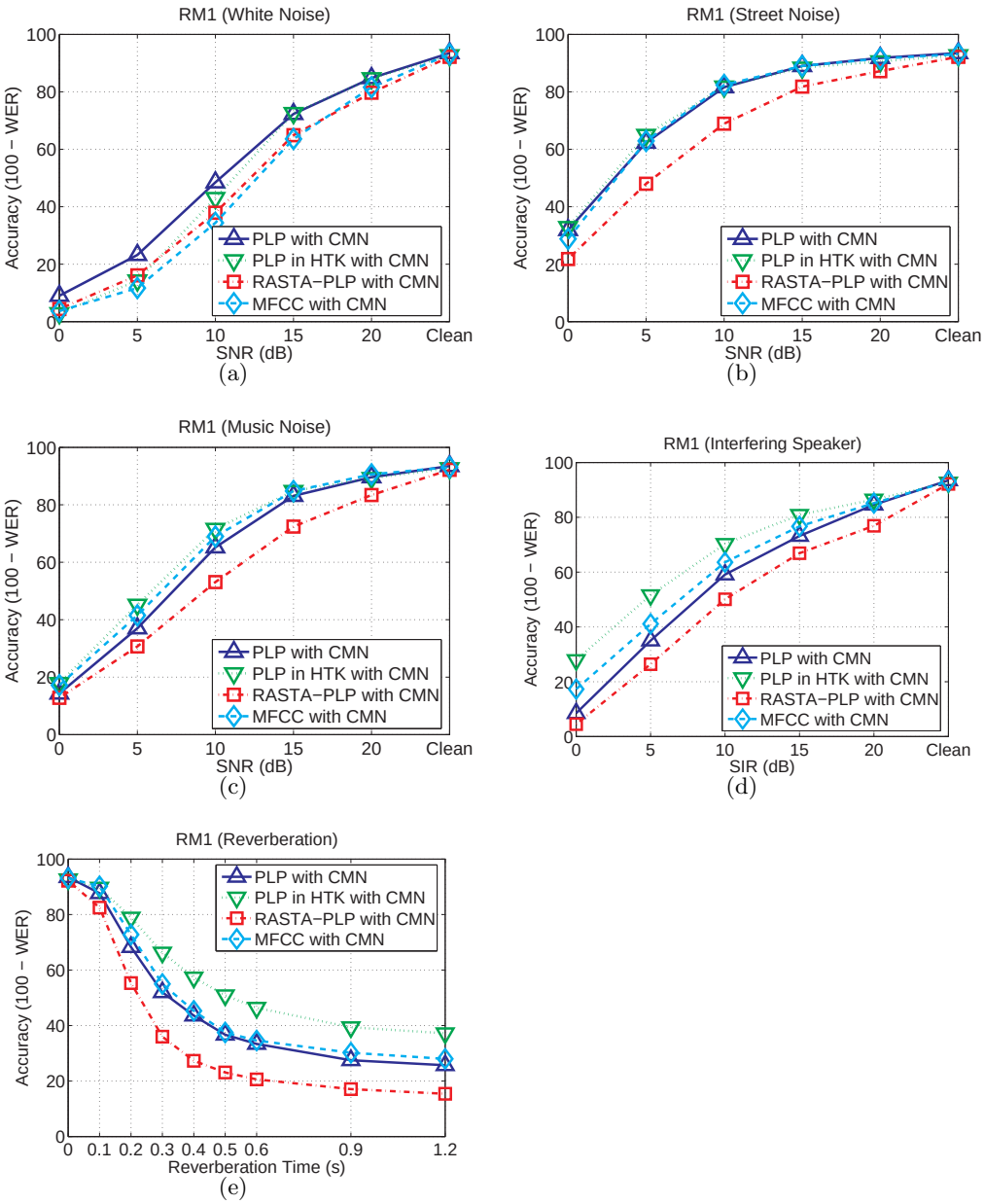


Fig. 2.5: Comparison between MFCC and PLP in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

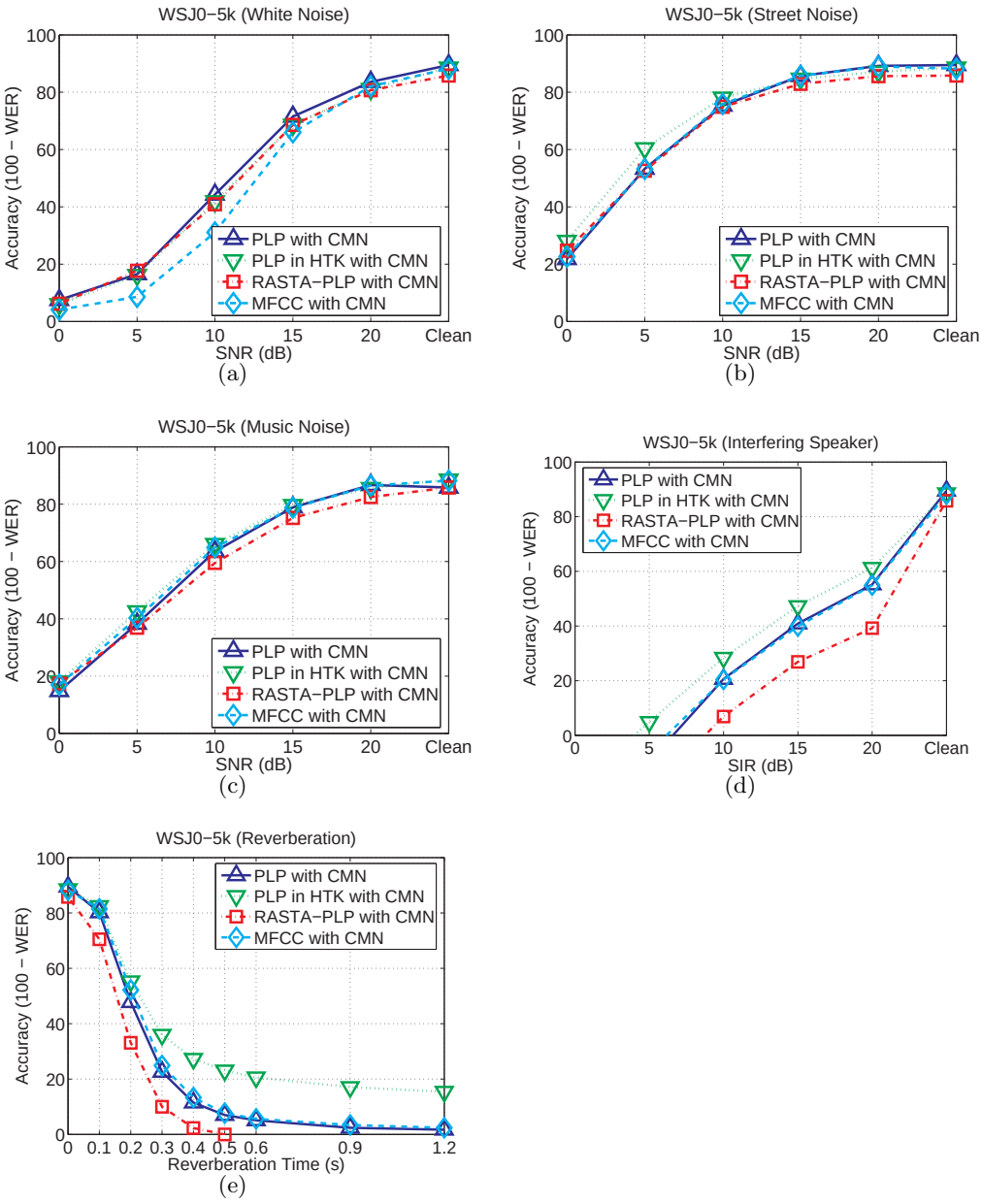


Fig. 2.6: Comparison between MFCC and PLP in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

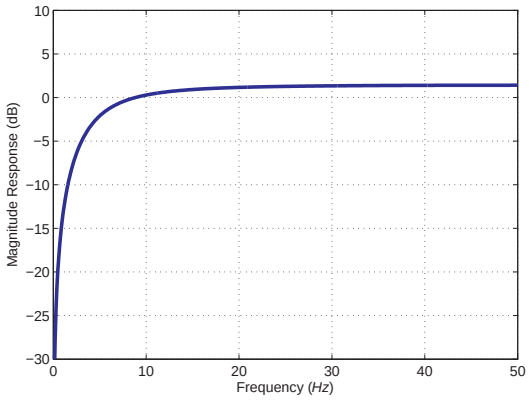


Fig. 2.7: The frequency response of the high-pass filter proposed by Hirsch et al. [2]

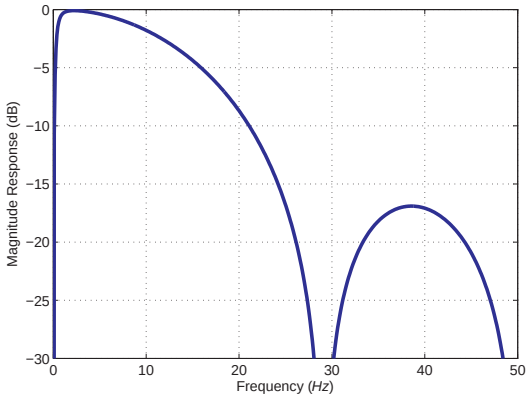


Fig. 2.8: The frequency response of the band-pass filter proposed by Hermansky et al. [3]

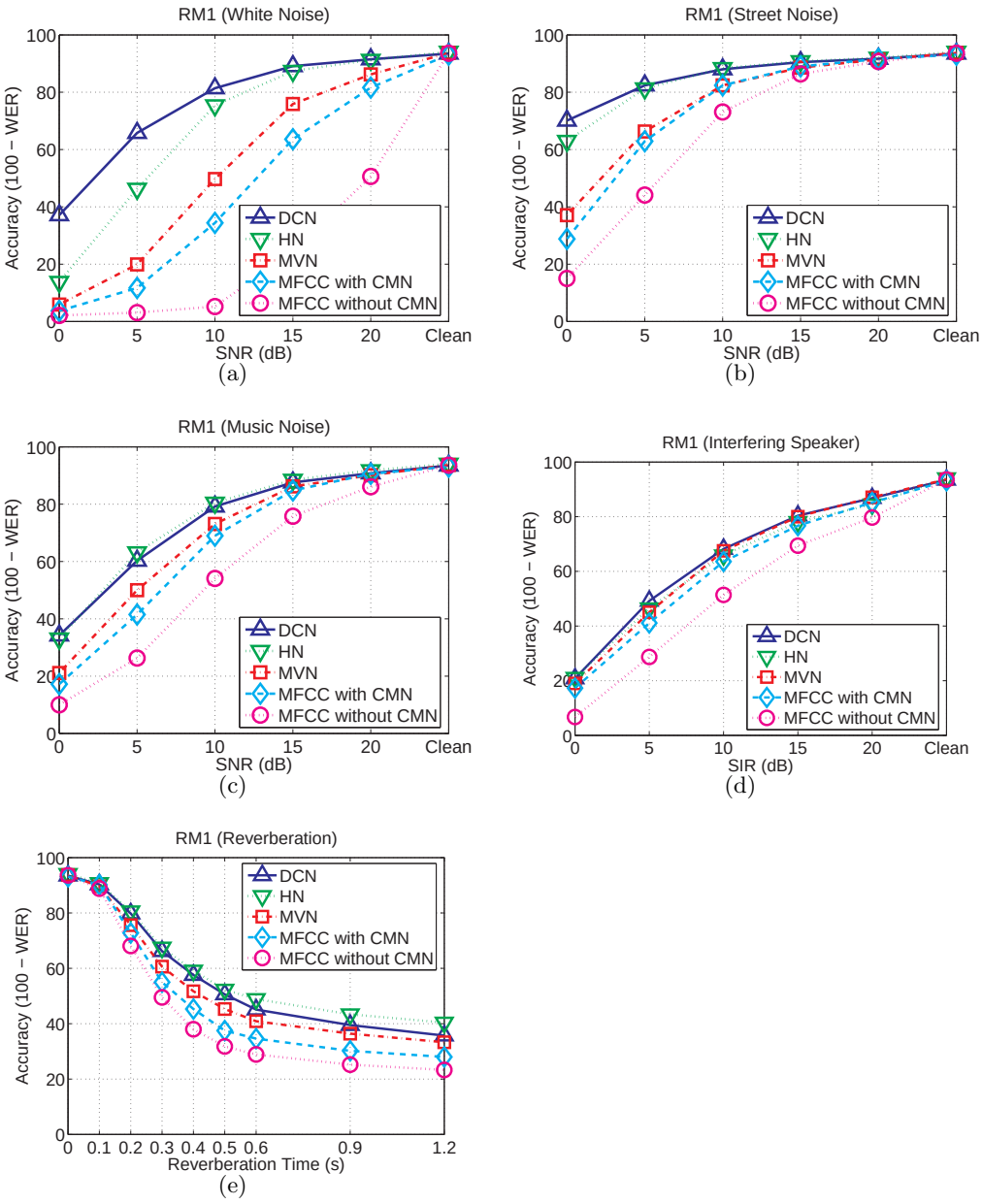


Fig. 2.9: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

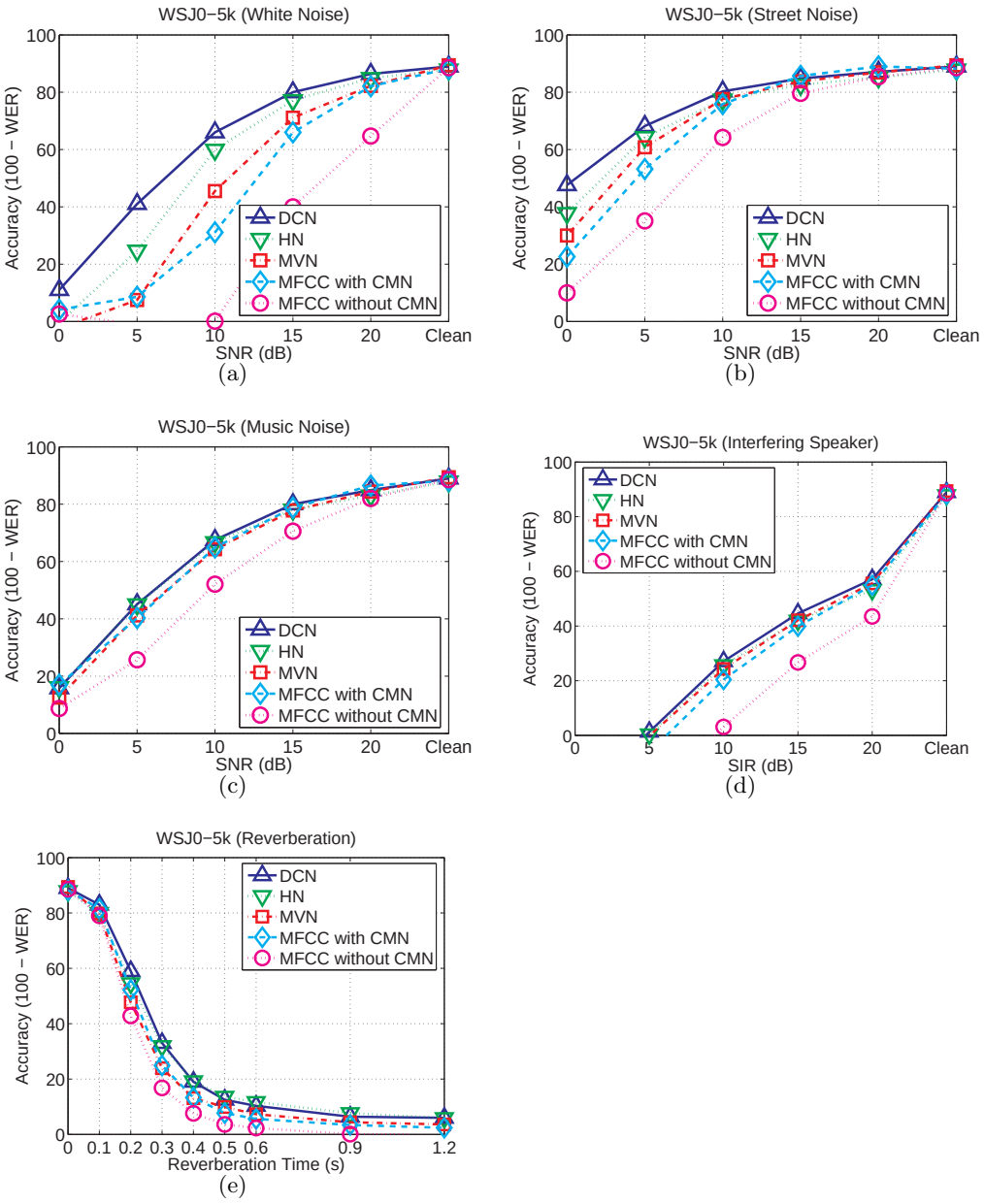


Fig. 2.10: Comparison between different normalization approaches in different environments on the WSJ0 5k test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

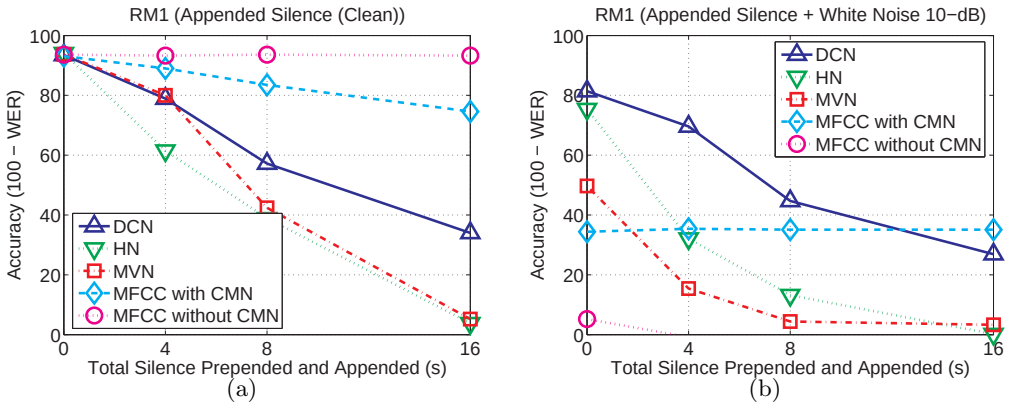


Fig. 2.11: : (a) Silence appended and prepended to the boundaries of clean speech (b) 10-dB of white Gaussian noise is added to the data used in (a)

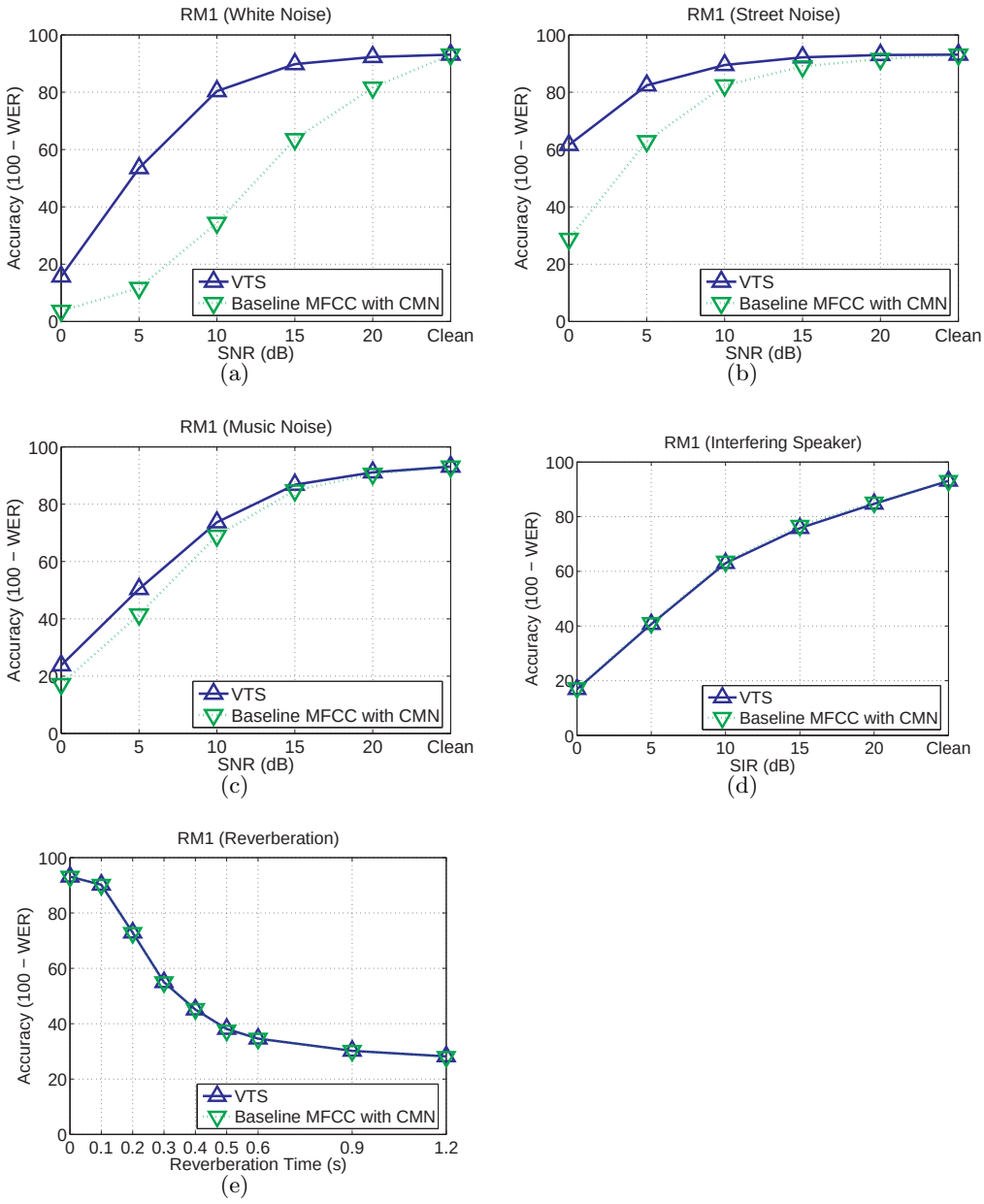


Fig. 2.12: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

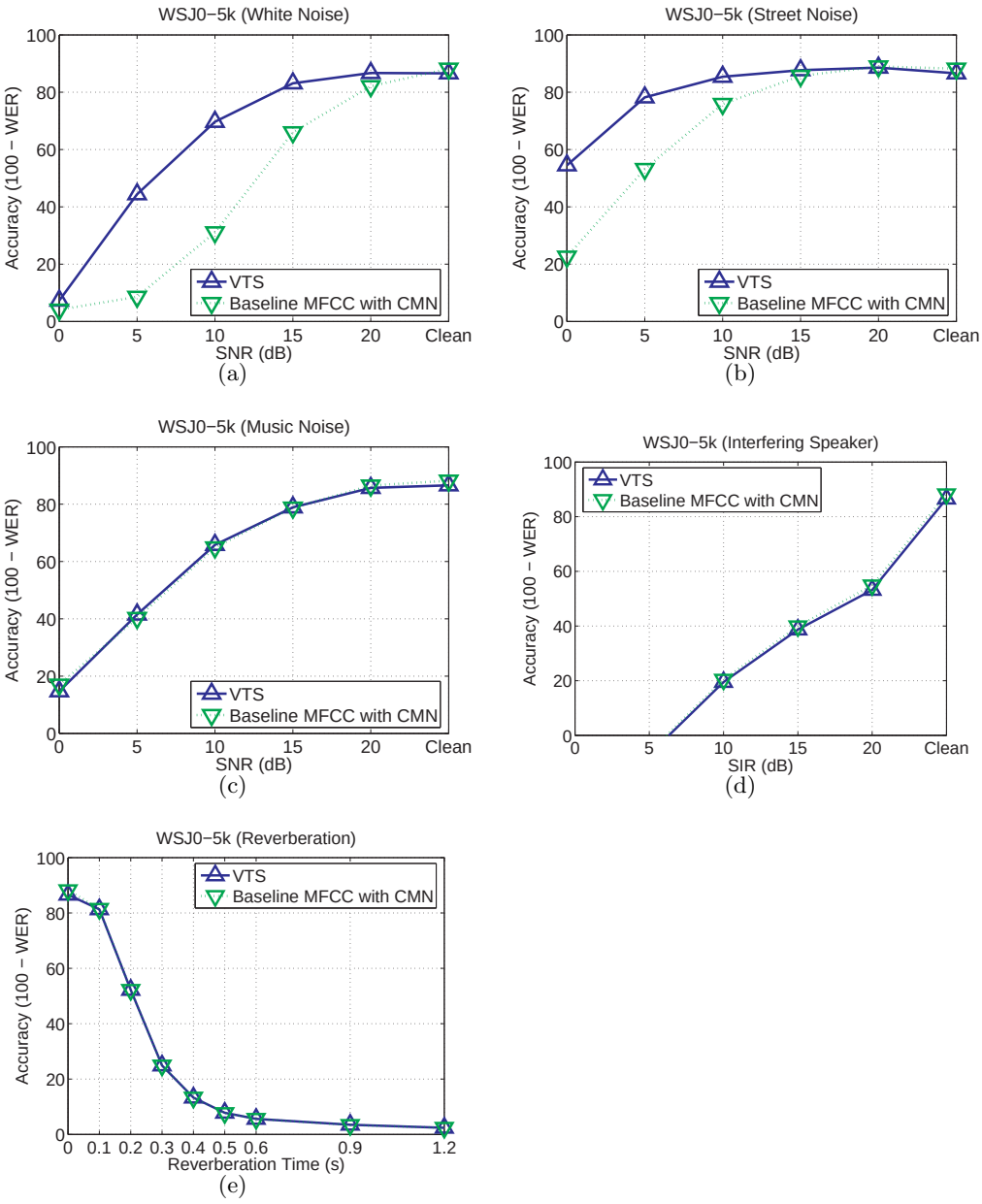


Fig. 2.13: Comparison between different normalization approaches in different environments on the RM1 test set : (a) additive white gaussian noise, (b) street noise, (c) background music, (c) interfering speaker , and (d) Reverberation

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.* [23]). 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 [24].

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.* [23]).

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 [23]. 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 throughly 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:

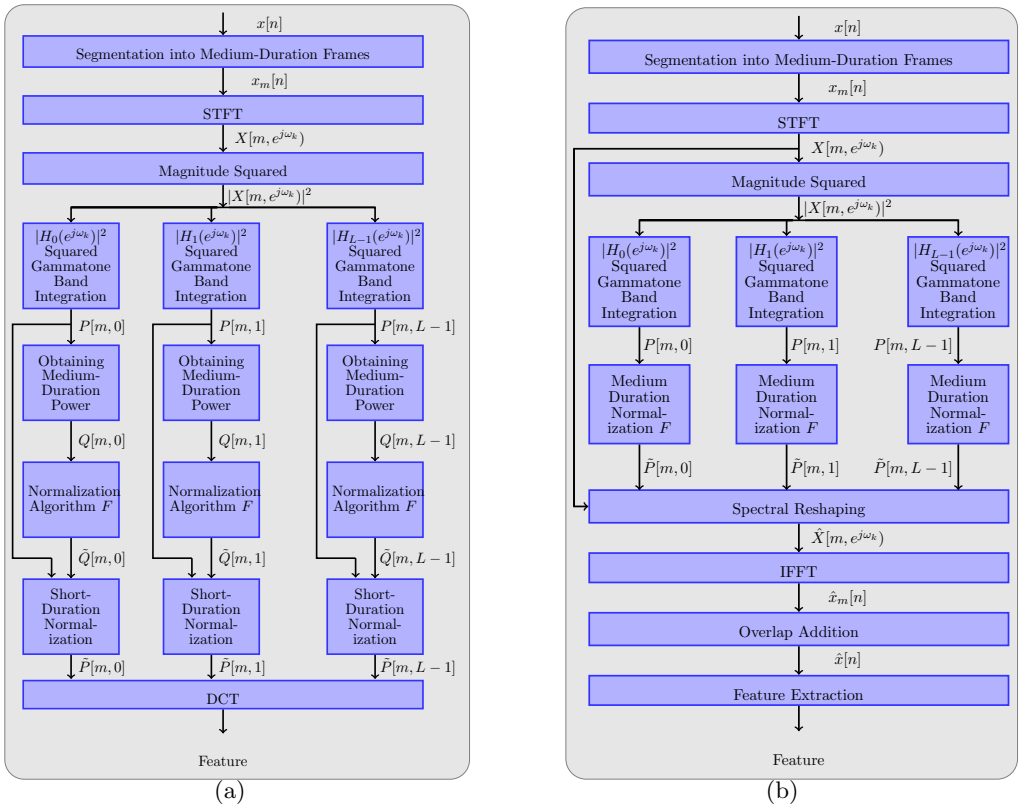


Fig. 3.1: (a) The block diagram of the Medium-duration-window Running Average (MRA) Method (b) The block diagram of the Medium-duration-window Analysis Synthesis (MAS) Method

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

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

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

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

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

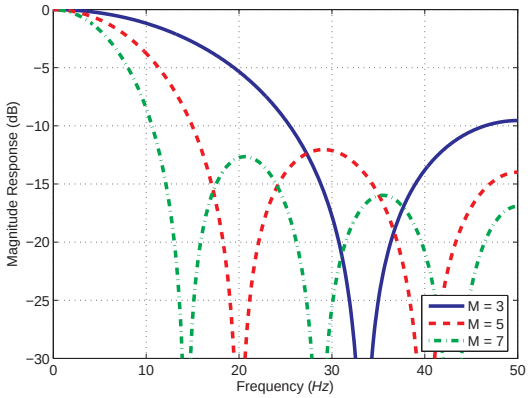


Fig. 3.2: Frequency response depending on the medium-duration parameter M

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:

$$\tilde{P}[m,l] = \frac{\tilde{Q}[m,l]}{Q[m,l]} P[m,l] \tag{3.6}$$

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

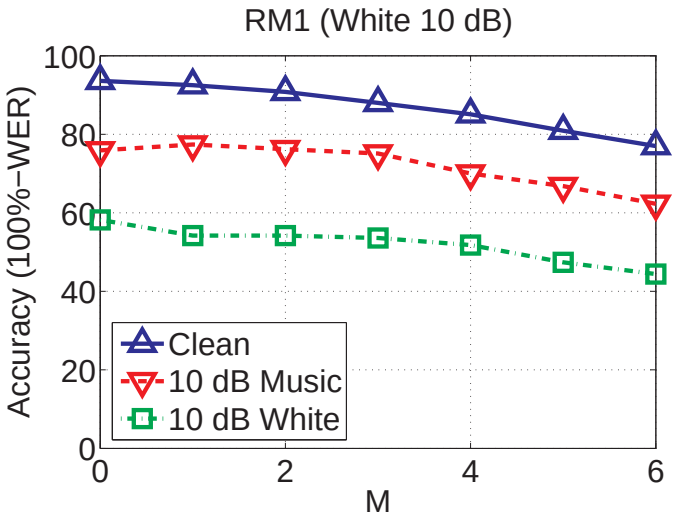


Fig. 3.3: Speech recognition accuracy depending on the medium-duration parameter M

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 9. 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

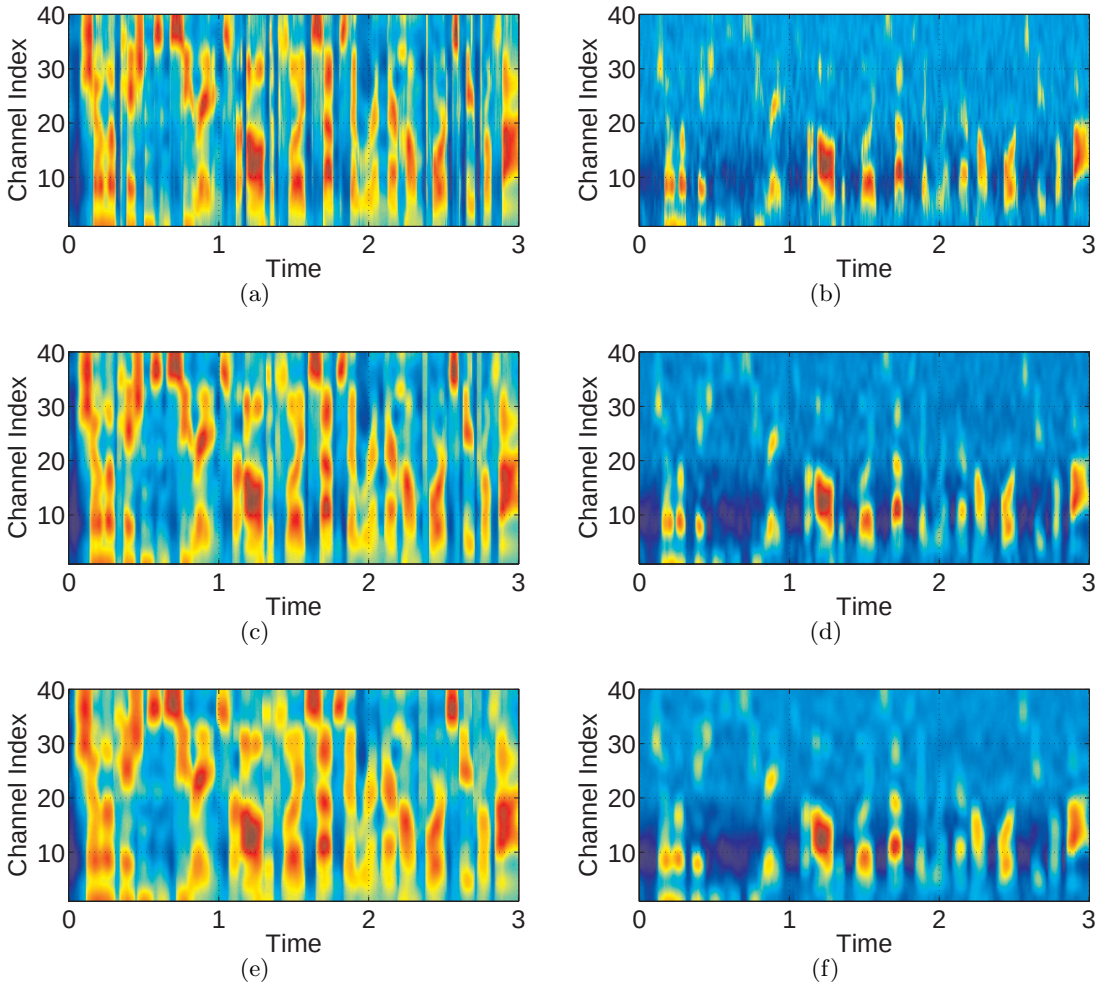


Fig. 3.4: (a) The spectrograms from clean speech with $M = 0$, (b) with $M = 2$, and (c) with $M = 4$ (d) The spectrograms from speech corrupted by 5 dB additive white noise with $M = 0$, (e) with $M = 2$, and (f) with $M = 4$

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.

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

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 (9.13), we obtain the smoothed weighting coefficient $\mu_g[k, m]$ using the following equation:

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

Finally, the reconstructed spectrum is given by:

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

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:

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

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.

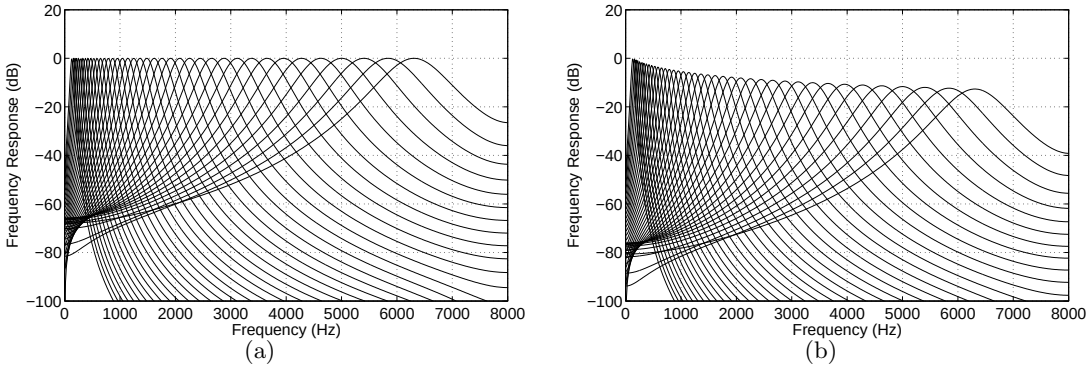


Fig. 3.5: (a) Gammatone Filterbank Frequency Response and (b) Normalized Gammatone Filterbank Frequency Response

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 [43].

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 [44] [45]. 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 [25]. 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 [46], 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-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),

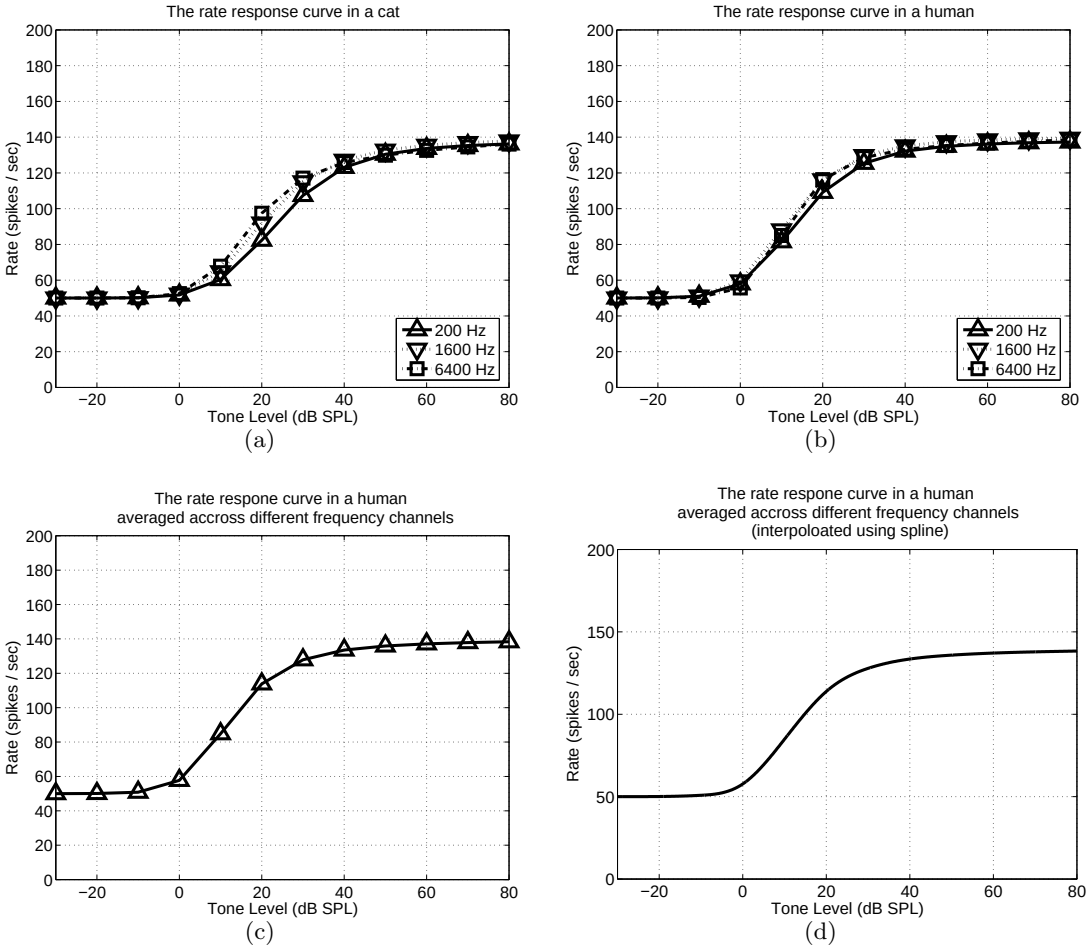


Fig. 4.1: The relation between the intensity and the rate. Simulation was done using the auditory model developed by Heinz. et al [4]: 4.1(a) shows the relation in a cat model at different frequencies. 4.1(b) shows the relation in a human model, and 4.1(c) shows the average across different channels,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

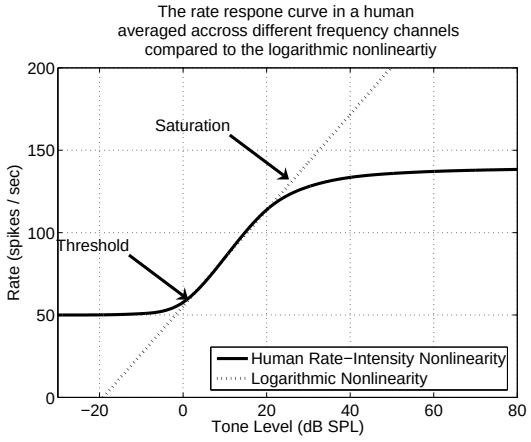


Fig. 4.2: The comparison between the intensity and rate response in the human auditory model [1] and the logarithmic curve used in MFCC. A linear transformation is applied to fit the logarithmic curve to the intensity-rate curve.

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

$$g(m,l) = \log_{10}\left(p(m,l)\right) \tag{4.1}$$

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

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

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

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

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

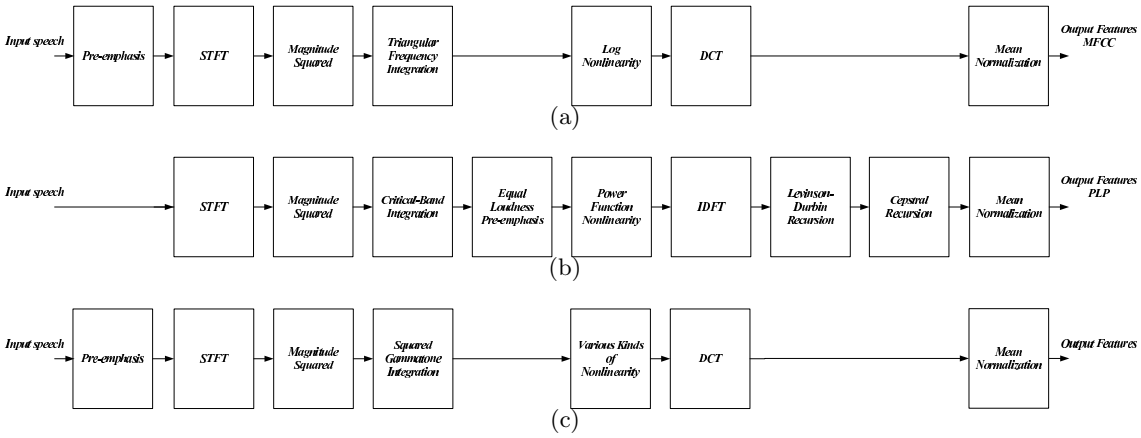


Fig. 4.3: The structure of the feature extraction system 4.3(a): MFCC, 4.3(b): PLP, and 4.3(c): General nonlinearity system

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 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.* [47]), 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 [48]. 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 [48, 49]. 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 he Room Impulse Response (RIR) software [50]. 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

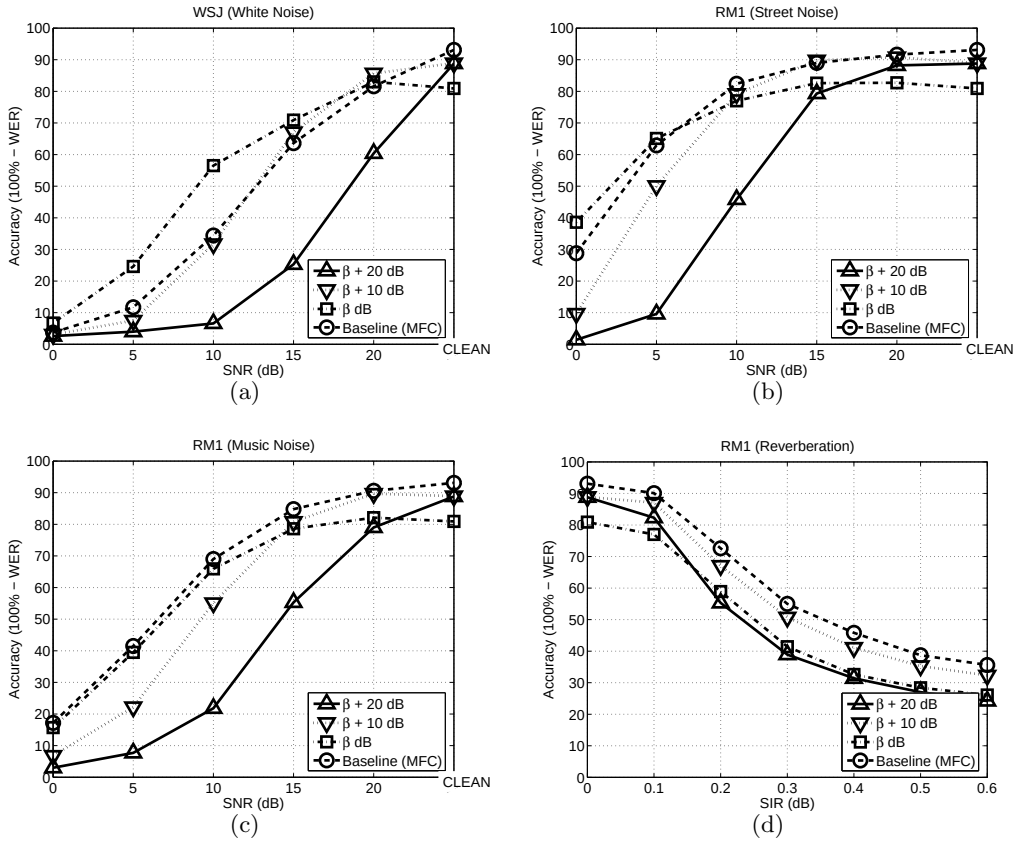


Fig. 4.4: Speech recognition accuracy obtained in different environments using the human auditory intensity-rate nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

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:

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

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.

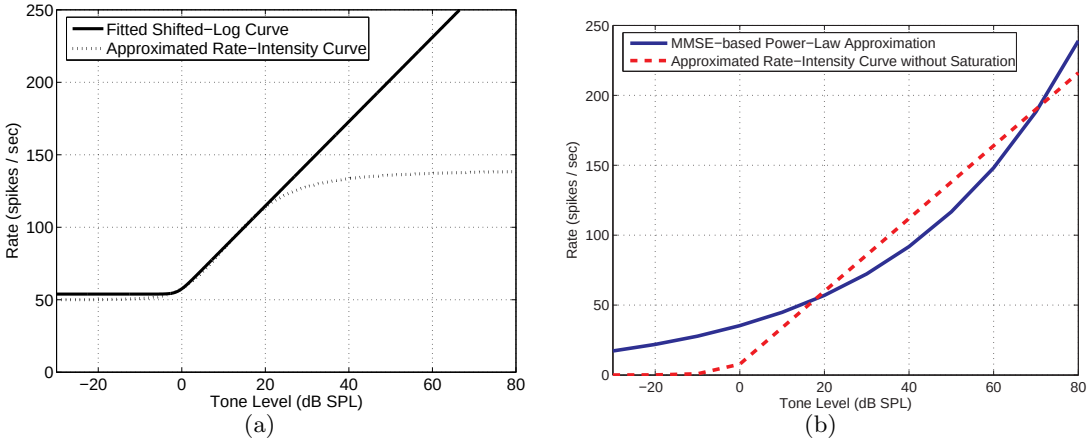


Fig. 4.5: 4.5(a) Rate-intensity curve and its stretched form in the form of shifted log 4.5(b) Power function approximation to the stretched from of the rate-intensity curve

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. **??**. This nonlinearity, which is what is used in PNCC feature extraction, is described by the equation

$$y = x^{a_0} \tag{4.5}$$

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 [49]. While our power-function nonlinearity may appear to be only a crude approximation to the physiological rate-intensity

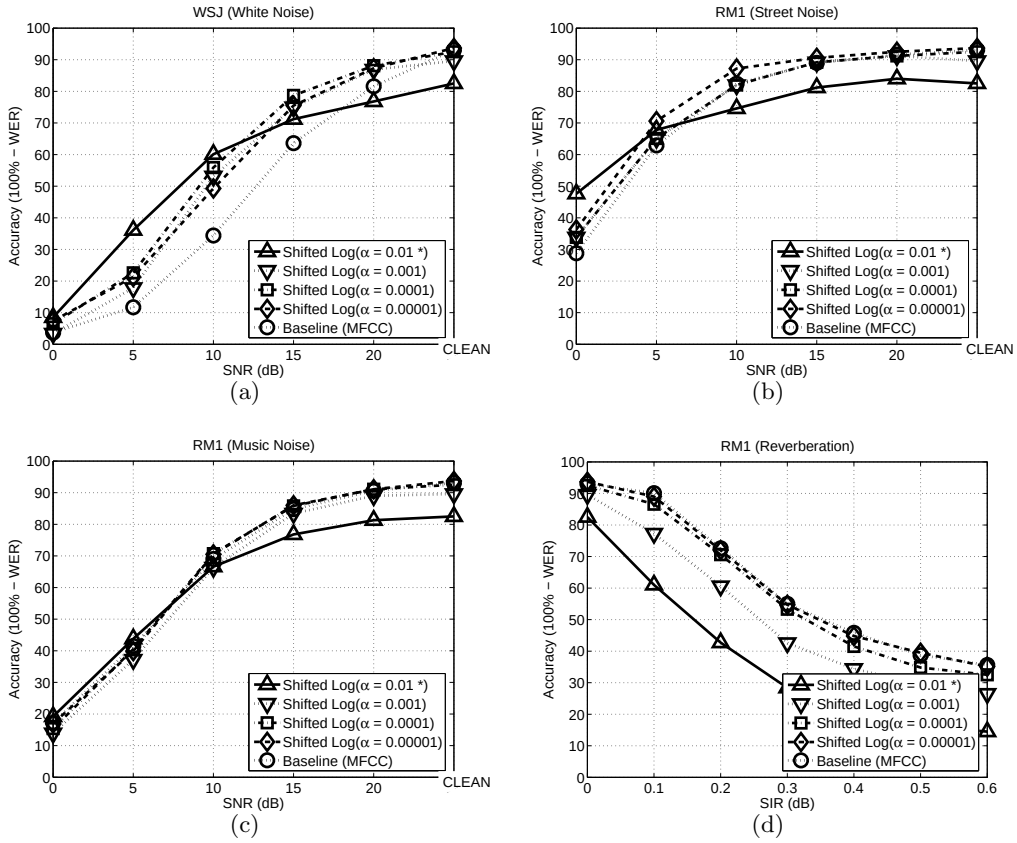


Fig. 4.6: Speech recognition accuracy obtained in different environments using the shifted log nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

function, we will show in Sec. 6.2.2 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

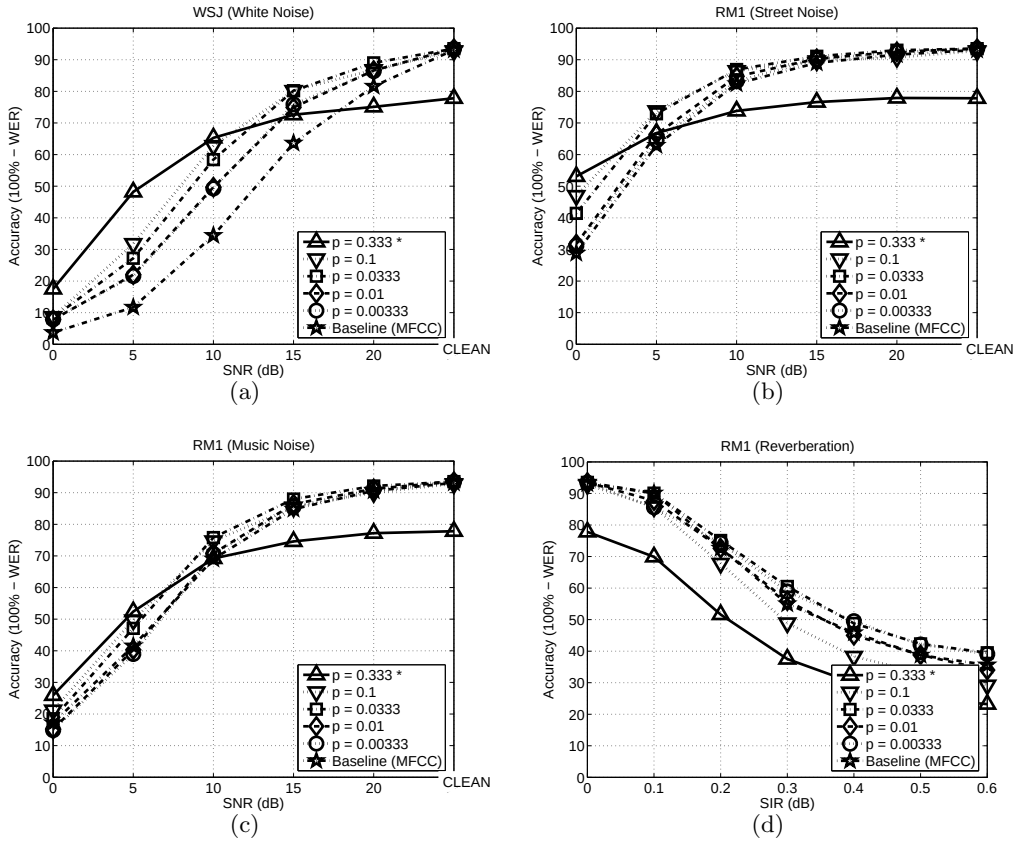


Fig. 4.7: Speech recognition accuracy obtained in different environments using the power function nonlinearity: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

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.

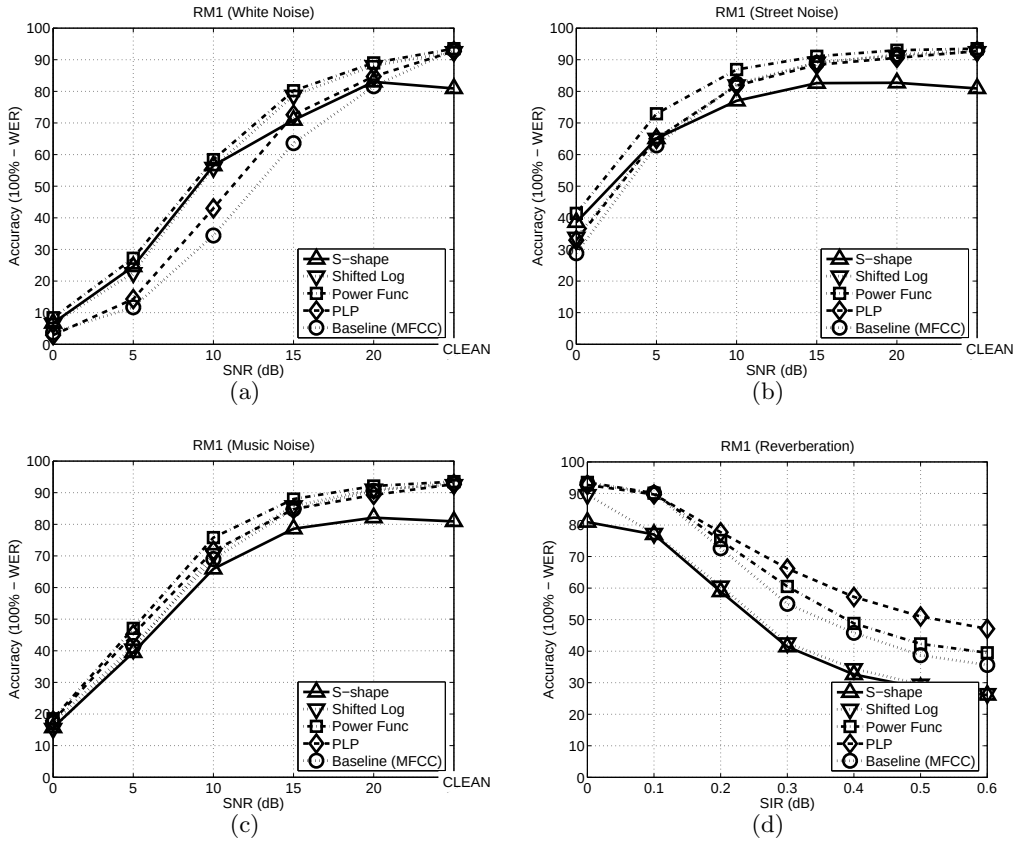


Fig. 4.8: Comparison of different nonlinearities (human rate-intensity curve, under different environments: (a) additive white gaussian noise, (b) street noise, (c) background music, (d) Reverberation

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 reveberation, PLP shows better performance, as shown in Fig. 4.8(d).

5. SMALL POWER BOOSTING ALGORITHM

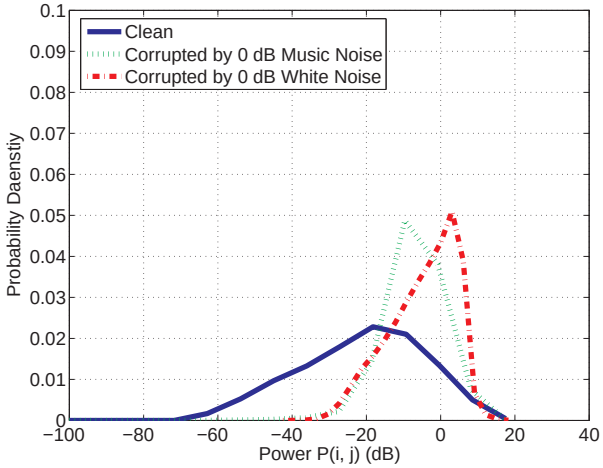
5.1 Introduction

Recent studies show that for non-stationary disturbances such as background music or background speech, algorithms based on missing features (*e.g.* [14, 51]) or auditory processing are more promising (*e.g* [47, 52, 53, 22]). Still, the improvement in non-stationary noise remains less than the improvement that is observed in stationary noise. In previous work [52] 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.

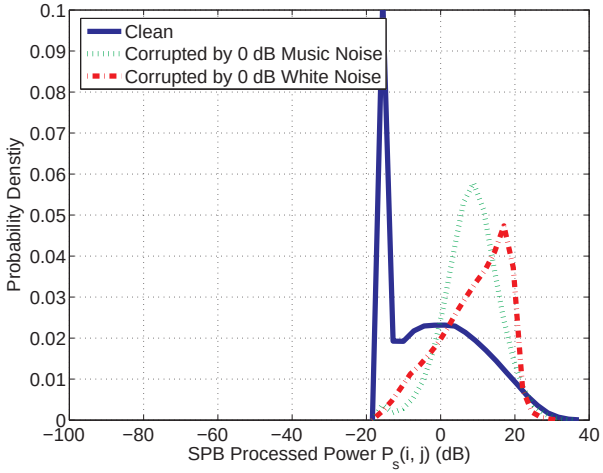
In this section, we explain the structure of the small boosting (SPB) algorithm in two different ways. In the first approach, we apply small power boosting to each time-frequency bin in the spectral domain, and then resynthesize speech (SPB-R). The resynthesized speech is fed to the feature extraction 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 [42]. 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

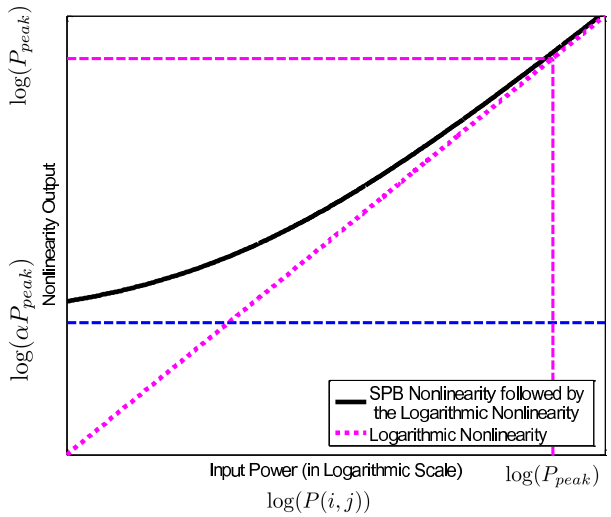


Fig. 5.2: The total nonlinearity consists of small power boosting and the subsequent logarithmic nonlinearity in the SPB algorithm

a series of weighting functions [54]. This procedure is represented by the following equation:

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

In the above equation i and j represent the channel and frame indices respectively, N is the FFT size, and $H_i(e^{jw_k})$ 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 $\omega_k = \frac{2\pi k}{N}, \quad 0 \leq k \leq N-1$.

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.* [10]), 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:

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

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:

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

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

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

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

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

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 [52], 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 [5] instead of the triangular integration with the MEL scale [20]. The advantages of gammatone-integration are described in [52], 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:

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

Using $w(i,j)$, we apply the spectral reshaping expressed in [42]:

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

where I is the total number of channels, and k is the discrete frequency index. The reconstructed spectrum is obtained from the original spectrum $X\left(e^{j\omega_k}; j\right)$ by using $\mu_g(k,j)$ in (9.14) as follows:

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

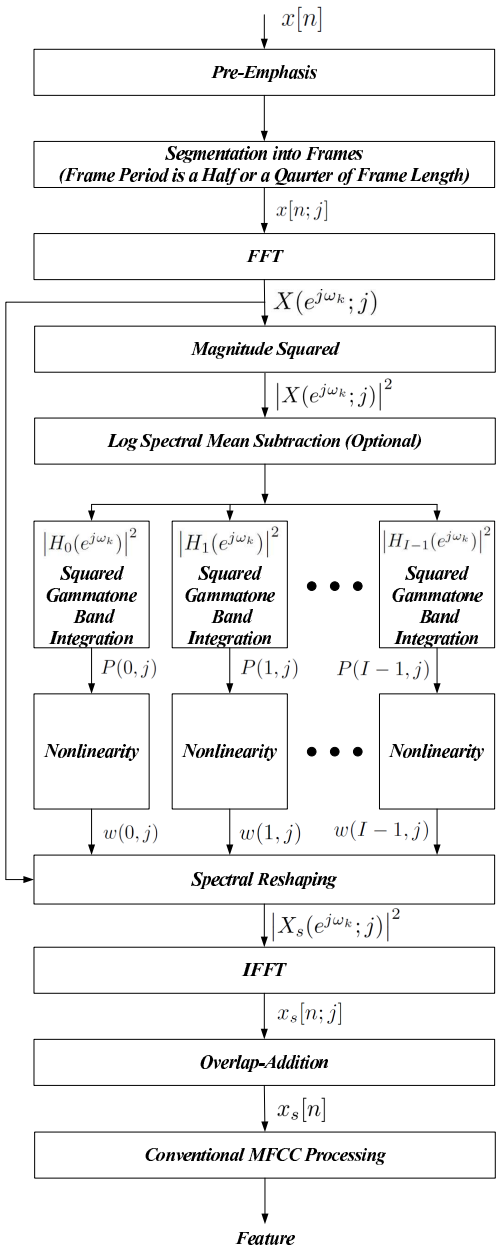


Fig. 5.3: Small power boosting algorithm which resynthesizes speech (SPB-R). Conventional MFCC processing is followed after resynthesizing the speech.

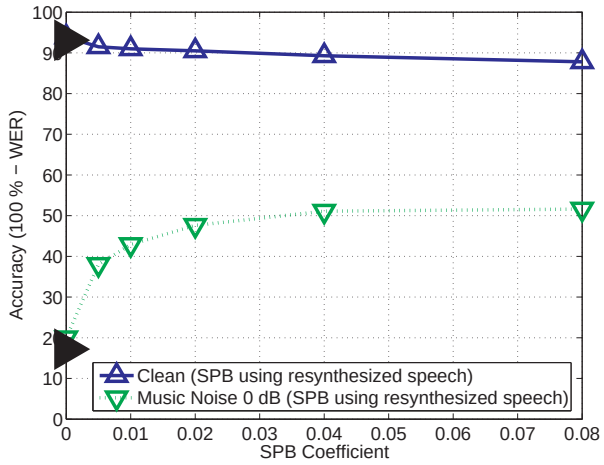


Fig. 5.4: Word error rates obtained using the SPB-R algorithm as a function of the value of the SPB Coefficient. The filled triangles at the y-axis represent the baseline MFCC performance for clean speech (upper triangle) and for additive background music noise at 0 dB SNR (lower triangle), respectively.

Speech is resynthesized using $X_s\left(e^{j\omega_k}; j\right)$ 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

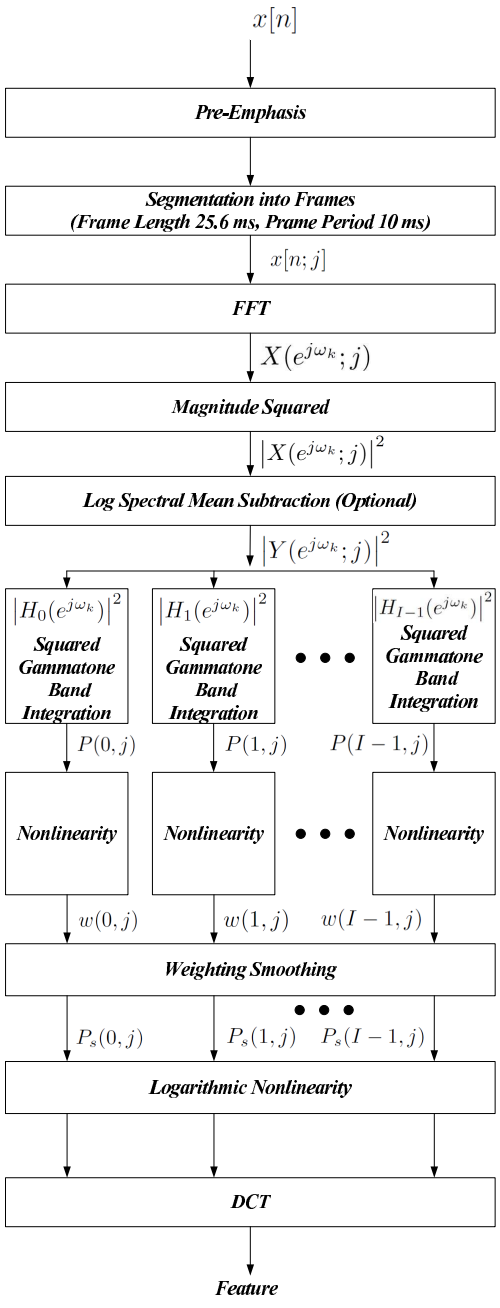


Fig. 5.5: Small power boosting algorithm with direct feature generation (SPB-D)

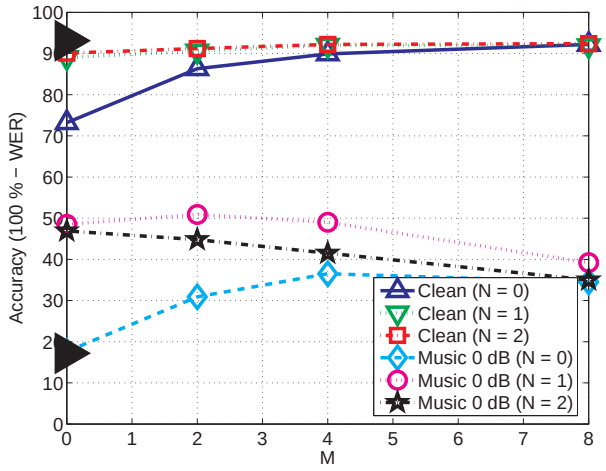


Fig. 5.6: The effects of weight smoothing on performance of the SPB-D algorithm for clean speech for speech corrupted by additive background music at 0 dB. The filled triangles at the y-axis represent the baseline MFCC performance for clean (upper triangle) and 0 dB additive background music (lower triangle)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

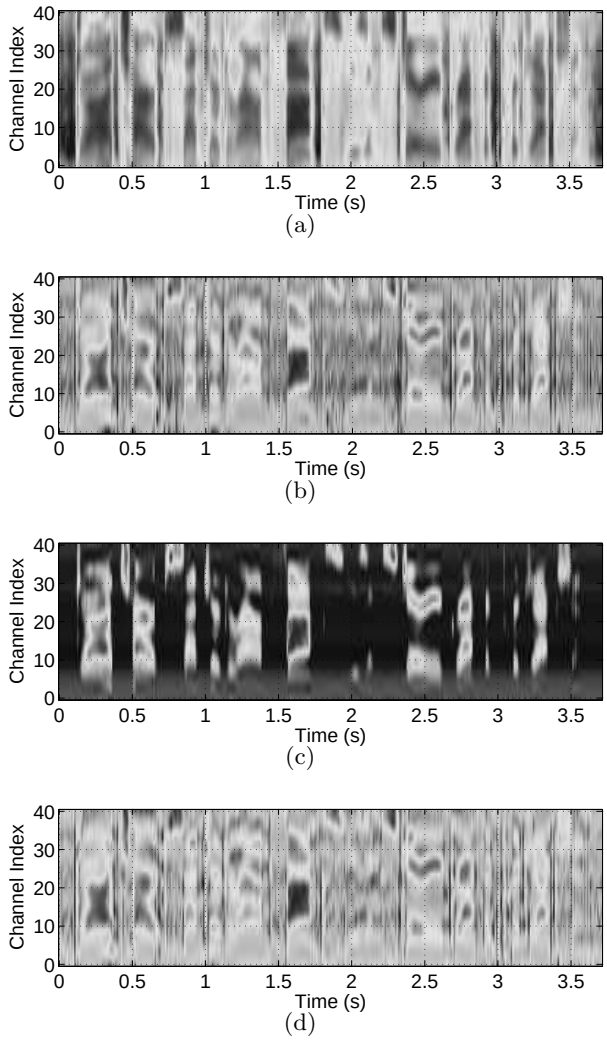


Fig. 5.7: Spectrograms obtained from a clean speech utterance using different processing: (a) conventional MFCC processing, (b) SPB-R processing, (c) SPB-D processing without any weight smoothing, and (d) SPB-D processing with weight smoothing $M = 4, N = 1$ *in* (5.9). *A value of* 0.02 *was used for 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 (9.14) and (5.8), 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:

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

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

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

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

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:

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

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 is 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.

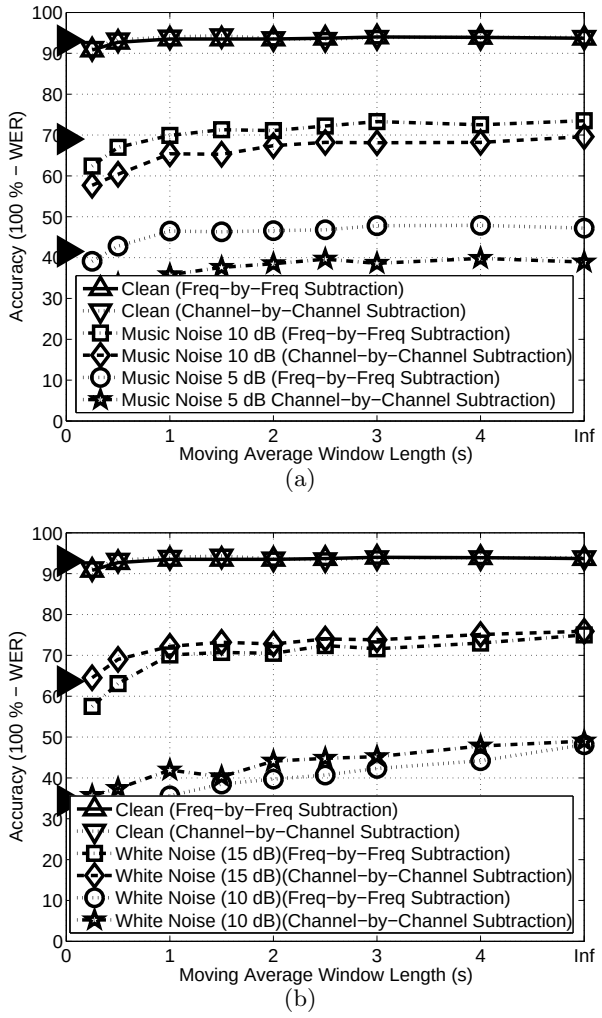


Fig. 5.8: The effect of Log Spectral Subtraction for (a) background music and (b) white noise as a function of the moving window length. The filled triangles at the y-axis represent baseline MFCC performance.

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 [42] that gammatone frequency integration is 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

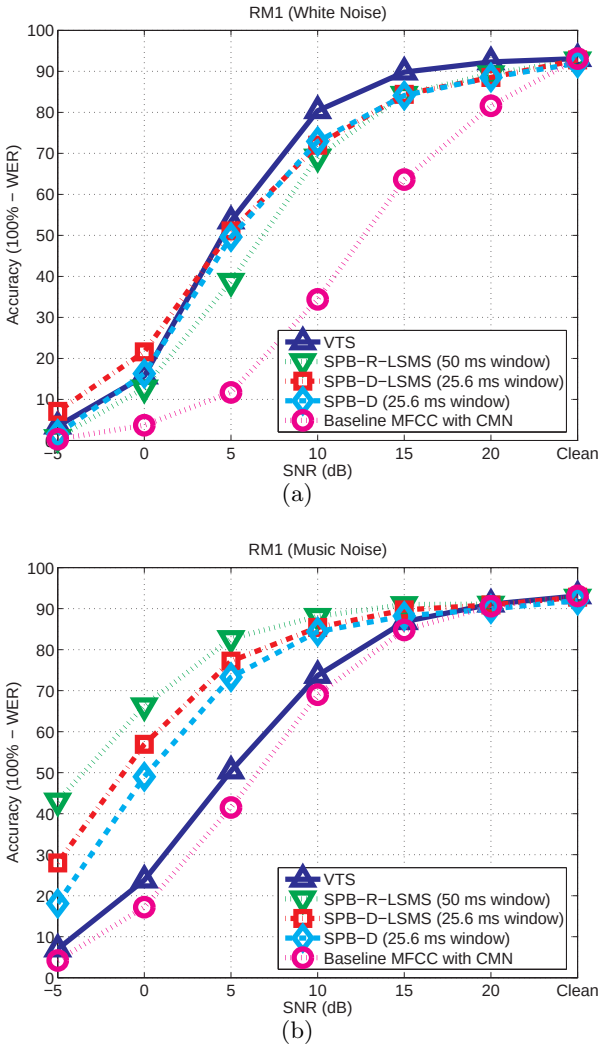


Fig. 5.9: Comparison of recognition accuracy between VTS, SPB-CW and MFCC processing: (a) additive white noise, (b) background music.

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 [9], 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.* [40]). 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.

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 [55], and it has also been used to estimate the signal-to-noise ratio [55].

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 freqeuncy 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 gamamtone 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:

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

, 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 $P[m, l]$.

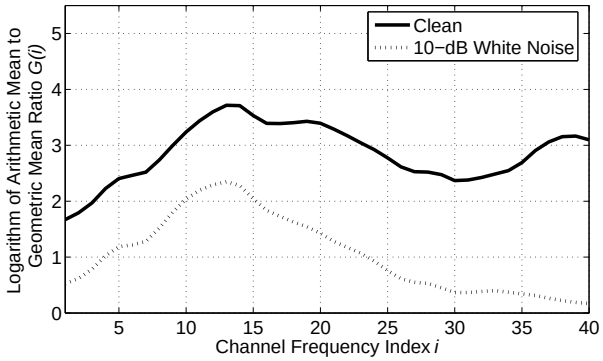


Fig. 6.1: Comparison between $G(l)$ coefficients for clean speech and speech in 10-dB white noise, using $M = 3$ in (8.3).

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:

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

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:

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

where the small positive constant ϵ is imposed to avoid evaluations of negative infinity. Fig. 6.1 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 (8.3).

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:

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

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 (6.27) and (6.4):

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

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

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

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

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

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

where $G_{cl}(l)$ is the value of $G(l)$ observed for clean speech, as shown in Fig. 6.1 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. (6.8). 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:

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

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,

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

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. **??**.

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 8 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.* [55]). Second, we apply a minimum

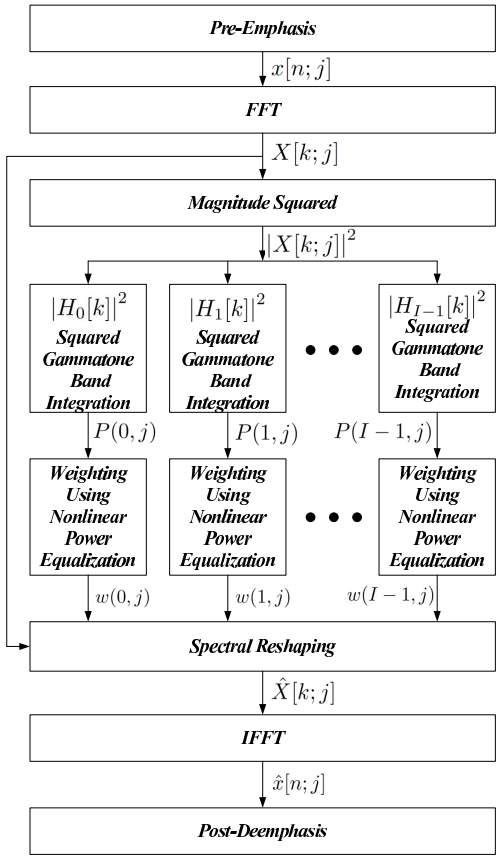


Fig. 6.2: The block diagram of the power function-based power equalization system

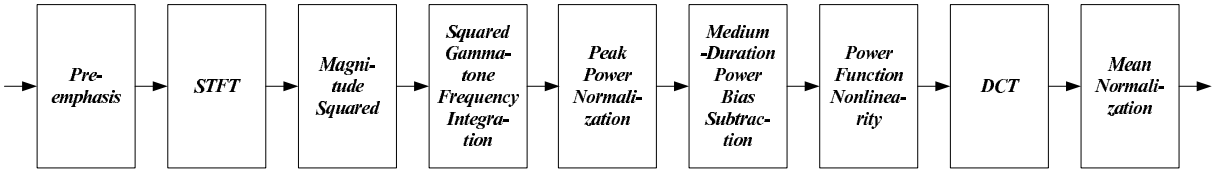


Fig. 6.3: The structure of PNCC feature extraction

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.* [34]). 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 8.

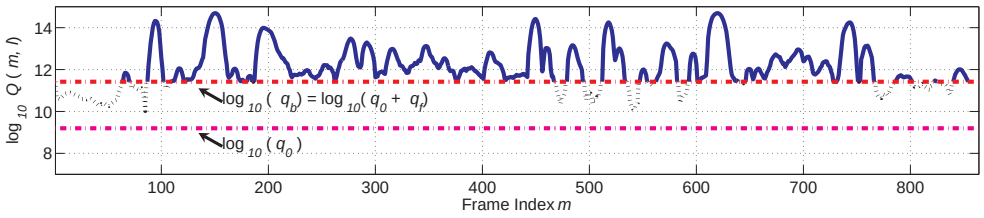


Fig. 6.4: Medium duration power $q[m,l]$ obtained from the 10^{th} channel of a speech utterance corrupted by 10-dB additive background music. The bias power level (q_b) and subtraction power level (q_0) are represented as horizontal lines. Those power levels are the actual calculated levels calculated 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:

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

where $Q[m,l]$ is the medium-duration power given by (8.3). 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:

$$\mathcal{Q}_{(t)}(l) = \Big\{ Q[m,l] : Q[m,l] > t, 1 \leq m \leq M, l' = l \Big\} \tag{6.12}$$

We use the symbol μ to represent the mean of $Q(l)$:

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

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

$$max\big\{\mathcal{Q}(l), c\big\} \quad = \quad \Big\{ \max\big\{q, c\big\} : q \in \mathcal{Q}(l) \Big\} \tag{6.14}$$

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

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

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.* [52, 34]), 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 [34].

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

$$\mathcal{R}(l|q_0) = \Big\{ R[m',l'] \quad : \quad R[m',l'] = Q[m',l'] - q_0, \\ 1 \le m' \le M, l' = l \Big\} \tag{6.16}$$

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:

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

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:

$$\mathcal{R}_{(0)}(l|q_0) = \Big\{ R[m',l'] : R[m',l'] > 0, 1 \le m' \le M, l' = l \Big\} \tag{6.18}$$

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:

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

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.

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

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):

$$\widehat{q_0} = \arg\max_{q_0}\left\{\xi\left(\max\left\{\mathcal{R}_{(q_t)}(l|q_0), q_f\right\}\right)\right\} \tag{6.21}$$

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

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

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:

$$\tilde{Q}[m,l] = \max\left\{Q[m,l] - q_0, q_f\right\} \tag{6.23}$$

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.* [34], [42]) 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:

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

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) [9] 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 [52], 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:

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

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:

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

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.

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

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.

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

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 (6.27) 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:

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

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:

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

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.

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

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:

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

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:

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

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:

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

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

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

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:

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

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 $\sim$ 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 [52] [42]. 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.

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

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

$$a_l = 1,2,...,10$$

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 $S_2[m,l]$.

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

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

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

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 $M[m,l]$.

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

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

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:

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

Now, using $w[m,l]$ in (6.36), we can normalize spectrum and can do resynthesize speech suing 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) [9]. 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 [13]. 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.

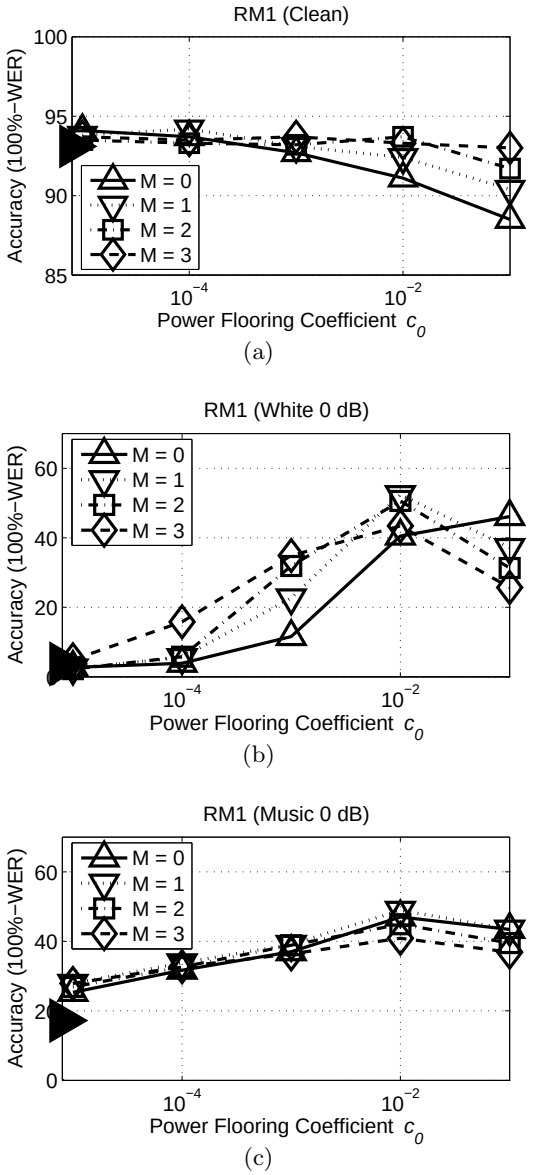


Fig. 6.5: The dependence of speech recognition accuracy obtained using PNCC on the medium-duration window factor M and the power flooring coefficient c_0. Results were obtained for (a) the clean RM1 test data (b) the RM1 test set corrupted by 0-dB white noise, and (c) the RM1 test set corrupted by 0-dB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

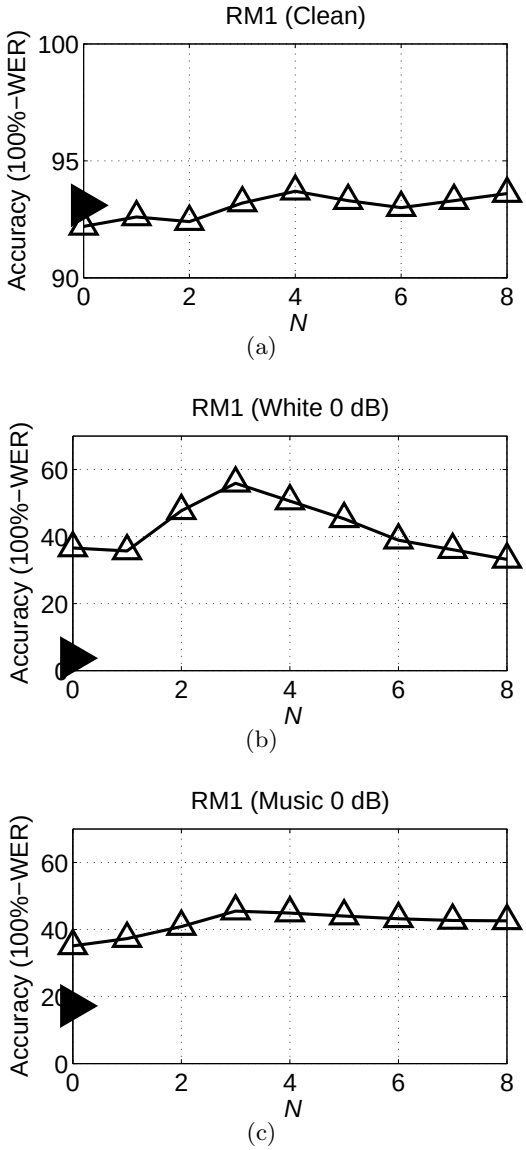


Fig. 6.6: The corresponding dependence of speech recognition accuracy on the value of the weight smoothing factor N. The filled triangle on the y-axis represents the baseline MFCC result for the same test set. For c_0 and M, we used 0.01 *and* 2 *respectively.*

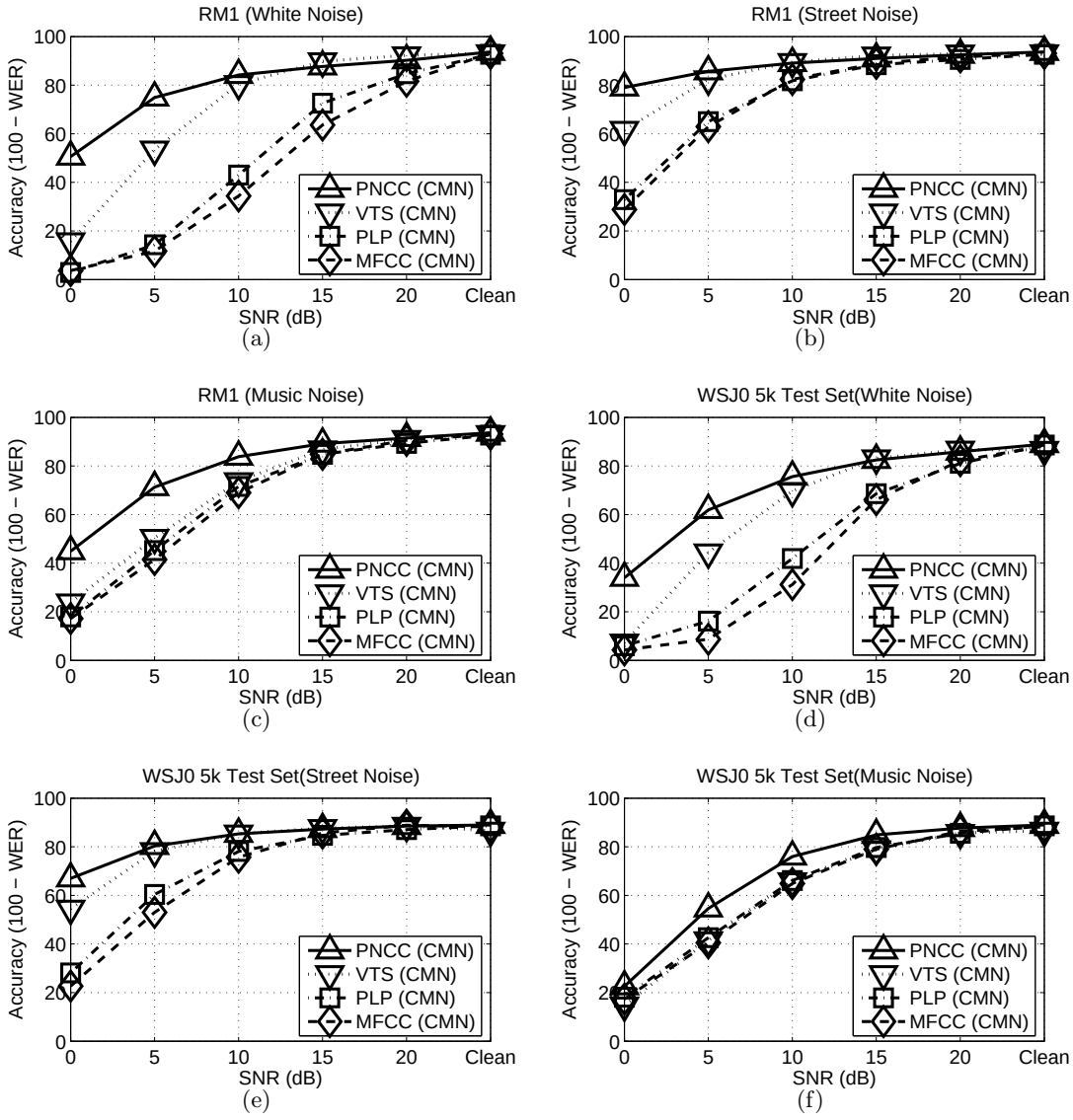


Fig. 6.7: Speech recognition accuracy obtained in different environments for different training and test sets. The RM1 database was used to produce the data in (a), (b), and (c), and the WSJ0 SI-84 training set and WSJ0 5k test set were used for the data of panels (d), (e), and (f).

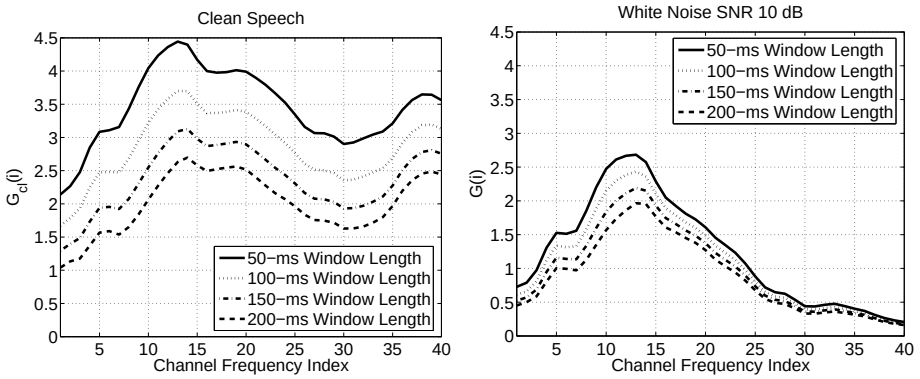


Fig. 6.8: The logarithm of the ratio of arithmetic mean to geometric mean of power from clean (a) and noise speech corrupted by 10 dB white noise (b). Data is collected from 1,600 training utterances of the resource management DB

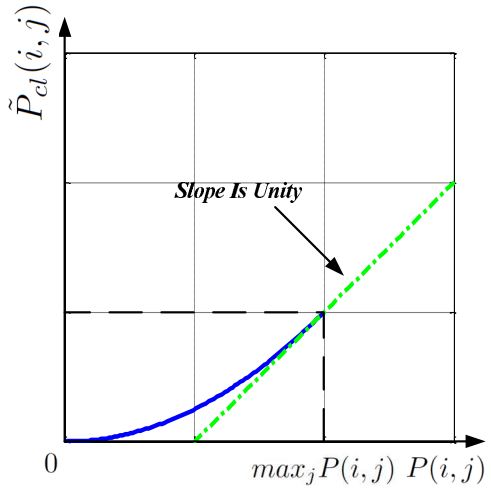


Fig. 6.9: The assumption about the relationship between $P_{cl}[m,l]$ and $P[m,l]$

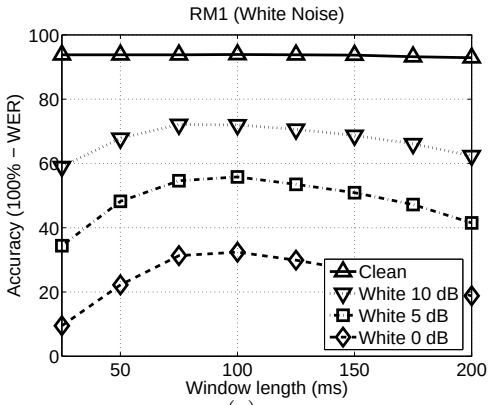


Fig. 6.10: Speech recognition accuracy as a function of the window length for the DARPA RM database corrupted by (a) white noise and (b) background music noise.

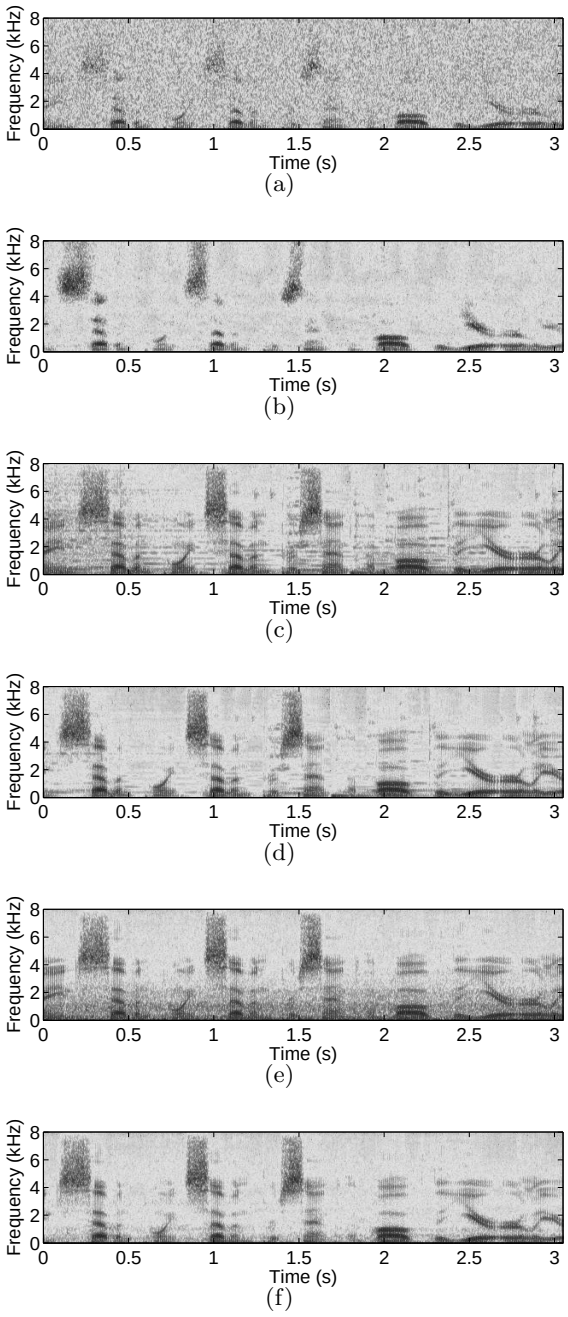


Fig. 6.11: Sample spectrograms illustrating the effects of on-line PPDN processing. (a) original speech corrupted by 0-dB additive white noise, (b) processed speech corrupted by 0-dB additive white noise (c) original speech corrupted by 10-dB additive music noise (d) processed speech corrupted by 10-dB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

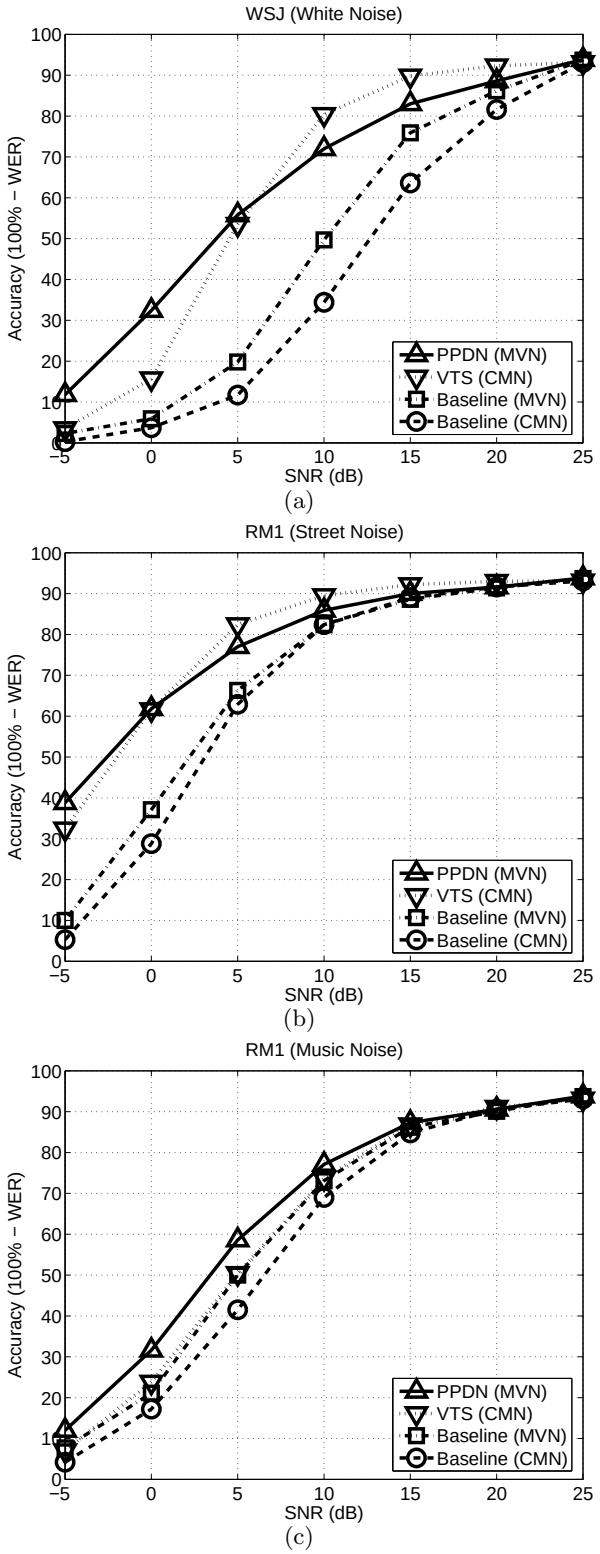


Fig. 6.12: Performance comparison for the DARPA RM database corrupted by (a) white noise, (b) street noise, and (c) music noise.

7. ONSET ENHANCEMENT

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.

It has long been believed that modulation frequency plays an important role in human hearing. For example, it is observed that the human auditory system is more sensitive to modulation frequencies less than 20 Hz (*e.g.* [30] [31]). On the other hand, very slowly changing components (*e.g.* less than 5 Hz) are usually related to noisy sources (*e.g.*[32] [33] [34]). Based on these observations, researchers have tried to utilize modulation frequency information to enhance the speech recognition performance in noisy environments. Typical approaches use highpass or bandpass filtering in either the spectral, log-spectral, or cepstral domains (*e.g.* [29]). In [2], Hirsch *etal*. investigated the effects of highpass filtering of spectral envelopes of each frequency subband. Hirsch conducted highpass filtering in the log spectral domain, using the transfer function:

$$H(z) = \frac{1 - z^{-1}}{1 - 0.7z^{-1}} \tag{7.1}$$

This first-order IIR filter can be implemented by subtracting an exponentially weighted moving average from the current log spectral value. For robust speech recognition, the other common difficulty is reverberation. Many hearing scientists believe that human speech perception in reverberation is enabled by the "precedence effect", which refers to the emphasis that appears to be given to the first-arriving wave-front of a complex signal in sound localization and possibly speech perception (*e.e.* [56]). To detect the first wave-front, we can either measure the envelope of the signal or energy in the frame (*e.g.* [57] [58]).

In this paper, we introduce an approach we refer to as SSF processing, which represents Suppression of Slowly-varying components and the Falling edge of the power envelope. This processing mimics aspects of both the precedence effect and modulation spectrum analysis.

SSF processing operates on frequency weighted power coefficients as they evolve over time, as described below. The DC-bias term is first removed in each frequency band by subtracting an exponentially-weighted moving average. When the instantaneous power in a given frequency channel is smaller than this average, the power is suppressed, either by scaling by a small constant, or by replacement by the scaled moving average. The first approach results in better sound quality for non-reverberated speech, but the latter results in better speech recognition accuracy in reverberant environments. SSF processing is normally applied to both training and testing data in speech recognition applications.

In speech signal analysis, we normally use a short-duration window with duration between 20 and 30 ms. With the SSF algorithm, we observe that windows longer than this length are more appropriate for estimating or compensating for noise components, which is consistent with our observations in previous work (*e.g.* [52][42][32]). Nevertheless, even if we use a longer-duration window for noise estimation, we must use a short-duration window for speech feature extraction. After performing frequency-domain processing we use an IFFT and the overlap-add method (OLA) to re-synthesize speech, as in [33]. Feature extraction and subsequent speech recognition can be performed on the re-synthesized speech. We will call this general approach medium-duration analysis and synthesis approach (MAS).

7.1 Structure of the SSF algorithm

Figure 7.1 shows the structure of the SSF algorithm. The input speech signal is pre-emphasized and then multiplied by a medium-duration Hamming window as in [33]. This signal is represented by $x_m[n]$ in Fig. 7.1 where m denotes the frame index. We use a 50-ms window and 10 ms between frames. After windowing, the FFT is computed and integrated over frequency using gammatone weighting functions to obtain the power $P[m,l]$ in the m^{th} frame and l^{th} frequency band as shown below:

$$P[m,l] = \sum_{k=0}^{N-1} |X[m,e^{j\omega_k})H_l(e^{j\omega_k})|^2, \quad 0 \le l \le L-1 \tag{7.2}$$

where k is a dummy variable representing the discrete frequency index, and N is the FFT size. The discrete frequency is $\omega_k = \frac{2\pi k}{N}$. Since we are using a 50-ms window, for 16-kHz audio samples N is 1024. $H_l(e^{j\omega_k})$ is the spectrum of the gammatone filter bank for the l^{th}

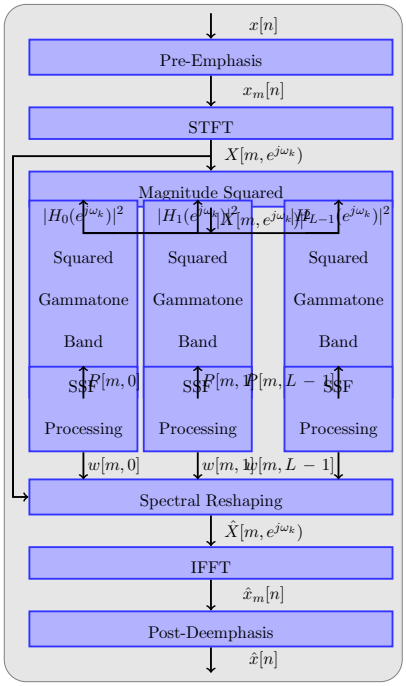


Fig. 7.1: The block diagram of the SSF processing system

channel evaluated at frequency index k, and $X[m, e^{j\omega_k})$ is the short-time spectrum of the speech signal for the m^{th} frame, where $L = 40$ is the total number of gammatone channels. After the SSF processing described below, we perform spectral reshaping and compute the IFFT using OLA to obtain enhanced speech.

7.2 SSF Type-I and SSF Type-II Processing

In SSF processing, we first obtain lowpassed power $M[m, l]$ from each channel:

$$M[m, l] = \lambda M[m-1, l] + (1-\lambda)P[m, l] \tag{7.3}$$

where λ is a forgetting factor that is adjusted for the bandwidth of the lowpass filter. The processed power is obtained by the following equation:

$$P_1[m, l] = \max\left(P[m, l] - M[m, l], c_0 P[m, l]\right) \tag{7.4}$$

where c_0 is a small fixed coefficient to prevent $P[m, l]$ from becoming negative. In our experiments we find that $c_0 = 0.01$ is appropriate for suppression purposes. As is obvious

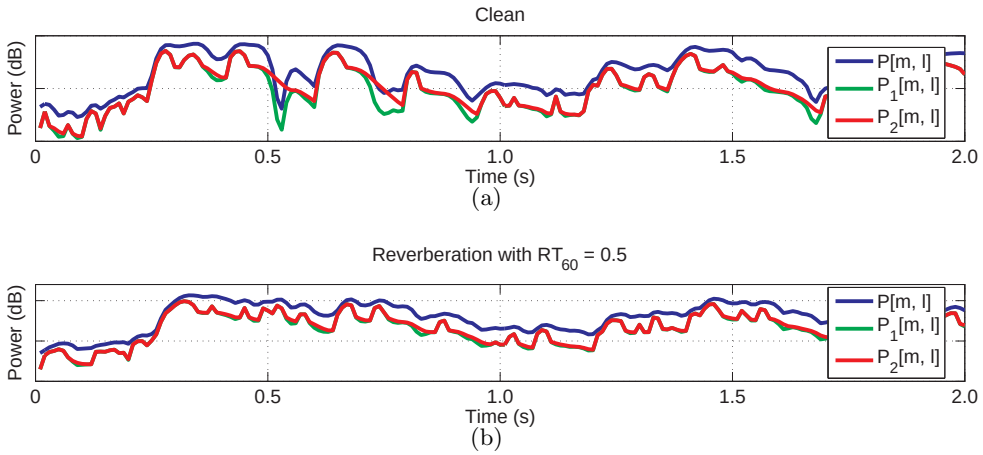


Fig. 7.2: Power contour $P[m,l]$, $P_1[m,l]$ (processed by SSF Type-I processing), and $P_2[m,l]$ (processed by SSF Type-II processing) for the 10-th channel in clean environment (a) and in the reverberant environment (b).

from (7.4), $P_1[m,l]$ is intrinsically a highpass filter signal, since the lowpassed power $M[m,l]$ is subtracted from the original signal power $P[m,l]$. From (7.4), we observe that if power $P[m,l]$ is larger than $M[m,l]+c_0P_1[m,l]$ then, $P_1[m,l]$ is the highpass filter output. However, if $P[m,l]$ is smaller than the latter, the power is suppressed. These operations have the effect of suppressing the falling edge of the power contour. We call processing using (7.4) SSF Type-I.

A similar approach uses the following equation instead of (7.4):

$$P_2[m,l] = \max\left(P[m,l] - M[m,l], c_0 M[m,l]\right) \tag{7.5}$$

We call this processing SSF Type-II.

The only difference between (7.4) and (7.5) is one term, but as shown in Fig 7.3 and 7.4, this term has a major impact on recognition accuracy in reverberant environments. We also note that using SSF Type-I processing, if $0.2 \leq \lambda \leq 0.4$, substantial improvements are observed for clean speech compared to baseline processing. In the power contour of Fig. 7.2, we observe that if we use SSF Type-II, the falling edge is smoothed (since $M[m,l]$ is basically a lowpass signal), which significantly reduces spectral distortion between clean and reverberant environments.

Fig. 7.3 shows the dependence of performance dependence on the forgetting factor λ and the window length. For additive noise, a window length of 75 or 100 ms provided the best

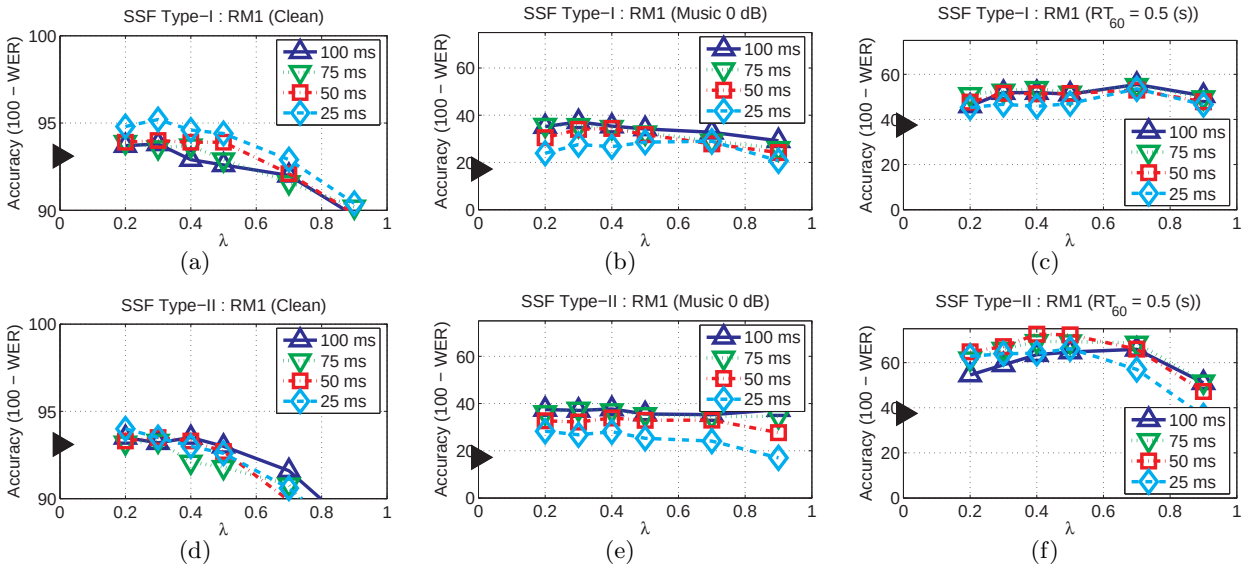


Fig. 7.3: The dependence of speech recognition accuracy on the forgetting factor λ and the window length. In (a), (b), and (c), we used (7.4) *for normalization. In (d), (e), and (f), we used* (7.5) *for normalization. The filled triangles along the vertical axis represent the baseline MFCC performance in the same environment.*

performance. However, for reverberation, 50 ms provided the best performance. Thus we use $\lambda = 0.4$ and a window length of 50 ms.

7.3 *Spectral reshaping*

After obtaining processed power $\tilde{P}[m,l]$ (which is either $P_1[m,l]$ in (7.4) or $P_2[m,l]$ (7.5)), we obtain a processed spectrum $\tilde{X}[m,e^{j\omega_k}]$. To achieve this goal, we use a similar spectral reshaping approach as in [33] and [42]. Assuming that the phases of the original and the processed spectra are identical, we modify only the magnitude spectrum.

First, for each time-frequency bin, we obtain the weighting coefficient $w[m,l]$ as a ratio of the processed power $\tilde{P}[m,l]$ to $P[m,l]$.

$$w[m,l] = \frac{\tilde{P}[m,l]}{P[m,l]}, \qquad 0 \le l \le L-1 \tag{7.6}$$

Each of these channels is associated with H_l, the frequency response of one of a set of gammatone filters with center frequencies distributed according to the Equivalent Rectangular

Bandwidth (ERB) scale [5]. The final spectral weighting $\mu[m,k]$is obtained using the above weight $w[m,l]$

$$\mu[m,k] = \frac{\sum_{l=0}^{L-1} w[m,l]\left|H_l\left(e^{j\omega_k}\right)\right|}{\sum_{l=0}^{L-1}\left|H_l\left(e^{j\omega_k}\right)\right|}, \quad 0 \le k \le N/2-1, 0 \le l \le L-1 \tag{7.7}$$

After obtaining $\mu[m,k]$ for the lower half frequency region $0 \le k \le N/2$, we can obtain the upper half from the symmetric characteristic:

$$\mu[m,k] = \mu[m,N-k], \qquad N/2 \le k \le N-1 \tag{7.8}$$

Using $\mu[m,k]$, the reconstructed spectrum is obtained by:

$$\tilde{X}[m,e^{j\omega_k}) = \mu[m,k]X[m,e^{j\omega_k}), \;\; 0 \le k \le N-1 \tag{7.9}$$

The enhanced speech $\hat{x}[n]$is re-synthesized using the IFFT and OverLap Addition (OLA) method as described above.

7.4 Experimental results

In this section we describe experimental results obtained on the DARPA Resource Management (RM) database using the SSF algorithm. For quantitative evaluation of SSF we used 1,600 utterances from the DARPA Resource Management (RM) database for training and 600 utterances for testing. We used `SphinxTrain 1.0` for training the acoustic models, and `Sphinx 3.8` for decoding. For feature extraction we used `sphinx_fe` which is included in `sphinxbase 0.4.1`. Even though SSF is developed for reverberant environment, we also conducted experiments in additive noise as well. In Fig. 7.4(a), we used test utterances corrupted by additive white Gaussian noise, and in Fig. 7.4(b), we used test utterances corrupted by musical segments of the DARPA Hub 4 Broadcast News database.

We prefer to characterize improvement as amount by which curves depicting WER as a function of SNR shift laterally when processing is applied. We refer to this statistic as the "threshold shift". As shown in these figures, SSF provides 8-dB threshold shifts for white noise and 3.5-dB shifts for background music. Note that obtaining improvements for background music is not easy. For comparison, we also obtained similar results using

the state-of-the-art noise compensation algorithm vector Taylor series (VTS) [9]. We also conducted experiments using an open source RASTA-PLP implementation [27]. For white noise, VTS and SSF show almost the same performance, but for background music, SSF provides significantly better performance. In additive noise, both SSF Type-I and SSF Type-II provide almost the same performance. For clean utterances, SSF Type-I performs slightly better than SSF Type-II.

To simulate the effects of room reverberation, we used the software package Room Impulse Response (RIR) [50]. We assumed a room of dimensions of 5 x 4 x 3m , a distance between the microphone and the speaker of 2 m, with the microphones located at the center of the room. In reverberant environments, as shown in Fig. 7.4(c), SSF Type-II shows the best performance by a very large margin. SSF Type-I shows the next performance, but the performance difference between SSF Type-I and SSF-Type-II is large. On the contrary, VTS does not provide any performance improvement, and PLP-RASTA provides worse performance than MFCC.

7.5 Conclusions

In this paper, we present a new algorithm that is especially robust with respect to reverberation. Motivated by modulation frequency concept and the precedence effect, we apply a first-order high-pass filtering to power coefficients. The falling edges of power contours are suppressed in two different ways. We observe that using the lowpassed signal for the falling edge is especially helpful for reducing spectral distortion for reverberant environments. Experimental results show that this approach is more effective than previous algorithms in reverberant environments.

7.6 Open source MATLAB code

MATLAB code for the SSF algorithm may be found at `http://www.cs.cmu.edu/~robust/archive/algorithms/SSF_IS2010/`. This code was used to obtain the results in Section 7.4.

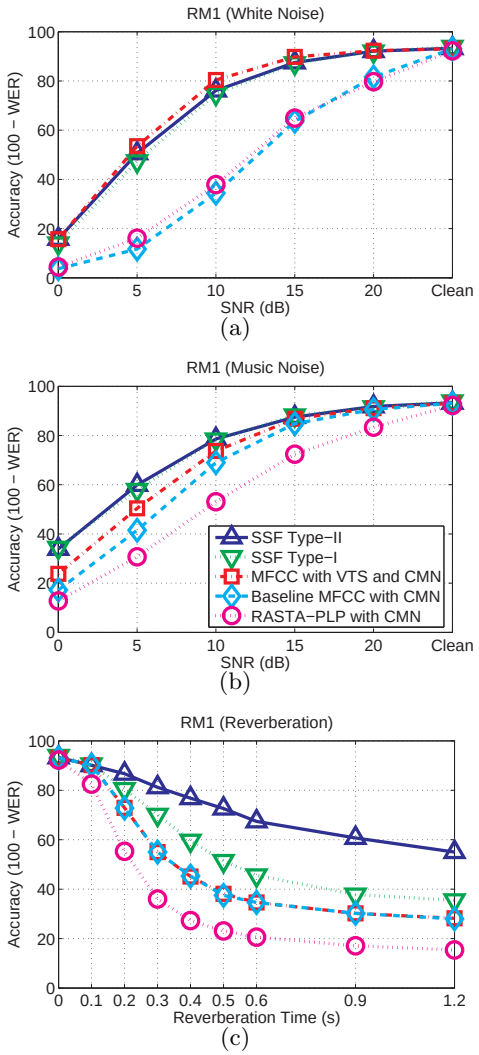


Fig. 7.4: Speech recognition accuracy using different algorithms (a) for white noise (b) for musical noise, and (c) under reverberant environments.

8. 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. In recent decades following the introduction of hidden Markov models (*e.g.* [59]) and statistical language models (*e.g.*[**?**]), the performance of speech recognition systems in benign acoustical environments has dramatically improved. Nevertheless, most speech recognition systems remain sensitive to the nature of the acoustical environments within which they are deployed, and their performance deteriorates sharply in the presence of sources of degradation such as additive noise, linear channel distortion, and reverberation.

One of the most challenging contemporary problems is that recognition accuracy degrades significantly if the test environment is different from the training environment and/or if the acoustical environment includes disturbances such as additive noise, channel distortion, speaker differences, reverberation, and so on. Over the years dozens if not hundreds of algorithms have been introduced to address this problem. Many of these conventional noise compensation algorithms have provided substantial improvement in accuracy for recognizing speech in the presence of qusai-stationary noise (*e.g.* [8, 9, 7, 39, 11]). Unfortunately these same algorithms frequently do not provide significant improvements in more difficult environments with transitory disturbances such as a single interfering speaker or background music (*e.g.* [40]).

Virtually all of the current systems developed for automatic speech recognition, speaker identification, and related tasks are based on variants of one of two types of features: *mel frequency cepstral coefficients* (MFCC) [60] or *perceptual linear prediction* (PLP) coefficients [22]. In this paper we describe the development of a third type of feature set for speech recognition which we refer to as *power-normalized cepstral coefficients* (PNCC). As we will

show, PNCC features provide superior recognition accuracy over a broad range of conditions of noise and reverberation using features that are computable in real time using "online" algorithms, and with a computational complexity that is comparable to that of traditional MFCC and PLP features.

In the subsequent subsections of this Introduction we discuss the broader motivations and overall structure of PNCC processing. We specify the key elements of the processing in some detail in Sec. 8.1. In Sec. 8.2 we provide experimental findings that are used to specify the some of the free parameters of the model, as well as evaluations of the recognition accuracy provided by PNCC processing under a variety of conditions. We compare the computational complexity of the MFCC, PLP, and PNCC feature extraction algorithms in Sec. 8.3 and we summarize our results in the final section.

8.0.1 Broader motivation for the PNCC algorithm

The development of PNCC feature extraction was motivated by a desire to obtain a set of practical features for speech recognition that are more robust with respect to acoustical variability in their native form, without loss of performance when the speech signal is undistorted, and with a degree of computational complexity that is comparable to that of MFCC and PLP coefficients. While many of the attributes of PNCC processing have been strongly influenced by consideration of various attributes of human auditory processing, we have favored approaches that provide pragmatic gains in robustness at modest computational cost over approaches that are more faithful to auditory physiology in developing the specific processing that is performed.

Some of the innovations of the PNCC processing that we consider to be the most important include:

- The replacement of the log nonlinearity in MFCC processing by a power-law nonlinearity that is carefully chosen to approximate the nonlinear relation between signal intensity and auditory-nerve firing rate. We believe that this nonlinearity provides superior robustness by suppressing small signals and their variability, as discussed in Sec. 8.1.7.

- The use of "medium-time" frames with a duration of 70-120 ms to analyze the param-

eters characterizing environmental degradation, in combination with the traditional short-time Fourier analysis with frames of 20-30 ms used in conventional speech recognition systems. We believe that this approach enables us to estimate environmental degradation more accurately while maintaining the ability to respond to rapidly-changing speech signals, as discussed in Sec. 8.1.2.

- The use of a form of "asymmetric nonlinear lowpass filtering" to estimate the level of the acoustical background noise for each time frame and frequency bin. We believe that this approach enables us to remove background components easily without needing to deal with many of the artifacts associated with over-correction in techniques such as spectral subtraction [10], as discussed in Sec. 8.1.3.

- The development of computationally-efficient realizations of the algorithms above that support "online" real time processing.

8.0.2 Structure of the PNCC algorithm

Fig. 8.1 compares the structure of conventional MFCC processing [60], PLP processing [22, 3], and the new PNCC approach which we introduce in this paper. As was noted above, the major innovations of PNCC processing include the redesigned nonlinear rate-intensity, along with the series of processing elements to suppress the effects of background acoustical activity based on medium-time analysis.

As can be seen from Fig. 8.1, the initial processing stages of PNCC processing are quite similar to the corresponding stages of MFCC and PLP analysis, except that the frequency analysis is performed using Gammatone filters [54]. This is followed by the series of nonlinear time-varying operations that are performed using the longer-duration temporal analysis that accomplish noise subtraction as well as a degree of robustness with respect to reverberation. The final stages of processing are also similar to MFCC and PLP processing, with the exception of the carefully-chosen power-law nonlinearity with exponent 1/15, which will be discussed in Sec. 8.1.7 below.

8.1 Components of PNCC processing

In this section we describe and discuss the major components of PNCC processing in greater detail. While the detailed description below assumes a sampling rate of 16 kHz, the PNCC features are easily modified to accommodate other sampling frequencies.

8.1.1 Initial processing

As in the case of MFCC, a pre-emphasis filter of the form $H(z) = 1 - 0.97z^{-1}$ is applied. A short-time Fourier transform (STFT) is performed using Hamming windows of duration 25.6 ms, with 10 ms between frames, using a DFT size of 512. Spectral power in 40 analysis bands is obtained by weighting the magnitude-squared STFT outputs for positive frequencies by the frequency response associated with a 40-channel gammatone-shaped filter bank [54]. whose center frequencies are linearly spaced in Equivalent Rectangular Bandwidth (ERB) [5] between 200 Hz and 8000 Hz, using the implementation of gammatone filters in Slaney's Auditory Toolbox [43]. In previous work [52] we observed that the use of gammatone frequency weighting provides slightly better ASR accuracy in white noise, but the differences compared to the traditional triangular weights in MFCC processing are small. The frequency response of the gammatone filterbank is shown in Fig. 8.2. In each channel the area under the squared transfer function is normalized to unity to satisfy the equation:

$$\int_0^{8000} |H_l(f)|^2 df = 1 \tag{8.1}$$

where $H_l(f)$ is the frequency response of the l^{th} gammatone channel.

We obtain the short-time spectral power $P[m, l]$ using the squared gammatone summation as below:

$$P[m, l] = \sum_{k=0}^{(K/2)-1} |X[m, e^{j\omega_k}) H_l(e^{j\omega_k})|^2 \tag{8.2}$$

where K is the DFT size, m and l represent the frame and channel indices, respectively, and $\omega_k = 2\pi k/F_s$, with F_s representing the sampling frequency. $X[m, e^{j\omega})$ is the short-time spectrum of the m^{th} frame of the signal.

8.1.2 Temporal integration for environmental analysis

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 (*e.g.* [32, 33]), because the power associated with most background noise conditions changes more slowly than the instantaneous power associated with speech.

In PNCC processing we estimate a quantity we refer to as "medium-duration power" $\tilde{Q}[m,l]$ by computing the running average of $P[m,l]$, the power observed in a single analysis frame, according to the equation:

$$\tilde{Q}[m,l] = \frac{1}{2M+1} \sum_{m'=m-M}^{m+M} P[m',l] \tag{8.3}$$

where m represents the frame index and l is the channel index. As mentioned before, we use a 25.6-ms Hamming window, and 10 ms between successive frames. As will be shown below in Sec. 8.2.2, it is observed that $M = 2$ is best for speech recognition performance, which corresponds to five consecutive windows for a total duration of 65.6 ms.

Since $\tilde{Q}[m,l]$ is the moving average of $P[m,l]$, $\tilde{Q}[m,l]$ is a low-pass filtered signal. If $M = 2$, the upper frequency is approximately 15 Hz. Nevertheless, if we were to use features based on $\tilde{Q}[m,l]$ directly for speech recognition, recognition accuracy would be degraded because onsets and offsets of the frequency components would become blurred. Hence in PNCC, we use $\tilde{Q}[m,l]$ only for noise estimation and compensation, which are used to modify the information based on the short-time power estimates $P[m'l]$. We also apply smoothing over the various frequency channels, which will discussed in Sec. 8.1.5 below. We will apply the tilde symbol to all power estimates that are performed using medium-time analysis.

8.1.3 Asymmetric noise suppression

In this section, we discuss a new approach to noise compensation which we refer to as *asymmetric noise suppression* (ANS). This procedure is motivated by the observation mentioned above that the speech power in each channel usually changes more rapidly than the background noise power in the same channel. Alternately we might say that speech usually has a higher-frequency modulation spectrum than noise. Motivated by this observation, many

algorithms have been developed using either high-pass filtering or band-pass filtering in the modulation spectrum domain (*e.g.* [3, 29]). The simplest way to accomplish this objective is to perform high-pass filtering in each channel (*e.g.* [28, 61]) which has the effect of removing slowly-varying components.

One significant problem with the application of conventional linear high-pass filtering in the power domain is that the filter output can become negative. Negative values for the power coefficients are not only problematic in the formal mathematical sense (in that power itself is positive) but they cause problems in the application of the compressive nonlinearity and in speech resynthesis unless a suitable floor value is applied to the power coefficients (*e.g.* [61]). Rather than filtering in the power domain, we could perform filtering after applying the logarithmic nonlinearity, as is done with conventional cepstral mean normalization in MFCC processing. Nevertheless, as will be seen in Sec. 8.2, this approach is not very helpful for environments with additive noise. Spectral subtraction is another way to reduce the effects of noise, whose power changes slowly (*e.g.* [10]). In spectral subtraction techniques, noise level is typically estimated from the power of non-speech segments (*e.g.* [10]) or using a continuous-update approach (*e.g.* [28]). In the approach that we introduce, we obtain a running estimate of the time-varying noise floor using an asymmetric nonlinear filter, and subtract that from the instantaneous power.

Figure 8.3 is a block diagram of the complete asymmetric nonlinear suppression processing. Let us begin by describing the general characteristics of the asymmetric nonlinear filter which is the first stage of processing. This filter is represented by the following equation for arbitrary input and output $\tilde{Q}_{in}[m,l]$ and $\tilde{Q}_{out}[m,l]$, respectively:

$$\tilde{Q}_{out}[m,l] = \begin{cases} \lambda_a \tilde{Q}_{out}[m-1,l] + (1-\lambda_a)\tilde{Q}_{in}[m,l], \\ \qquad \text{if } \tilde{Q}_{in}[m,l] \geq \tilde{Q}_{out}[m,l] \\ \lambda_b \tilde{Q}_{out}[m-1,l] + (1-\lambda_b)\tilde{Q}_{in}[m,l], \\ \qquad \text{if } \tilde{Q}_{in}[m,l] < \tilde{Q}_{out}[m,l] \end{cases} \tag{8.4}$$

where m is the frame index and l is the channel index, and λ_a and λ_b are constants between zero and one.

If $\lambda_a = \lambda_b$ it is easy to verify that Eq. 8.4 reduces to a conventional IIR filter that is lowpass in nature because of the positive values of the λ parameters, as shown in Fig. 8.4(a).

In contrast, If $1 > \lambda_b > \lambda_a > 0$, the nonlinear filter functions as a conventional "upper" envelope detector, as illustrated in Fig. 8.4(b). Finally, and most usefully our purposes, if $1 > \lambda_a > \lambda_b > 0$, the filter output $\tilde{Q}_{out}$ tends to follow the *lower envelope* of $\tilde{Q}_{in}[m,l]$, as seen in Fig. 8.4(c). In our processing, we will use this slowly-varying lower envelope in Fig. 8.4(c) to serve as a model for the estimated medium-time noise level, and the activity above this envelope is assumed to represent speech activity. Hence, subtracting this low-level envelope from the original input $\tilde{Q}_{in}[m,l]$ will remove a slowly varying non-speech component.

We will use the notation

$$\tilde{Q}_{out}[m,l] = \mathcal{ANS}_{\lambda_a,\lambda_b}[\tilde{Q}_{in}[m,l]] \tag{8.5}$$

to represent the nonlinear filter described by Eq. 8.4. We note that that this filter operates only on the frame indices m for each frequency index l.

Keeping the characteristics of the asymmetric filter described above in mind, we may now consider the structure shown in Fig. 8.3. In the first stage, the lower envelope $\tilde{Q}_{le}[m,l]$, which represents the average noise power, is obtained by ANS processing according to the equation

$$\tilde{Q}_{le}[m,l] = \mathcal{ANS}_{0.999,0.5}[\tilde{Q}[m,l]] \tag{8.6}$$

as depicted in Fig. 8.4(c). $\tilde{Q}_{le}[m,l]$ is subtracted from the input $\tilde{Q}[m,l]$, effectively highpass filtering the input, and that signal is passed through an ideal half-wave linear rectifier to produce the rectified output $\tilde{R}_1[m,l]$. The impact of the specific values of the forgetting factors λ_a and λ_b on speech recognition accuracy is discussed in Sec. 8.2.2.

The remaining elements of ANS processing in the lower half of Fig. 8.3 (excluding the temporal masking block) are included to cope with problems that develop when the rectifier output $\tilde{R}_1[m,l]$ remains zero for an interval, or when the local variance of $\tilde{R}_1[m,l]$ becomes excessively small. Our approach to this problem is motivated by previous work [32] in which it was noted that applying a well-motivated time-varying minimum to the local power coefficients improves robustness. In PNCC processing we apply the asymmetric nonlinear filter for a second time to obtain the lower envelope of the rectifier output $\tilde{R}_{le}[m,l]$, and we use this envelope as a flooring level. This envelope $R_{le}[m,l]$ is obtained using asymmetric

filtering as before:

$$R_{le}[m,l] = \mathcal{ANS}_{0.999,0.5}[\tilde{R}_1[m,l]] \tag{8.7}$$

We use the lower envelope of the rectified signal $\tilde{R}_{le}[m,l]$ as a floor level for the rectified signal $\tilde{R}[m,l]$ to provide the final speech output before temporal masking:

$$\tilde{R}[m,l] = \max\left(\tilde{R}[m,l], \tilde{R}_{le}[m,l]\right) \tag{8.8}$$

Temporal masking for speech segments is discussed in Sec. 8.1.4.

We have found that applying lowpass filtering to the non-speech segments improves recognition accuracy in noise by a small amount, and for that reason we use the lower envelope of the rectified signal $\tilde{R}_{le}[m,l]$ directly for these non-speech segments. This operation, which is effectively a further lowpass filtering, is not performed for the speech segments because blurring the power coefficients for speech degrades recognition accuracy.

Speech/non-speech decisions for this purpose are obtained for each value of m and l in a very simple fashion:

$$\text{“speech segment” if} \quad \tilde{Q}[m,l] \;\geq\; c\tilde{Q}_{le}[m,l] \tag{8.9a}$$

$$\text{“non-speech segment” if} \quad \tilde{Q}[m,l] \;<\; c\tilde{Q}_{le}[m,l] \tag{8.9b}$$

where $\tilde{Q}_{le}[m,l]$ is the lower envelope of $\tilde{Q}[m,l]$ as described above, and in and c is a fixed constant. In other words, a particular value of $\tilde{Q}[m,l]$ is considered to represent speech rather than background noise if it is greater than a fixed multiple of its own lower envelope. We use the parameter value $c = 2$ for reasons to be discussed in Sec. 8.2.2.

8.1.4 Temporal masking

Many authors have noted that the human auditory system appears to focus more on the onset of an incoming power envelope rather than the falling edge of that same power envelope (*e.g.* [57, 62, 63]). In this section we describe a simple way to incorporate this effect in PNCC processing, by obtaining a moving peak for each frequency channel l and suppressing the instantaneous power if it falls below this envelope.

The processing invoked for temporal masking is depicted in block diagram form in Fig. 8.5. We first obtain on-line peak power $R_t[m,l]$ for each channel using the following equation:

$$\tilde{R}_t[m,l] = \max\left(\lambda_t \tilde{R}_t[m-1,l], \tilde{R}[m,l]\right) \tag{8.10}$$

where λ_t is the forgetting factor for obtaining the on-line peak. As before, m is the frame index and l is the channel index. Temporal masking for speech segments is accomplished using the following equation:

$$\tilde{S}_{sp}[m,l] = \begin{cases} \tilde{R}[m,l], & \tilde{R}[m,l] > \lambda_t \tilde{R}_t[m-1,l] \\ \mu_t \tilde{R}_t[m-1,l], & \tilde{R}[m,l] < \lambda_t \tilde{R}_t[m-1,l] \end{cases} \tag{8.11}$$

We use values of $\lambda_t = 0.85$ and $\mu_t = 0.2$ for reasons that are discussed in Sec. 8.2.2 below.

Figure 8.6 illustrates the effect of this temporal masking. In general, with temporal masking the response of the system is inhibited for portions of the input signal $\tilde{R}[m,l]$ other than rising "attack transients". The difference between the signals with and without masking is especially pronounced in reverberant environments, for which the temporal processing module is especially helpful.

The final output of the asymmetric noise suppression and temporal masking modules is $\tilde{S} = \tilde{S}_{sp}$ for the speech segments and $\tilde{S} = \tilde{R}_{le}$ for the non-speech segments.

8.1.5 Spectral weight smoothing

In our previous research on speech enhancement and noise compensation techniques (*e.g.*, [52, 32, 33, 42, 34]) it has been frequently observed that smoothing the response across channels is helpful. This is true especially in processing schemes such as PNCC where there are nonlinearities and/or thresholds that vary in their effect from channel to channel, as well as processing schemes that are based on inclusion of responses only from a subset of time frames and frequency channels (*e.g.* [42]) or systems that rely on missing-feature approaches (*e.g.* [14]).

From the discussion above, we can represent the combined effects of asymmetric noise suppression and temporal masking for a specific time frame and frequency bin as the transfer function $\tilde{S}[m,l]/\tilde{Q}[m,l]$. Smoothing the transfer function across frequency is accomplished by

computing the running average over the frequency index l of the ratio $\tilde{S}[m,l]/\tilde{Q}[m,l]$. Hence, the frequency averaged weighting function $\tilde{T}[m,l]$ (which had previously been subjected to temporal averaging) is given by:

$$\tilde{T}[m,l] \;=\; \left(\frac{1}{l_2 - l_1 + 1} \sum_{l'=l_1}^{l_2} \frac{\tilde{S}[m,l']}{\tilde{Q}[m,l']} \right) \tag{8.12}$$

where $l_2 = \min(l - N, L)$ and $l_1 = \max(l + N, 1)$, and L is the total number of channels.

The time-averaged frequency-averaged transfer function $\tilde{T}[m,l]$ is used to modulate the original short-time power $P[m,l]$:

$$U[m,l] = P[m,l]\tilde{T}[m,l] \tag{8.13}$$

In the present implementation of PNCC, we use a value of $N = 4$ and a total number of $L = 40$ gammatone channels, for reasons discussed in Sec. 8.2.2. Note that if we were to use a different number of channels, the optimal value of N would be also change.

8.1.6 Mean power normalization

In conventional MFCC processing, multiplication of the input signal by a constant scale factor produces only an additive shift of the C_0 coefficient because a logarithmic nonlinearity is included in the processing, and this shift is easily removed by cepstral mean normalization. In PNCC processing, however, the replacement of the log nonlinearity by a power-law nonlinearity as discussed below, causes the response of the processing to be sensitive to changes in absolute power. In order to minimize the potential impact of amplitude scaling in PNCC we invoke a stage of mean power normalization.

While the easiest way to normalize power would be to divide the instantaneous power by the average power over the utterance, this is not feasible for real-time online processing because of the "look ahead" that would be required. For this reason, we normalize input power in the present online implementation of PNCC by dividing the incoming power by a running average of the overall power. The running power estimate $\mu_p[m]$ is computed from the simple difference equation:

$$\mu_p[m] = \lambda_p \mu_p[m-1] + \frac{(1-\lambda_p)}{L} \sum_{l=0}^{L-1} U[m,l] \tag{8.14}$$

where m and l are the frame and channel indices, as before, and L represents the number of frequency channels. We use a value of 0.999 for the forgetting factor λ_p.

The normalized power is obtained directly from the running power estimate $\mu_p[m]$:

$$V[m,l] = k \frac{U[m,l]}{\mu_p[m]} \tag{8.15}$$

where the value of the constant k is arbitrary. In pilot experiments we found that the speech recognition accuracy obtained using the online power normalization described above is comparable to the accuracy that would be obtained by normalizing according to a power estimate that is computed over the entire estimate in offline fashion.

8.1.7 Rate-level nonlinearity

Several studies in our group (*e.g.* [52, 64]) have confirmed the critical importance of the nonlinear function that describes the relationship between incoming signal amplitude in a given frequency channel and the corresponding response of the processing model. This "rate-level nonlinearity" is explicitly or implicitly a crucial part of every conceptual or physiological model of auditory processing (*e.g.* [**?**, **?**, 46]). In this section we summarize our approach to the development of the rate-level nonlinearity used in PNCC processing.

It is well known that the nonlinear curve relating sound pressure level in decibels to the auditory-nerve firing rate is compressive (*e.g* [65] [1]). It has also been observed that the average auditory-nerve firing rate exhibits an overshoot at the onset of an input signal. As an example, we compare in Fig. 8.8 the average onset firing rate versus the sustained rate as predicted by the model of Heinz *et al.* [1]. The curves in this figure were obtained by averaging the rate-intensity values obtained from sinusoidal tone bursts over seven frequencies, 100, 200, 400, 800, 1600, 3200, and 6400 Hz. For the onset-rate results we partitioned the response into bins of length of 2.5 ms, and searched for the bin with maximum rate during the initial 10 ms of the tone burst. To measure the sustained rate, we averaged the response rate between 50 and 100 ms after the onset of the signals. The curves were generated under the assumption that the spontaneous rate is 50 spikes/second. We observe in Fig. 8.8 that

the sustained firing rate (broken curve) is S-shaped with a threshold around 0 dB SPL and a saturating segment that begins at around 30 dB SPL. The onset rate (solid curve), on the other hand, increases continuously without apparent saturation over the conversational hearing range of 0 to 80 dB SPL. We choose to model the onset rate-intensity curve for PNCC processing because of the important role that it appears to play in auditory perception.

Figure 8.9 compares the onset rate-intensity curve depicted in Fig. 8.8 with various analytical functions that approximate this function. The curves are plotted as a function of dB SPL in the lower panel of the figure and as a function of absolute pressure in Pascals in the upper panel, and the putative spontaneous firing rate of 50 spikes per second is subtracted from the curves in both cases.

The most widely used current feature extraction algorithms are Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP) coefficients. Both the MFCC and PLP procedures include an intrinsic nonlinearity, which is logarithmic in the case of MFCC and a cube-root power function in the case of PLP analysis. We plot these curves relating the power of the input pressure p to the response s in Fig. 8.9 using values of the arbitrary scaling parameters that are chosen to provide the best fit to the curve of the Heinz *et al.* model, resulting in the following equations:

$$s_{cube} = 4294.1p^{2/3} \tag{8.16}$$

$$s_{log} = 120.2\log(p) + 1319.3 \tag{8.17}$$

We note that the exponent of the power function is doubled because we are plotting power rather than pressure. Even though scaling and shifting by fixed constants in Eqs. (8.16) and (8.18) do not have any significance in speech recognition systems, we included them in the above equation to fit these curves to the rate-intensity curve in Fig. 8.9(a). The constants in Eqs. (8.16) and (8.18) are obtained using an MMSE criterion for the sound pressure range between 0 dB (20μPa) and 80 dB (0.2 Pa) from the linear rate-intensity curve in the upper panel of Fig. 8.8.

In our own work in PNCC processing we also use a power-law nonlinearity but with an exponent of 1/15, which was selected for reasons that are discussed in Sec. 8.2.2 below. In

other words, in PNCC processing we use the power-law nonlinearity

$$V[m,l] \;\; = \;\; U[m,l]^{2/15} \tag{8.18}$$

where again $U[m,l]$ and $V[m,l]$ have the dimensions of power. This curve is closely approximated by the equation

$$s_{power} = 1389.6p^{0.1264} \tag{8.19}$$

which is also plotted in Fig. 8.9, again using a set of arbitrary free parameters that provide the best fit to the Heinz *et al.* data as depicted in the upper panel of Fig. 8.8. As before, a best fit in the MMSE sense was obtained over the interval of are obtained using a MMSE criterion in the sound pressure range between 0 dB (20μPa) and 80 dB (0.2 Pa).

The power low function was chosen for PNCC processing for several reasons. First, it is a relationship that is not affected in form by multiplying the input by a constant. Second, it has the attractive property that its asymptotic response at very low intensities is zero rather than negative infinity, which reduces variance in the response to low-level inputs such as spectral valleys or silence segments. Finally, the power law has been demonstrated to provide a good approximation to the "psychophysical transfer functions" that are observed in experiments relating the physical intensity of sensation to the perceived intensity using direct magnitude-estimation procedures (*e.g.* [49]). In general, we believe that the power-law curve with an exponent of 1/15 for sound pressure provides a reasonably good fit to the physiological data while optimizing recognition accuracy in the presence of noise. For further discussion of the development of this nonlinearity the reader is referred to [**?**].

Figure 8.10 is a final comparison of the effects of the asymmetric noise suppression, temporal masking, channel weighting, and power-law nonlinearity modules discussed in Secs. 8.1.3 through 8.1.7. The curves in both panels compare the response of the system [WHICH FILTER BAND?] to clean speech and speech in the presence of street noise at an SNR of 5 dB. The curves in the upper panel were obtained using conventional MFCC processing, including the logarithmic nonlinearity and without ANS processing or temporal masking. The curves in the lower panel were obtained using PNCC processing, which includes the power-law transformation described in this section, as well as ANS processing and temporal masking. We note that the difference between the two curves representing clean and noisy

speech is much greater with MFCC processing (upper panel), especially for times at which the signal is at a low level.

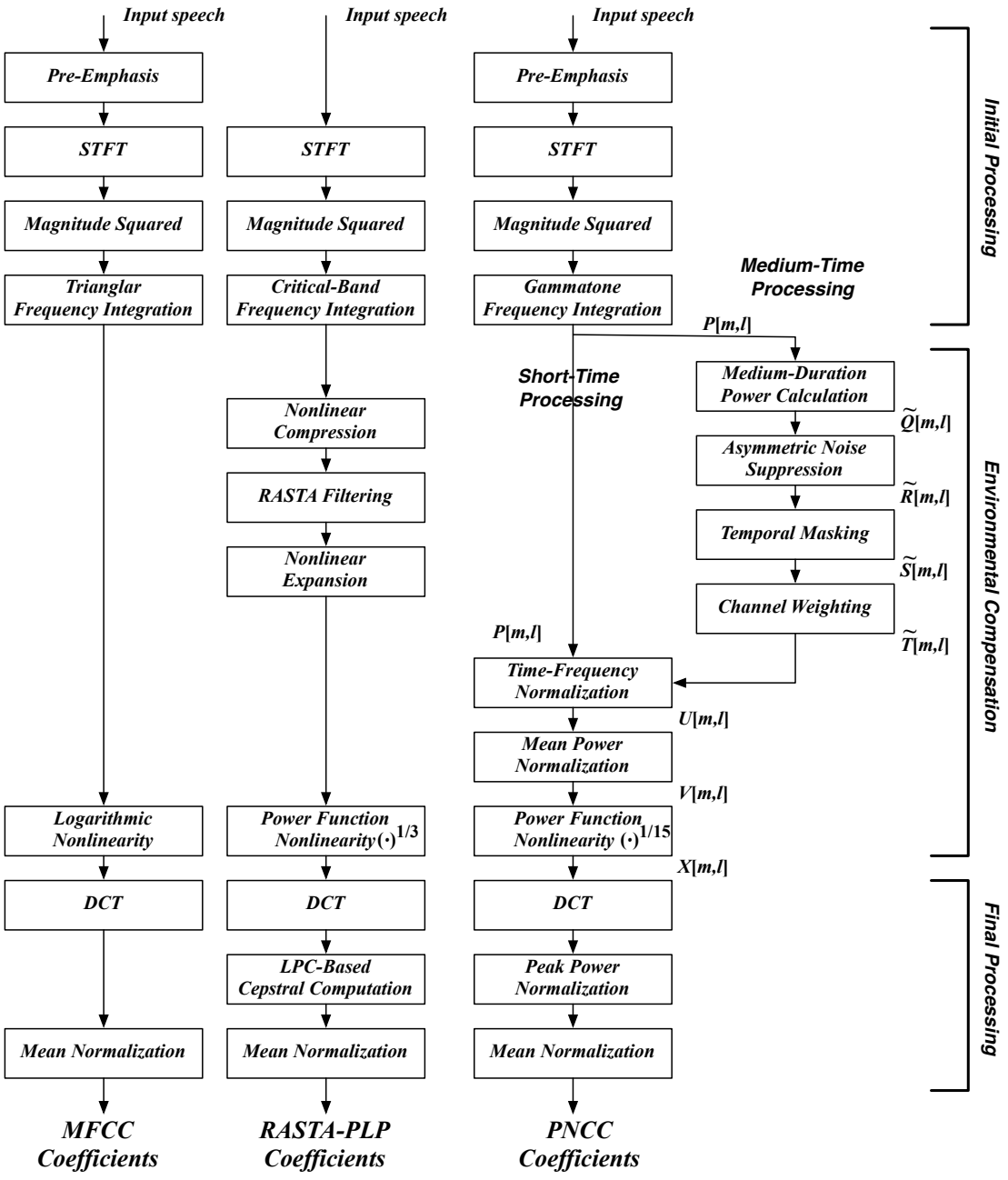


Fig. 8.1: Comparison of the structure of the MFCC, PLP, and PNCC feature extraction algorithms. The modules of PNCC that function on the basis of "medium-time" analysis (with a temporal window of 70 ms) are plotted in the rightmost column. The PNCC processing depicted applies to speech segments; non-speech segments are processed slightly differently, as discussed in Sec. 8.1.3.

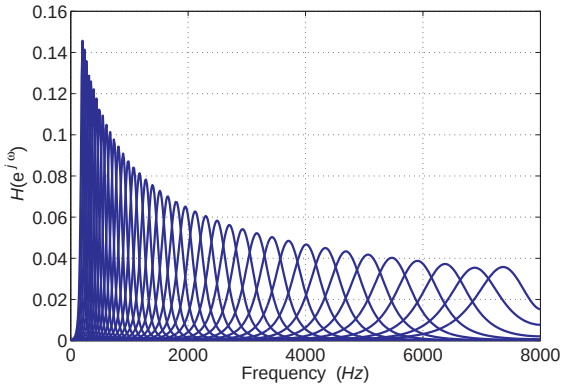


Fig. 8.2: The frequency response of a gammatone filterbank with each area of the squared frequency response is normalized to be unity. Characteristic frequencies are uniformly spaced between 200 and 8000 Hz according to the Equivalent Rectangular Bandwidth (ERB) scale [5].

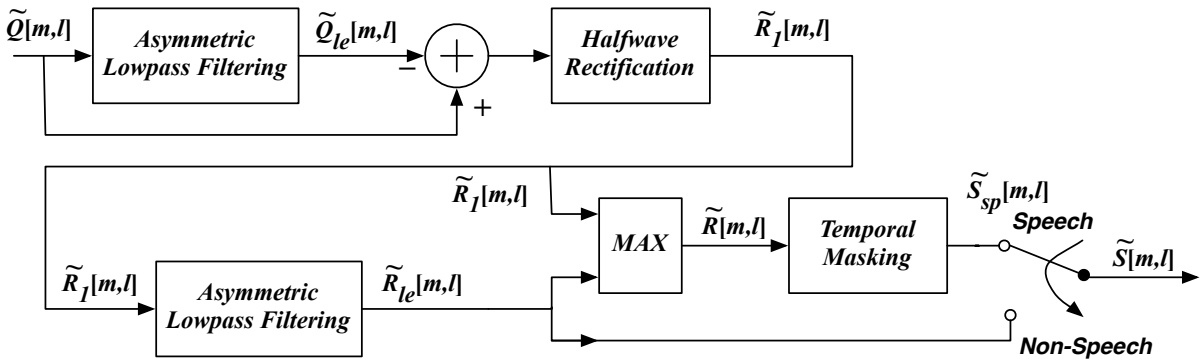


Fig. 8.3: Functional block diagram of the modules for asymmetric noise suppression (ANS) and temporal masking in PNCC processing. All processing is performed on a channel-by-channel basis. $\tilde{Q}[m,l]$ is the medium-time-averaged input power as defined by Eq.(8.3), $\tilde{R}[m,l]$ is the speech output of the ANS module, , and $\tilde{S}[m,l]$ is the output after temporal masking (which is applied only to the speech frames). The block labelled Temporal Masking is depicted in detail in Fig. 8.5

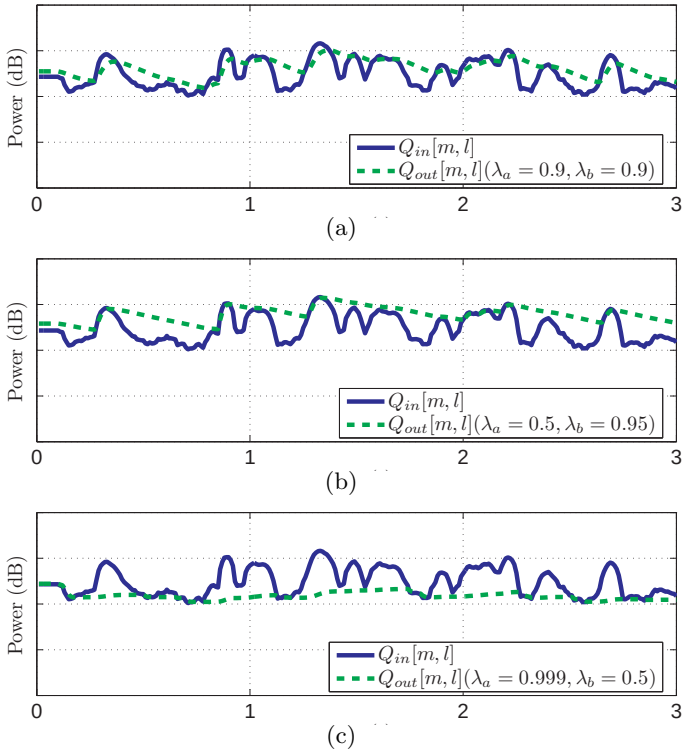


Fig. 8.4: Sample inputs (solid curves) and outputs (dashed curves) of the asymmetric nonlinear filter defined by Eq. (8.4) for conditions when (a) $\lambda_a = \lambda_b$ (b) $\lambda_a < \lambda_b$, and (c) $\lambda_a > \lambda_b$

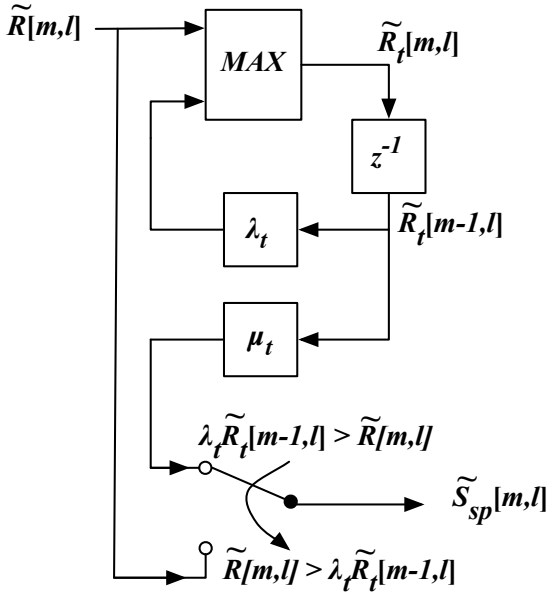


Fig. 8.5: Block diagram of the components that accomplish temporal masking in Fig. 8.3

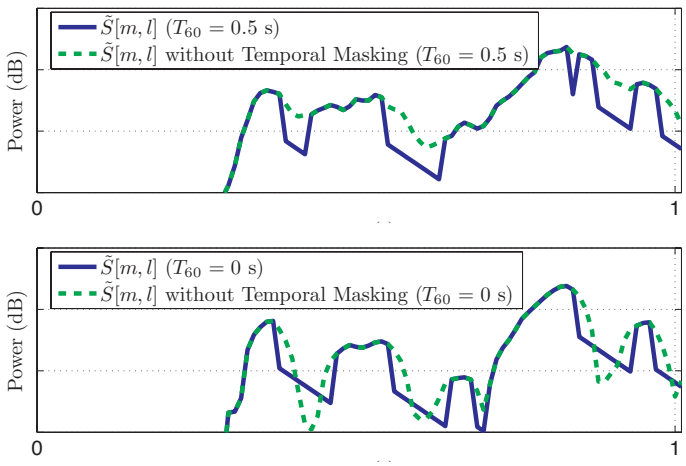


Fig. 8.6: Demonstration of the effect of temporal masking in the ANS module for speech in simulated revereration with $T_{60} = 0.5$ s (upper panel) and clean speech (lower panel).

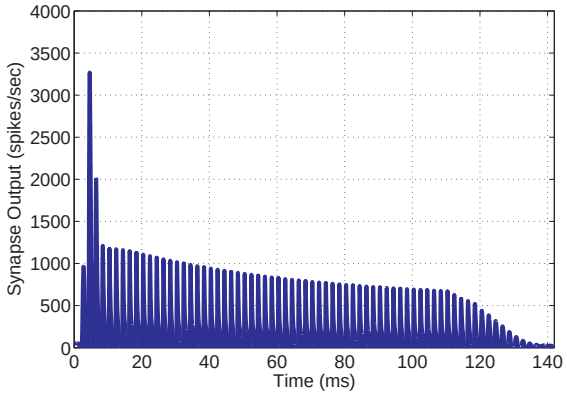


Fig. 8.7: Synapse output for a pure tone input with a carrier frequency of 500 Hz at 60 dB SPL. This synapse output is obtained using the auditory model by Heinz et al. [1].

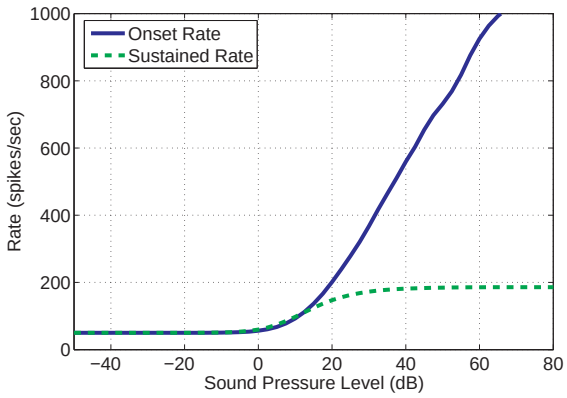


Fig. 8.8: Comparison of the onset rate (solid curve) and sustained rate (dashed curve) obtained using the model proposed by Heinz *et al.* [1]. The curves were obtained by averaging responses over seven frequencies. See text for details.

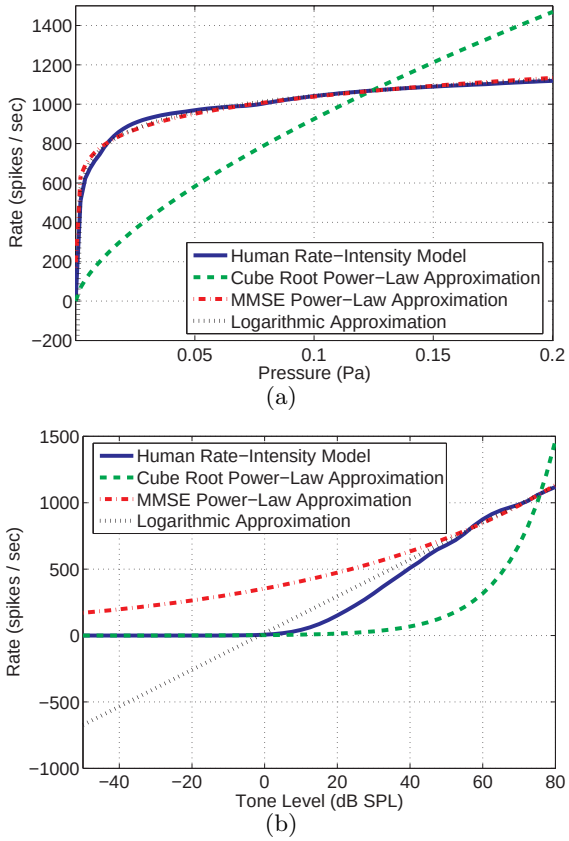


Fig. 8.9: Comparison between a human rate-intensity relation using the auditory model developed by Heinz *et al.* [1], a cube root power-law approximation, an MMSE power-law approximation, and a logarithmic function approximation. Upper panel: Comparison using the pressure (Pa) as the x-axis. Lower panel: Comparison using the sound pressure level (SPL) in dB as the x-axis.

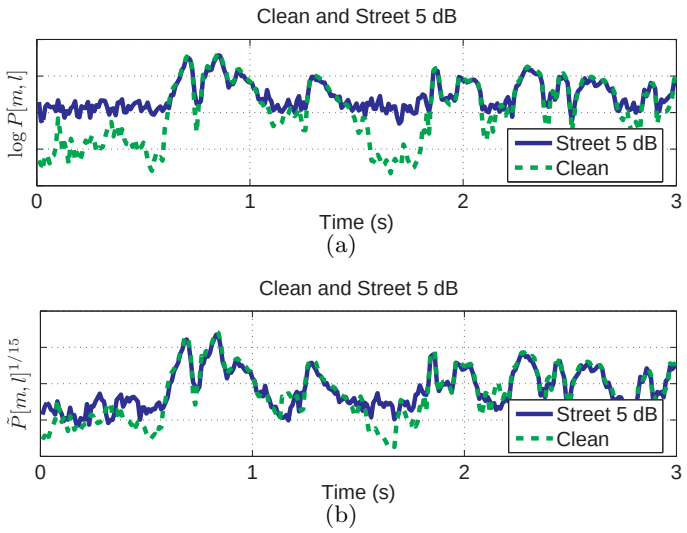


Fig. 8.10: Nonlinearity output of a specific channel under clean and noisy environment (corrupted by 5-dB street noise) (a) when we use the logarithmic nonlinearity without the ANS processing and the temporal masking (b) when we use the power-law nonlinearity with the ANS processing and the temporal masking

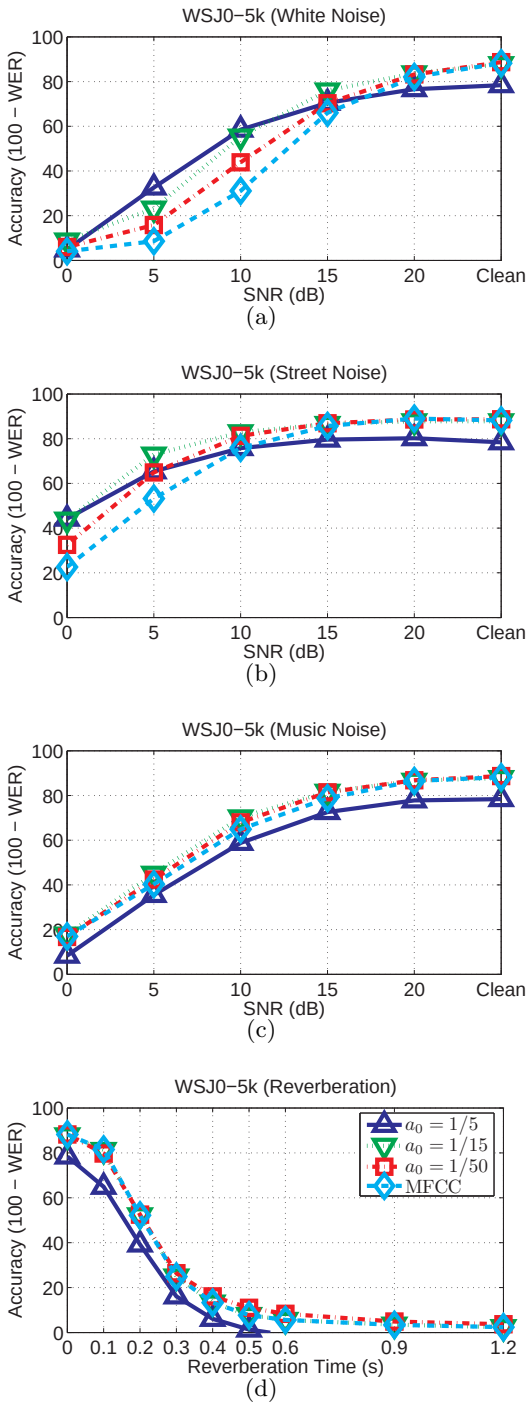


Fig. 8.11: Dependence on speech recognition accuracy on power coefficient in different environments: (a) additive white gaussian noise, (b) street noise, (c) background music, and (d) reverberant environment.

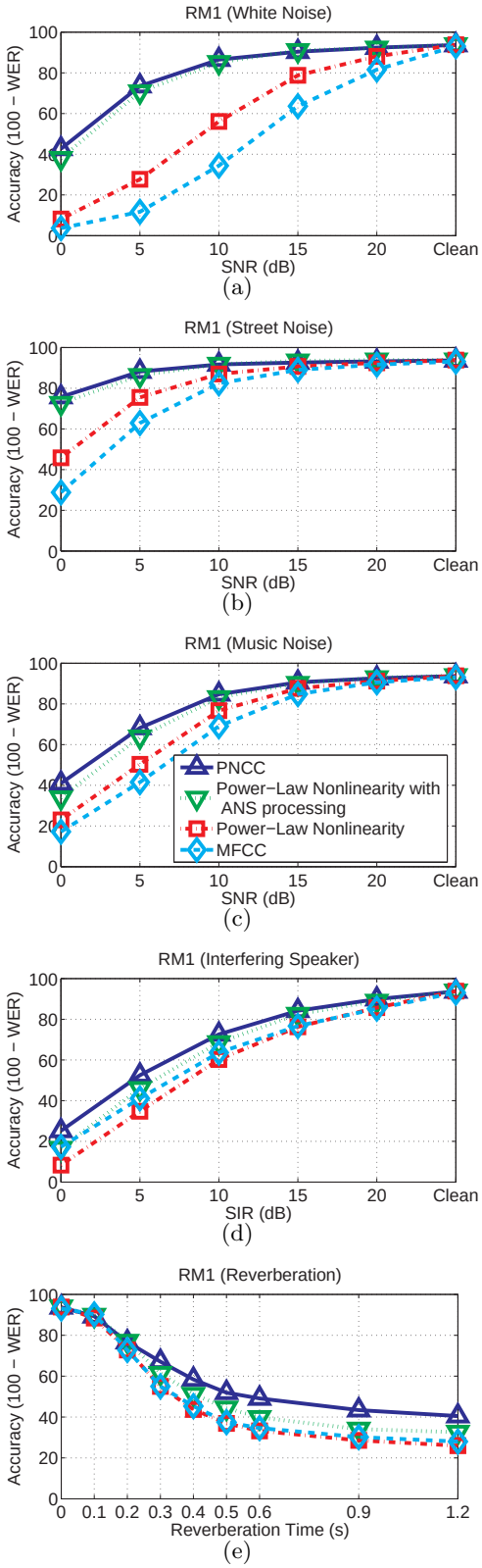


Fig. 8.12: The contribution of each block in the on-line PNCC. Speech recognition accuracy was obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of

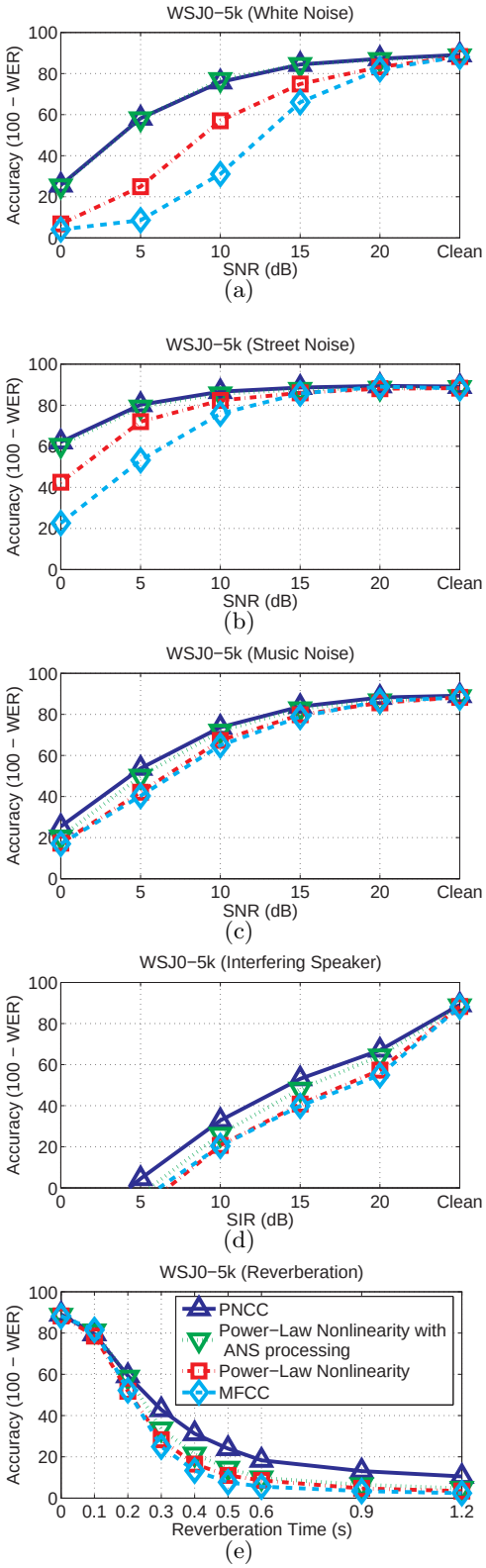


Fig. 8.13: The contribution of each block in the on-line PNCC. Speech recognition accuracy was obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of white Gaussian noise added to the data used in panel (c).

8.2 Experimental results and conclusions

In this section we present experimental results that are intended to demonstrate the superiority of PNCC processing over competing approaches in a wide variety of environments. We begin with a review of the experimental procedures that were used in Sec. 8.2.1. In Sec. 8.2.2 we describe the results of a series of pilot experiments that were run to select and optimize a number of the free parameters that are part of the PNCC processing. We assess the contributions of the various components in PNCC processing in Sec. 8.2.3, and we compare PNCC to a small number of other approaches in Sec. 8.2.4.

It should be noted that in general we selected an algorithm configuration and associated parameter values that provide very good performance over a wide variety of conditions using a single set of parameters and settings, without sacrificing word err rate in clean conditions relative to MFCC processing. In previous work we had describe slightly different feature extraction algorithms for speech recognition in the presence of reverberation [32] and background music [61], but each of these approaches do not perform as well as MFCC processing in clean speech. We used five standard testing environments in our work: (1) digitally-added white noise, (2) digitally-added noise that had been recorded live on urban streets, (3) digitally-added single-speaker interference, (4) digitally-added background music, and (5) passage of the signal through simulated reverberation. The street noise was recorded by us on streets with steady but moderate traffic. The masking signal consisted of other utterances in the same database for the single-speaker-interference experiments, and the background music was selected from music segments from the original DARPA Hub 4 Broadcast News evaluation. The reverberation simulations were accomplished using the *Room Impulse Response* open source software package [50] based on the image method [**?**]. The room size is 3 x 4 x 5 meters, the microphone is in the center of the room, the spacing between the target speaker and the microphone was assumed to be 3 meters, and reverberation time was manipulated by changing the assumed absorption coefficients in the room appropriately. These conditions were selected to include interfering additive noise sources of progressively greater difficulty, as well as basic reverberation effects.

8.2.1 Experimental Configuration

The PNCC feature 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. 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 used the PLP-RASTA implementation availabe at [27].

[CHECK FOR REDUNDANCIES BELOW (multiple headers)]

The PNCC system described in this paper was evaluated by comparing the recognition accuracy obtained using the CMU Sphinx 3.8 system with Sphinxbase 0.4.1, with PNCC introduced in this paper, (Sphinx webiste reference) with that of conventional MFCC processing, and with PLP processing as included in HCopy of HTK 3.4. (Dan-Ellips website reference) 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.

8.2.2 Optimization of parameter values

The effect of using medium-duration power is shown in Fig. 8.16. In this experiment, we used the entire PNCC structure and we changed only the M value and N value. N is a spectral weight averaging factor, which will be explained in Subsection 8.1.5. The speech recognition configuration is described in Subsection 8.2.1. As before, we trained the acoustic

model using WSJ0 si-84 training set and performed decoding on the WSJ0 5k test set. From the Fig. 8.16(b), we observe that we can significantly enhancing performance by temporal integration especially in stationary noise case. In a more difficult environment such as music noise, we still obtain improvements using temporal integration combined with spectral weight averaging as shown in Fig. 8.16(c).

Fig. 8.18 illustrates speech recognition accuracies depending on the choice of c. In this experiment, we used the entire PNCC structure excluding the temporal masking structure. From Fig. 8.18(b), we observe that $c = 2$ gives us the best performance for white noise. For music noise or reverberation, c has little effect as shown in the same figure.

Fig. 8.19 shows how recognition accuracy depends on the forgetting factor λ_t and the suppression factor μ_t. Experimental configuration is described in Subsection 8.2.1. In obtaining speech recognition results in this figure, we used the entire PNCC structure shown in Fig. 8.1 and changed only the forgetting factor λ_t and the suppression factor μ_t.

In clean environment, as shown in Fig. 8.19(a), if the forgetting factor is equal to or less than 0.85 and if $\mu_t \leq 0.2$, then performance remains almost constant. However, if λ_t is larger than 0.85, then performance degrades. Similar tendency is also observed in additive noise such as white and music noise as shown in Fig. 8.19(b) and in Fig. 8.19(c). For reverberation, as shown in Fig. 8.19(d), we observe that by applying the temporal masking scheme, we observe substantial benefit. As will be shown in Subsection 8.2.3, this temporal masking scheme also shows a remarkable improvement in a very difficult environment like a single-channel interfering speaker case.

Fig. 8.16 shows how recognition accuracy depends on the value of the smoothing parameter N. The x-axis of this figure represent the weight smoothing factor N. From this figure we can see that performance is best for $N = 3$ or $N = 4$. The improvement induced by weight smoothing is especially large for stationary noise like white noise as shown in Fig. 8.16(b). Noticeable improvement is also observed in a more difficult environment such as background music noise as shown in Fig. 8.16(b).

In the present implementation of PNCC, we use $N = 4$ and a total number of $L = 40$ gammatone channels.Note that if we use a different number of channels, then the optimal N would be also different.

To evaluate whether the power function approximation obtained in the above way is ef-

fective for speech recognition, we conducted speech recognition experiments using the WSJ si-84 training set on the WSJ0 5k test set. The detailed configuration about speech recognition experiments is described in Subsection 8.2.1. The general power function nonlinearity is described by the following equation:

$$s[m,l] \quad = \quad P[m,l]^{a_0} \tag{8.20}$$

where a_0 is the power coefficient and $P[m,l]$ is power in each time-frequency bin, which is shown in (8.2). When extracting features, we used the PNCC system shown in Fig. 8.1 excluding the blocks in red color. This is because we want to check the effectiveness of the rate nonlinearity part without being affected by other additional blocks in PNCC.

As shown in Fig. 8.11, the power function coefficient obtained from the MMSE power-fit gives us performance benefit compared to conventional logarithmic processing. If we use a bigger coefficient such as 1/5, it gives us better performance for white noise, but it loses performance in other environments as well as in clean environment.

8.2.3 Contribution of each component

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) [9], as shown in Fig. 9.10 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. 9.10. PNCC, on the other hand, degrades only slightly under the same conditions without VADs.

8.2.4 Comparison with other algorithms

Fig. **??** also demonstrates the amount of improvement provided by (1) the switch from the triangular MFCC filters to Gammatone filters, (2) the switch from the logarithmic nonlin-

earity to the power law nonlinearity, and (3) the use of medium-duration power bias removal. 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.

8.3 Computational Complexity

Table 8.1 shows computational amount comparison between MFCC, PNCC, and PLP. We assume that the window length is 25.6 ms and the interval between successive windows is 10 ms. The sampling rate is assumed to be 16 kHz, and we use a 512-pt FFT for each frame. Regarding the FFT size, we observe that 1024-pt FFT might do slightly better than 512-pt FFT (since it results in better resolution in frequency windows), but the performance difference is rather small. For the MFCC structure, we use the implementation in $sphinx_fe$ as a reference, and for PLP, we use the implementation in [27] as reference. We calculate the computational amount of PNCC using two different gammatone integration schemes: First, we use the original gammatone window shown in Fig. 8.2 using (8.2). Second, we use a truncated gammatone frequency windows. More specifically, if the amplitude of the response at a specific frequency index is less than 0.01 in Fig. 8.2, we assume that this value is zero, and don't perform multiplication.

In Table 8.1, values with the "asterisk" represent the computational amount using the truncated gammtaone frequency window. If we use the full gammatone frequency window, then computational amount of PNCC is roughly twice of MFCC. However, if we use a truncated gammatone frequency window, then it requires only 25 % more computation than MFCC.

As shown in Table 8.1, the most computationally costly parts are FFT and spectral integration. Since all three features include FFT, the difference in computational amount is primarily due to spectral integration. In case of MFCC, since the bandwidth of mel-filter banks is narrower than either the gammatone filter bank and the trapezoidal filter bank, the computational amount is less.

Tab. 8.1: Number of multiplications and divisions in each frame

Item	**MFCC**	**PNCC**	**PLP**
pre-emphasis	410	410	
windowing	410	410	410
FFT	9216	9216	9216
Magnitude Squared	512	512	512
Medium Duration Power Calculation		40	
Spectral Integration	413	10240 (1757*)	2268
ANS Filtering		160	
Equal Loudness Pre-emphasis			256
Temporal Masking		120	
Weight Averaging		400	
IDFT			21
LPC and Cepstra Recursion			21
DCT	40	40	
Sum	10742	21292 (12809*)	12448

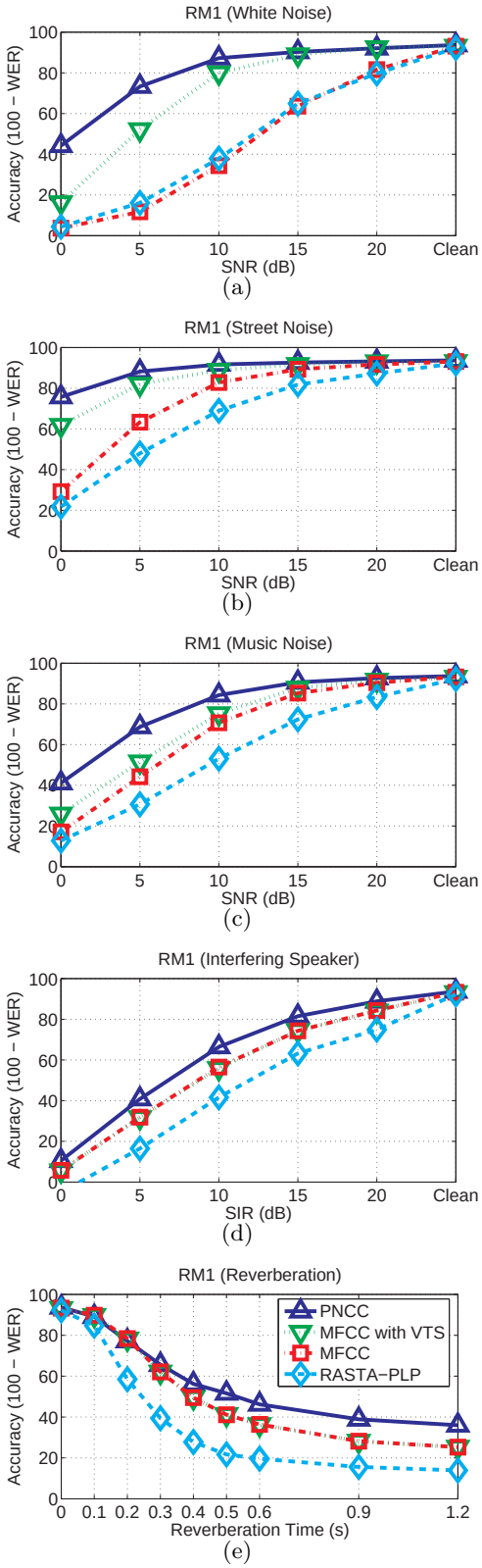


Fig. 8.14: Speech recognition accuracy obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of white Gaussian noise added to the data used in panel (c).

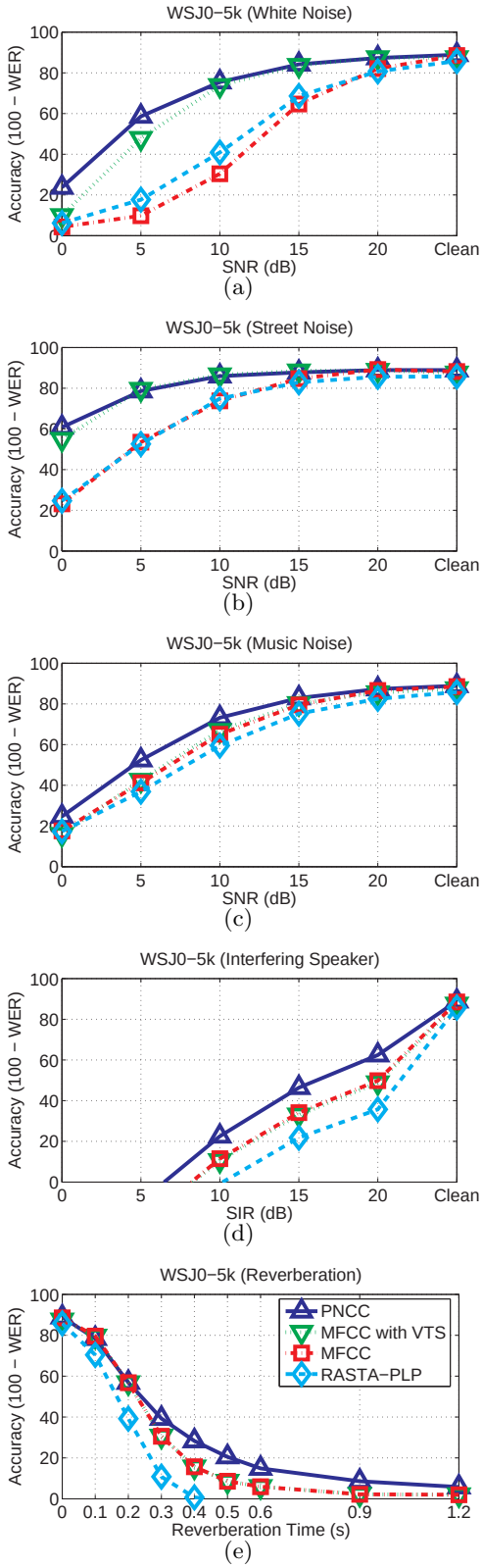


Fig. 8.15: Speech recognition accuracy obtained in different environments: (a) additive white gaussian noise, (b) background music, (c) silence prepended and appended to the boundaries of clean speech, and (d) 10-dB of white Gaussian noise added to the data used in panel (c).

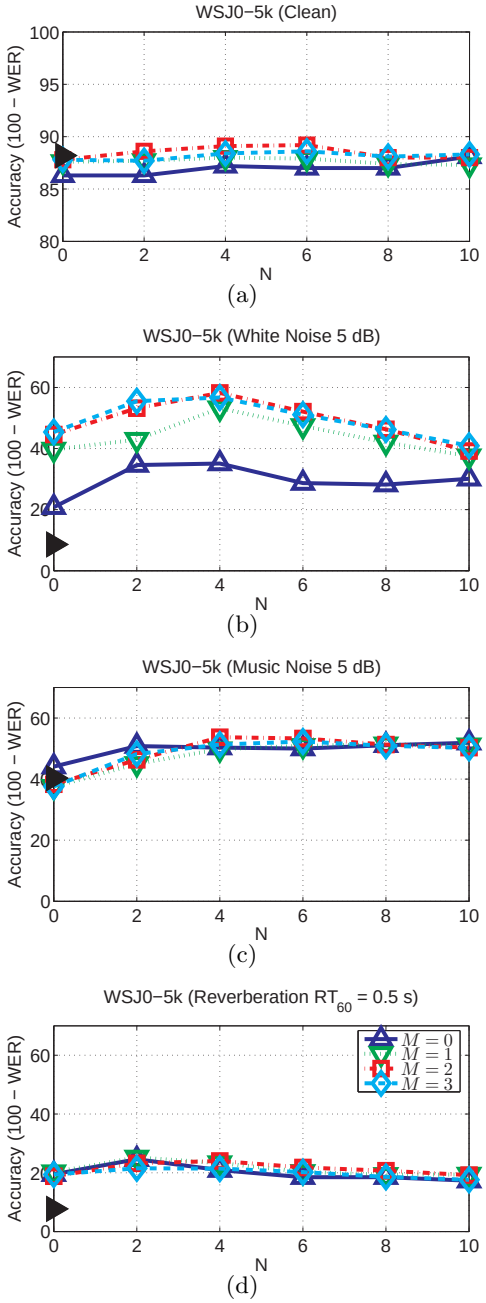


Fig. 8.16: The dependence of speech recognition accuracy on the value of the temporal integration factor M and spectral weight smoothing factor N. The filled triangle on the y-axis represents the baseline MFCC result for the same test set. (a) the WSJ0 5k clean test set, (b) 5-dB Gaussian white noise, (c) 5-dB musical noise, and (d) reverberation with $RT_{60} = 0.5$

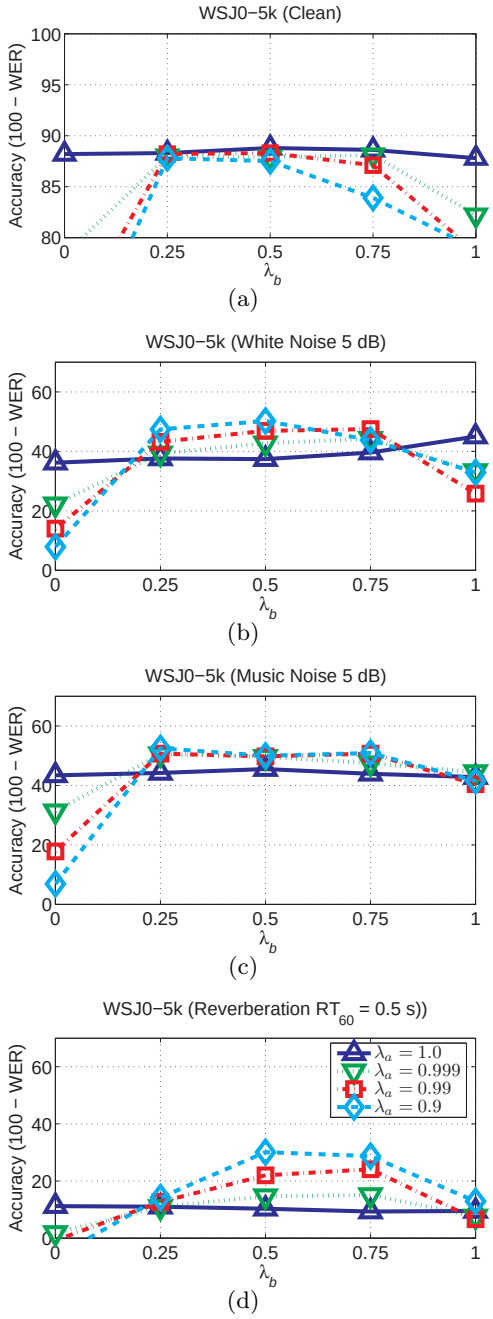


Fig. 8.17: The corresponding dependence of speech recognition accuracy on the forgetting factors λ_a and λ_b. The filled triangle on the y-axis represents the baseline MFF result for the same test set: (a) Clean, (b) 5-dB Gaussian white noise, (c) 5-dB musical noise, and (d) reverberation with $RT_{60} = 0.5$

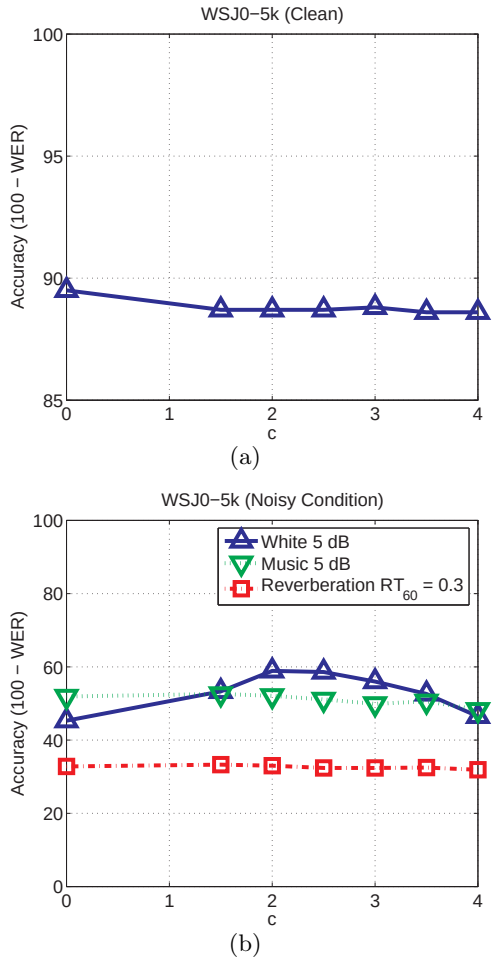


Fig. 8.18: The dependence of speech recognition accuracy on the speech/non-speech decision coefficient c in (8.9) : (a) clean and (b) noisy environment

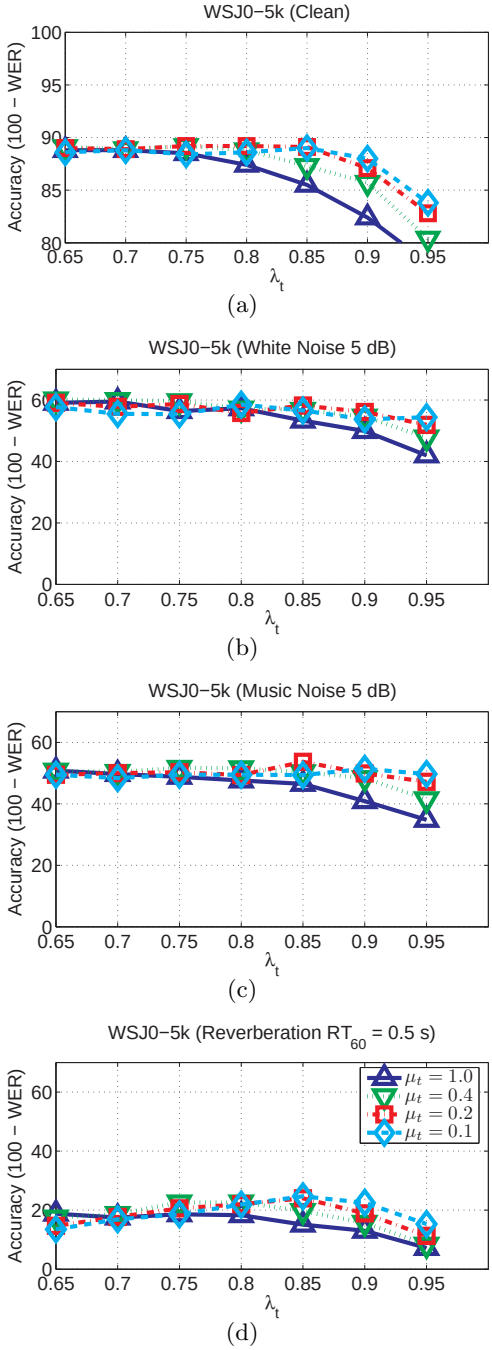


Fig. 8.19: The dependence of speech recognition accuracy on the forgetting factor λ_t and the suppression factor μ_t , which are used for temporal masking block. The filled triangle on the y-axis represents the baseline MFCC result for the same test set: (a) Clean, (b) 5-dB Gaussian white noise, (c) 5-dB musical noise, and (d) reverberation with $RT_{60} = 0.5$

9. 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 Interaual Intensity Difference (IID) (*e.g.* [16] [66]). 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.

9.1 Introduction

In recent days, speech recognition systems have significantly improved, and they have been used in many devices such as cell phones, Personal Data Assistants (PDAs), navigation systems and so on. Even though, we can obtain a very high speech recognition in clean environments using a state of the art speech recognition systems, performance seriously degrades in noisy environments. Thus, noise robustness still remains a critical issue for speech recognition systems to be used in real consumer products in difficult acoustical environments.

To tackle this problem, researchers have made a lot of efforts to enhance noise robustness of speech recognition systems. A number of algorithms have shown significant improvements for stationary noise (*e.g.* [11, 12, 8]). Nevertheless, improvement in non-stationary noise remains a difficult issue (*e.g.* [13]). In these environments, auditory processing (emphe.g.

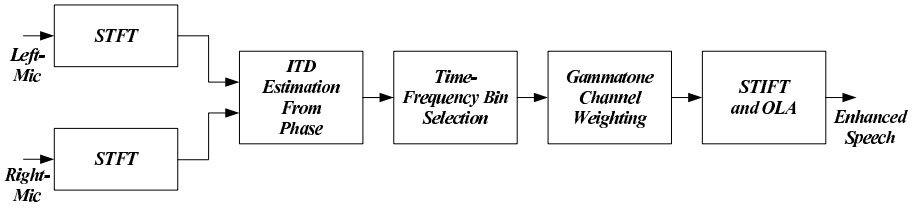


Fig. 9.1: The block diagram of the Phase Difference Channel Weighting (PDCW)) algorithm

[47] [34] [52]) and missing-feature-based approaches (emphe.g. [14]) 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] [67]). It has been observed that various types of cues are used to segregate the target signal from interfering sources. Motivated by these observations, many models and algorithms have been developed using interaural time differences (ITDs), interaural intensity difference (IIDs), interaural phase differences (IPDs), and other cues (*e.g.* [15, 16, 66]). IPD and ITD have been extensively used in binaural processing because this information can be easily obtained by spectral analysis (*e.g.* [66] [68] [42]).

In many of the algorithms above either a binary or continuous "mask" is developed that indicates which time-frequency bins that are believed to be dominated by the target source. Studies show that the continuous mask technique shows better performance than the binary masking technique (*e.g.* [16]), but it usually requires that we know the exact location of the noise source (*e.g.* [16]). Binary masking techniques (*e.g.* [52]) might be more realistic for omni-directional noise cases but we still need to know what would be an appropriate mask threshold. Typically this is done by sorting the time-frequency bins according to ITD (either calculated directly or inferred from estimated IPD). In both cases performance depends on how the threshold ITD for selection is selected, and the optimal threshold depends on the configuration of the noise sources including their locations and strength. If the optimal ITD from a particular environment is applied to a somewhat different environment, the system performance will be degraded. In addition, the characteristics of the environment typically vary with time.

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 algo-

rithms (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 domain. We describe in the sections below how the binary mask is obtained using frequency information. We also discuss the duration and shape of the analysis windows, which can contribute to further improvements in WER.

In many cases, we have the control of the target source, but we don't have control on the noise source.

When target identification is obtained by a binary masked based on an ITD threshold, the value of that threshold is typically estimated from development test data. As noted above, the optimal ITD threshold itself will depend on the number of noise sources and their locations, both of which may be time-varying. If the azimuth of the noise source is very different from that of the target, a threshold that ITD is relatively far from that of the target may be helpful. On the other hand, if an interfering noise source is very close to the target and we use a similar ITD threshold, the system will also classify many components of the interfering signal as part of the target signal. If there is more than one noise source, or if the noise sources are moving, the problem becomes even more complicated.

In our approach, we construct two complementary masks using a binary threshold. Using

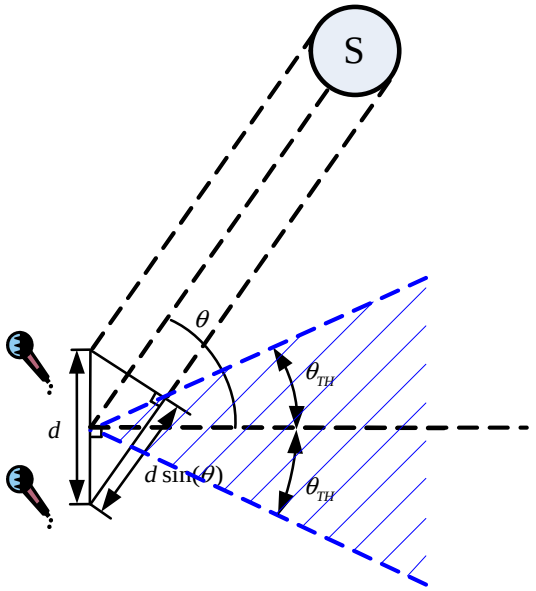


Fig. 9.2: Selection region in the binaural sound source separation system: If the location of the sound source is inside the shaded region, the sound source separation system assumes that it is a target. If the location of the sound source is outside this shaded region, then it is masked out by the sound source separation system.

these two complementary masks, we obtain two different spectra: one for the target and the other for everything except for the target. From these spectra, we obtain the short-time power for the target and the interference. These power sequences are passed through a compressive nonlinearity. We compute the cross-correlation coefficient for the two resulting power sequences, and we obtain the ITD threshold by minimizing the correlation coefficient.

The rest of the paper is organized as follows: In Sec. 9.2, we give an overview of the entire system structure. In Sec. 9.3, we discuss how to obtain ITD from phase difference information. Sec. 9.4 deals with temporal resolution and gammatone channel weighting to obtain further improvement in speech recognition accuracy. In Section 9.5, we explain how to construct the complementary masks and how to obtain the optimal ITD threshold. We present experimental results in Section 9.6.

9.2 Overview of PDCW-AUTO structure

In this section, we give overview of our sound source separation system. In the binaural sound source separation system, we usually assume that we have a prior knowledge about the target location. This is a reasonable assumption, because we usually have control over the target. For example, if the target is a user using a hand-held device with two microphones, then the user might be instructed to speak at a certain location with respect to the device. In this paper, we assume that the target is located along the perpendicular bisector to the line connecting two microphones. Under this assumption, let us consider a selection area as shown in Fig. 9.2, which is defined by an angle θ_{TH}. If the sound source is determined to be inside the shaded region in this figure, then we assume that it is a target. As shown in Fig. 9.2, suppose that there is a sound source S along a line with an angle θ. The we can set up a decision criterion as follows:

$$\begin{cases} \text{considered to be a target:} & \theta \leq \theta_{TH} \\ \text{considered to be a noise source:} & \theta > \theta_{TH} \end{cases} \tag{9.1}$$

In Fig. 9.2, if the sound source is located along the line of an angle θ, then using simple geometry, we find that the interaural distance d_i is given by:

$$d_i = d\sin(\theta) \tag{9.2}$$

where d is the distance between two microphones. In the discrete-time domain, the Interaural Time Delay (ITD) (in unit of discrete samples) is given by the following equation:

$$\tau = \frac{d\sin(\theta)}{c_0} f_s \tag{9.3}$$

where c_0 is the speed of sound and f_s is the sampling rate. Since d, c_0, and f_s are all fixed constants, θ is the only factor determining the ITD τ. Thus, the decision criterion in (9.1) can be expressed as follows:

$$\begin{cases} \text{considered to be a target:} & \tau < \tau_{TH} \\ \text{considered to be a noise source:} & \tau \leq \tau_{TH} \end{cases} \tag{9.4}$$

where $\tau_{TH} = \frac{d\sin(\theta_T H)}{c_0} f_s$. Thus, if we obtain a suitable ITD threshold then using (9.4) we can make a binary decision to determine whether the source is in the shaded region in Fig.

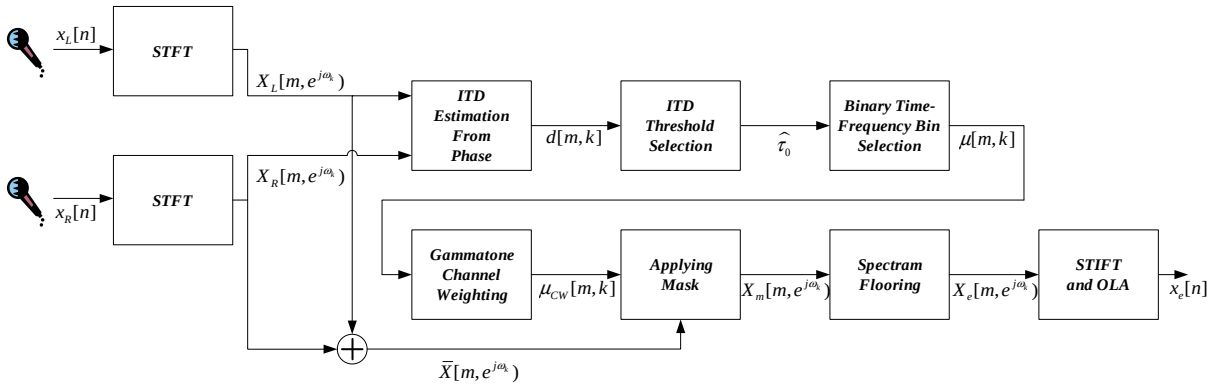


Fig. 9.3: The block diagram of a sound source separation system using the Phase Difference Channel Weighting (PDCW) algorithm and the automatic ITD threshold selection algorithm.

9.2. In our sound source separation system, ITD is obtained for each-time frequency bin, using phase information. Thus, a binary decision like (9.4) is made for each-time frequency bin. This procedure will be explained in detail in Sec. 9.3.

Our binaural sound source separation system is called Phase Difference Channel Weighting (PDCW), which was introduced in [42]. If we use an automatic threshold selection approach for selecting the target, we refer to the entire system as PDCW-AUTO. This automatic threshold selection approach is explained in detail in Section 9.5. If we use a fixed ITD threshold an angle θ_{TH}, which might be empirically chosen, we refer to this system as PDCW-FIXED.

Fig. 9.3 shows the entire structure of PDCW-AUTO. Using two-microphone signals, we obtain two spectra; one is from the left-microphone and the other is from the right-microphone. From the phase difference obtained these spectra, we calculate ITD. Using the ITD threshold selection algorithm, which is explained in Sec. 9.5, we construct binary masks for each time-frequency bin.

9.3 Obtaining ITD from phase information

In binaural sound source separation system, Our processing approach, which we refer to as Phase Difference Channel Weighting (PDCW), crudely emulates human binaural processing.

Its block diagram is shown in Fig. 9.3. 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 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 *a priori* and without loss of generality we assume its ITD to be equal to zero. In this section we review the procedure for obtaining ITD from phase information (*e.g.* [42]). Let $x_L[n]$ and $x_R[n]$ be the signals from the left and right microphones, respectively. We assume that we know where the target source is located and, without loss of generality, we assume that it is placed at the perpendicular bisector of the line between two microphones.

Suppose that the total number of interfering sources is S. Each source $s, 1 \leq s \leq S$ has an ITD of $d_s[m,k]$ where m is the frame index and k is the frequency index. Note that both S and $d_s[m,k]$ are unknown. We assume that $x_0[n]$ represents the target signal and that the notation $x_s[n], 1 \leq s \leq S$, represents signals from each interfering source received from the "left" microphone. In the case of signals from the "right" microphone, the target signal is still $x_0[n]$, but the interfering signals are delayed by $d_s[m,k]$. Note that for the target signal $x_0[n]$, $d_0[m,k] = 0$ for all m and k by the above assumptions.

To perform spectral analysis, we obtain the following short-time signals by multiplication with a hamming window $w[n]$:

$$x_L[n;m] = x_L[n - mL_{fp}]w[n] \tag{9.5a}$$

$$x_R[n;m] = x_R[n - mL_{fp}]w[n] \tag{9.5b}$$

$$\text{for } 0 \leq n \leq L_{fl} - 1$$

where m is the frame index, L_{fp} is the number of samples between frames, and L_{fl} is the frame length. The window $w[n]$ is a hamming window with a length of L_{fl}. We use a 75 ms

window length based on our previous result in [42]. The short-time Fourier transforms of (9.9) can be represented as

$$X_L[m, e^{j\omega_k}) = \sum_{s=0}^{S} X_s[m, e^{j\omega_k}) \tag{9.6a}$$

$$X_R[m, e^{j\omega_k}) = \sum_{s=0}^{S} e^{-jw_k d_s[m,k]} X_s[m, e^{j\omega_k}) \tag{9.6b}$$

where $w_k = 2\pi k/N$ and N is the FFT size. We represent the strongest sound source for a specific time-frequency bin $[m, k]$ as $s^*[m, k]$. This leads to the following approximation:

$$X_L[m, e^{j\omega_k}) \approx X_{s^*[m,k]}[m, e^{-jw_k}) \tag{9.7a}$$

$$X_R[m, e^{j\omega_k}) \approx e^{-jw_k d_{s^*[m,k]}[m,k]} \times X_{s^*[m,k]}[m, e^{-jw_k}) \tag{9.7b}$$

Note that $s^*[m, k]$ may be either 0 (the target source) or $1 \leq s \leq S$ (any of the interfering sources). From (9.11), The ITD for a particular time-frequency bin $[m, k]$ is given by:

$$|d_{s^*[m,k]}[m, k]| \approx \frac{1}{|w_k|} \min_r \Big| \angle X_R[m, e^{-jw_k}) - \angle X_L[m, e^{-jw_k}) - 2\pi r \Big| \tag{9.8}$$

Thus, by examining whether the obtained ITD from (9.12) is within a certain range from the target ITD (which is zero), we can make a simple binary decision concerning whether the time-frequency bin $[m, k]$ is likely to belong to the target speaker or not.

In this section we review the procedure for obtaining ITD from phase information (*e.g.* [42]). Let $x_L[n]$ and $x_R[n]$ be the signals from the left and right microphones, respectively. We assume that we know where the target source is located and, without loss of generality, we assume that it is placed at the perpendicular bisector of the line between two microphones.

Suppose that the total number of interfering sources is S. Each source $s, 1 \leq s \leq S$ has an ITD of $d_s[m, k]$ where m is the frame index and k is the frequency index. Note that both S and $d_s[m, k]$ are unknown. We assume that $x_0[n]$ represents the target signal and that the notation $x_s[n], 1 \leq s \leq S$, represents signals from each interfering source received from the "left" microphone. In the case of signals from the "right" microphone, the target signal is

still $x_0[n]$, but the interfering signals are delayed by $d_s[m,k]$. Note that for the target signal $x_0[n]$, $d_0[m,k] = 0$ for all m and k by the above assumptions.

To perform spectral analysis, we obtain the following short-time signals by multiplication by a hamming window $w[n]$:

$$\begin{aligned} x_L[n;m] &= x_L[n - mL_{fp}]w[n] && (9.9a) \\ x_R[n;m] &= x_R[n - mL_{fp}]w[n] && (9.9b) \\ & \quad \text{for } 0 \le n \le L_{fl} - 1 \end{aligned}$$

where m is the frame index, L_{fp} is the number of samples between frames, and L_{fl} is the frame length. The window $w[n]$ is a hamming window with a length of L_{fl}. The short-time Fourier transforms of (9.9) can be represented as

$$\begin{aligned} X_L[m, e^{j\omega_k}) &= \sum_{s=0}^{S} X_s[m, e^{j\omega_k}) && (9.10a) \\ X_R[m, e^{j\omega_k}) &= \sum_{s=0}^{S} e^{-jw_k d_s[m,k]} X_s[m, e^{j\omega_k}) && (9.10b) \end{aligned}$$

where $w_k = 2\pi k/N$ and N is the FFT size. We represent the strongest sound source for a specific time-frequency bin $[m,k]$ as $s^*[m,k]$. This leads to the following approximation:

$$\begin{aligned} X_L[m, e^{j\omega_k}) &\approx X_{s^*[m,k]}[m, e^{-jw_k}) && (9.11a) \\ X_R[m, e^{j\omega_k}) &\approx e^{-jw_k d_{s^*[m,k]}[m,k]} \\ & \quad \times X_{s^*[m,k]}[m, e^{-jw_k}) && (9.11b) \end{aligned}$$

Note that $s^*[m,k]$ may be either 0 (the target source) or $1 \le s \le S$ (any of the interfering sources). From (9.11), The ITD for a particular time-frequency bin $[m,k]$ is given by:

$$\begin{aligned} |d_{s^*[m,k]}[m,k]| &\approx \frac{1}{|w_k|} \min_r \Big| \angle X_R[m, e^{-jw_k}) \\ & \quad - \angle X_L[m, e^{-jw_k}) - 2\pi r \Big| && (9.12) \end{aligned}$$

Thus, by examining whether the obtained ITD from (9.12) is within a certain range from the target ITD (which is zero), we can make a simple binary decision concerning whether the time-frequency bin $[m,k]$ is likely to belong to the target speaker or not.

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.

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.

9.4 Temporal and Frequency Resolution

While the basic procedure described in Sec. 9.3 provides signals that are audibly separated, the phase estimates are generally too noisy to provide useful speech recognition accuracy. In this subsection we discuss the implementation of two methods that smooth the estimates over frequency and time.

9.4.1 Temporal resolution

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 as shown in our previous works (*e.g.* [33, 34, 32, 61, 52]). Using the configuration described in Sec. 9.6, we evaluated the effect of window length on recognition accuracy. Using the PD-FIXED structure explained in Subsection 9.3, we conducted speech recognition experiments. In Fig. 9.5(b) which indicate that best performance is achieved with window length of about 75 ms. In the experiments described below we Hamming windows of duration 75 ms with 37.5 ms between successive

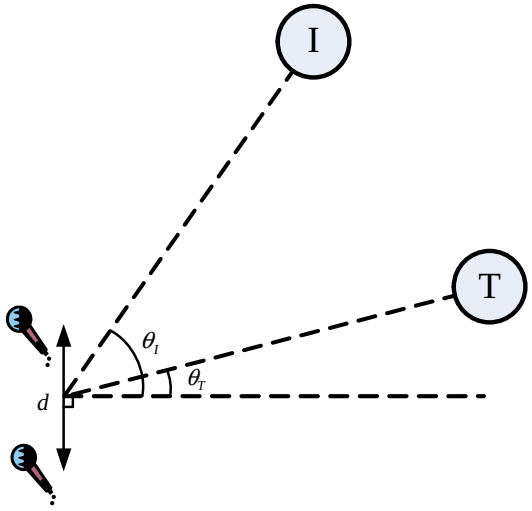


Fig. 9.4: The geometry when there is one target and one interfering source.

frames.

9.4.2 Gammatone channel weighting

As noted above, the estimates produced by Eq. (9.16b) are generally noisy and must be smoothed. To achieve smoothing along frequency, we use a gammatone weighting that functions in a similar fashion to that of the familiar triangular weighting in MFCC features. Specifically, we obtain the gammatone channel weighting coefficients $w[m, l]$ according to the following equation:

$$w[m,l] = \frac{\sum_{k=0}^{\frac{N}{2}-1} \mu[m,k] \left|\bar{X}[m;e^{j\omega k})H_l(e^{j\omega k})\right|}{\sum_{k=0}^{\frac{N}{2}-1} \left|\bar{X}[m;e^{j\omega k})H_l(e^{j\omega k})\right|} \tag{9.13}$$

where $\mu[m, k]$ is the original binary mask that is obtained using (9.16b). With this weghting we effectively map the ITD for each of the 256 original frequencies to an ITD for what we refer to as one of $L = 40$ channels. Each of these channels is associated with H_i, the frequency response of one of a set of gammatone filters with center frequencies distributed according to the Equivalent Rectangular Bandwidth (ERB) scale [5]. The final spectrum weighting is obtained using the gammatone mask μ_{cw}

$$\mu_{cw}[m,k] = \frac{\sum_{i=0}^{L-1} w[m,k] \left|H_l\left(e^{j\omega k}\right)\right|}{\sum_{i=0}^{L-1} \left|H_l\left(e^{j\omega k}\right)\right|} \tag{9.14}$$

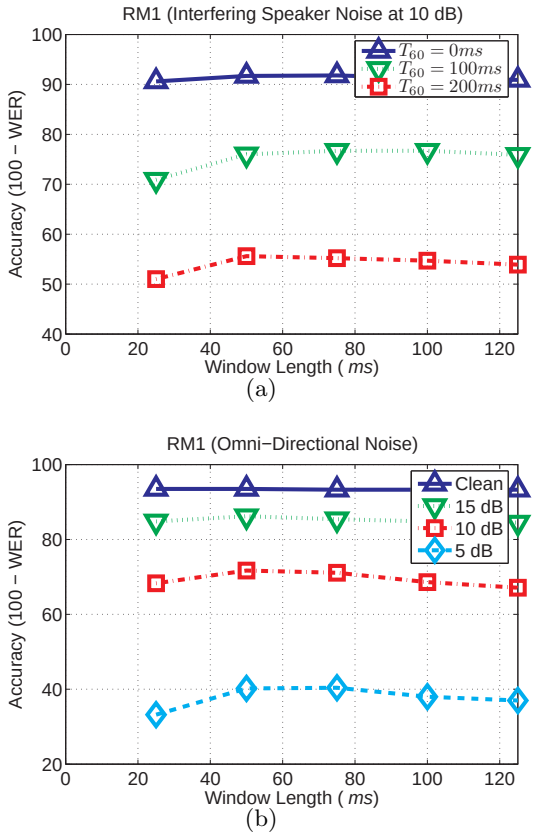


Fig. 9.5: The dependence of word recognition accuracy (100%-WER) on the window length under different conditions: (a) When there is an interference source at an angle of $\theta_I = 45°$. SIR is fixed at 10 dB. (b) When the target is corrupted by omni-directional natural noise, We used PD-FIXED with a threshold angle of $\theta_{TH} = 20°$

Examples of $w[m,k]$ and $\mu_{cw}[m,k]$ are shown shown for a typical spectrum in Fig. 9.7(e) and Fig. 9.7(f), respectively, with an SIR of 5 dB as before. The reconstructed spectrum is given by:

$$\tilde{X}[m,e^{j\omega k}) = \max\left(\mu_{cw}[m,k],\eta\right)\ \bar{X}[m,e^{j\omega k}) \tag{9.15}$$

where again we use $\eta = 0.01$ as in (9.16b).

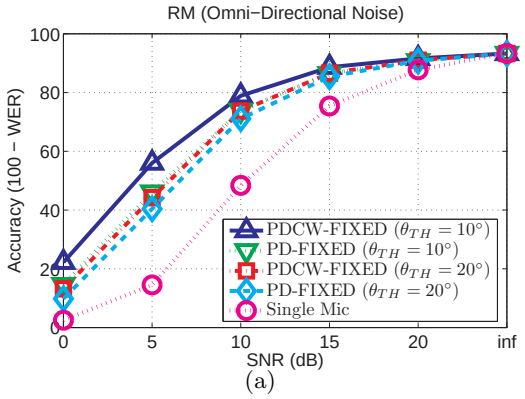


Fig. 9.6: The dependence of word recognition accuracy ($100\% - WER$) on the window length, using an SIR of 10 dB and various reverberation times. The filled symbols at 0 ms represent baseline results obtained with a single microphone.

9.5 Optimal ITD threshold selection from complementary masks

In the previous section, we used a fixed ITD threshold to construct binary masks. However, in the real-world environment, in many cases, we do not have control over noise sources. It is reasonable to assume that the optimal ITD threshold will be different depending on the noise source type and location. In this section, we discuss how to obtain an optimal threshold automatically without prior knowledge about the noise source. Before explaining our algorithm in very detail, we will show dependence of speech recognition accuracy on the interfering source location and the target location.

9.5.1 Dependence of speech recognition accuracy on the interfering source location and target angle

In Sec. 9.2, we presented experimental results using a fixed ITD threshold, which is empirically chosen. However, in real-world applications, we usually do not have control over noise sources. Thus, a specific threshold optimal for a particular environment might not be optimal for different environments.

To examine the dependence of the optimal threshold on interfering source location, let us consider the simulation configuration shown in Fig. 9.4. To simplify the discussion, we

assume that there is a single single interference source along the line of angle θ_I. As before, the distance between two microphones is 4 cm. In the first set of experiments, we assumed that the target angle θ_T is zero. For the interfering source angle θ_I, we used three different values (30°, 45°, and 75°). Signal-to-Interference Ratio (SIR) is assumed to be 0 dB and we assume that the room is anechoic. For speech recognition experiments, we used the configuration explained in Subsection 9.6.

In Fig. 9.8, we compare speech recognition results depending on the threshold angle θ_{TH} and the interference source angle θ_I. In Fig. 9.8(a), we used PD-FIXED, and in Fig. 9.8(b), we used PDCW-FIXED. When the interference source angle is θ_I, we obtain the best speech recognition accuracy when θ_{TH} is roughly $\theta_I/2$ or slightly larger than $\theta_I/2$. However, when θ_{TH} is larger than θ_I, the system failed to separate sound sources and it is reflected in very poor speech recognition accuracies. As an another set of experiment, we used natural omni-directional stereo noise, but we still assumed that the target angle $\theta_T = 0°$. The speech recognition result is shown in Fig. 9.9. We fixed the SNR level at 5 dB and changed θ_{TH}. In this experiment, the best speech recognition accuracy is obtained at a very small θ_T.

In the previous discussion, we observed that the optimal threshold angle $\hat{\theta}_{TH}$ heavily depends on noise source location. In the real environment, there is one more complication. Up to now, we have assumed that the target is placed at $\theta_T = 0°$. Even though we have control over the target, there might be still some errors in the target location. For example, even if a user of a hand-held device is instructed to use the device at this zero angle, the actual angle cannot be exactly zero. Thus, in another set of experiment, we used the configuration shown in Fig. 9.8, but we changed the target angles using five different values ($-20°$, $-10°$, 0°, 10°, and 20°). For the interference angle, we assumed that it is fixed at $\theta_I = 45°$. The experimental result is shown in Fig. 9.9. From this figure, we observe that if we choose a very small θ_{TH}, then the sound source separation system is not very robust against distortion in the target angle.

In this subsection, we observe that the optimal ITD threshold depends on both the target angle θ_T, the interference source angle θ_I, and the noise type. If the ITD threshold is inappropriate selected, then speech recognition accuracy is significantly degraded. From this observation, we conclude that we need to develop an automatic threshold selection algorithm which obtain a suitable ITD threshold without prior knowledge about prior noise source, and

at the same time, which is robust against error in the target angle θ_T.

9.5.2 Optimal ITD threshold algorithm

This algorithm is based on two complementary binary masks, one that identifies time-frequency components that are believed to belong to the target signal and the other that identifies the components that belong to the interfering signals (*i.e.* everything except the target signal). These masks are used to construct two different spectra corresponding to the power sequences representing the target and the interfering sources. We apply a compressive nonlinearity to these power sequences, and define the threshold to be the separating ITD threshold minimizes the cross-correlation between these two output sequences (after the nonlinearity).

Computation is performed in discrete fashion, considering a set T of a finite number of possible ITD candidates. We determine which element of this set is the most appropriate ITD threshold by performing an exhaustive search over the set T. Let us consider one element of this set, τ_0. We obtain the target mask and the complementary mask for τ_0 and for $0 \leq k \leq N/2$:

$$\mu_T[m,k] = \begin{cases} 1, & \text{if } |d[m,k]| \leq \tau_0 \\ \delta, & \text{otherwise} \end{cases} \tag{9.16a}$$

$$\mu_I[m,k] = \begin{cases} \delta, & \text{if } |d[m,k]| > \tau_0 \\ 1, & \text{otherwise} \end{cases} \tag{9.16b}$$

For $N/2 \leq k \leq N-1$, we use the following symmetry condition:

$$\mu_I[m,k] = \mu_T[m,N-k], \quad N/2 \leq k \leq N-1 \tag{9.17a}$$

$$\mu_I[m,k] = \mu_I[m,N-k], \quad N/2 \leq k \leq N-1 \tag{9.17b}$$

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 that 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 δ. The masks $\mu_T[m,k]$ and $\mu_I[m,k]$ in (9.16) are applied to $\bar{X}[m,e^{j\omega_k})$, the averaged signal spectrogram

from the two microphones:

$$\bar{X}[m, e^{j\omega_k}) \quad = \quad \frac{1}{2}\{X_L[m, e^{j\omega_k}) + X_R[m, e^{j\omega_k})\} \tag{9.18}$$

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:

$$X_T[m, e^{j\omega_k}|\tau_0) = \bar{X}[m, e^{j\omega_k})\mu_T[m, e^{j\omega_k}) \tag{9.19a}$$

$$X_I[m, e^{j\omega_k}|\tau_0) = \bar{X}[m, e^{j\omega_k})\mu_I[m, e^{j\omega_k}) \tag{9.19b}$$

In the above equation, we explicitly include τ_0 to show that the masked spectrum will depend on 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:

$$P_T[m|\tau_0) = \sum_{k=0}^{N-1} \left| X_T[m, e^{j\omega_k}) \right|^2 \tag{9.20a}$$

$$P_I[m|\tau_0) = \sum_{k=0}^{N-1} \left| X_I[m, e^{j\omega_k}) \right|^2 \tag{9.20b}$$

The compressive nonlinearity described above is invoked because the power signals in (9.20) have a very large dynamic range. A reasonable way of reducing dynamic range is by applying a compressive nonlinearity, which may be considered to represent the perceived loudness of the sound. While many nonlinearities have been proposed to characterize the relationship between signal intensity and perceived loudness [69] we chose the following power-law nonlinearity motivated by previous work (*e.g.*[52][32]):

$$R_T[m|\tau_0) = P_T[m|\tau_0)^{a_0} \tag{9.21a}$$

$$R_I[m|\tau_0) = P_I[m|\tau_0)^{a_0} \tag{9.21b}$$

where $a_0 = 1/15$ is the power coefficient as in [32].

The cross-correlation coefficient of the signals in (9.21) is obtained as follows:

$$\rho_{T,I}(\tau_0) = \frac{\frac{1}{N}\sum_{m=1}^{M} R_T[m|\tau_0)R_I[m|\tau_0) - \mu_{R_T}\mu_{R_I}}{\sigma_{R_T}\sigma_{R_I}} \tag{9.22}$$

where σ_{R_T} and σ_{R_I} are the standard deviations of $R_T[m|\tau_0)$ and $R_I[m|\tau_0)$, respectively, and μ_{R_1} and μ_{R_2} are the 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 cross-correlation

$$\hat{\tau}_1 = \arg\min_{\tau_0} |\rho_{T,I}(\tau_0)| \tag{9.23}$$

Even though cross-correlation is a very good measure to find an optimal ITD threshold when both of the target and interference sources are non-stationary, if the noise source is stationary, then cross correlation coefficient is not a good measure. This is especially true when the noise is stationary or semi-stationary like white noise or street noise.

To tackle this problem, we also use the normalized correlation coefficient as well.

$$r_{T,I}(\tau_0) = \frac{\frac{1}{N}\sum_{m=1}^{M} R_T[m|\tau_0)R_I[m|\tau_0)}{\sigma_{R_T}\sigma_{R_I}} \tag{9.24}$$

Another threshold τ_1 is obtained using the following equation:

$$\hat{\tau}_2 = \arg\min_{\tau_0} |r_{T,I}(\tau_0)| \tag{9.25}$$

The final ITD threshold τ is obtained as a smaller value between τ_0 and τ_1.

9.6 Experimental results

In this section we present experimental results using the PDCW-AUTO algorithm described in this paper. We compare PDCW-AUTO with PDCW-FIXED. To examine the effectiveness of channel weighting explained in Subsec. 9.4.2 , we also compare PDCW-AUTO with PD-AUTO. To compare our algorithm with a conventional approach, we repeated using ZCAE under the same environment. ZCAE refers to the time-domain algorithm described in [16] with binary masking, since the better-performing continuous-masking requires that there should be only one interfering source and that we should know the location of that source. [I WILL REDO ZCAE EXPERIMENTS LATER]. In all speech recognition experiments, we used `sphinx_fe` included in `sphinxbase 0.4.1` for speech feature extraction, `SphinxTrain 1.0` for acoustic model training, and `Sphinx 3.8` for decoding, all of which are readily available in Open Source form [70]. We used subsets of 1600 utterances and 600 utterances, respectively, from the DARPA Resource Management (RM1) database for training and testing. In our experiments, the two microphones are placed 4 cm from one another. We conducted three different types of experiments. To simulate reverberation effects we used the `Room Impulse Response (RIR)` software [50]. In Subsec. 9.6.1, we assumed that there is one interference speaker. In Subsec. 9.6.2, we assumed that there are three interference speakers at random location. In Subsec. 9.6.3, we used natural noise recorded using a real stereo-microphone hardware.

9.6.1 Experimental results when there is a single interference source

In the experiments in this subsection, we assumed a room of dimensions 5 x 4 x 3 m, with microphones that are located at the center of the room. Fig. 9.4 illustrates the condition where there is one interfering speaker. Both the target and interfering sources are 1.5 m away from the microphone For the fixed-ITD threshold system such as PDCW-FIXED, we used the threshold angle $\theta_{TH} = 20^\circ$ based on our experimental results shown in Subsec. 9.5.1.

We conducted two different sets of experiments. In the first set of the experiments, we assume that the target is located along the perpendicular bisector of the line between two microphones, which means $\theta_T = 0^\circ$. We assume that the interfering source is located at $\theta_T = 30^\circ$. We repeated experiments by changing SIR level and reverberation time. As shown in Fig. 9.11(a), in the absence of reverberation at 0-dB SIR, both the fixed ITD system and the automatic-ITD system show comparable performance. If the reverberation occurs, however, the automatic-ITD system provides substantially better performance than the fixed-ITD signal separation system.

In the second set of the experiments, we changed the location of the interfering speaker while maintaining the SIR at 0 dB. As shown in Fig. 9.12, even if the SIR is the same as in the calibration environment, the fixed-ITD threshold system shows significantly degraded performance if the actual interfering speaker location is different from the location used in the calibration environment. The automatic-ITD-threshold selection system provides recognition results that are much more robust with respect to the locations of the interfering sources. In this figure we observe that as the interfering speaker moves toward the target, the fixed-ITD threshold PD system shows increased word error rate. We repeated the same experiment by changing the reverberation time. As shown in Fig. 9.12(a) to Fig. 9.12(d), the automatic-threshold-selection algorithm provides consistently better recognition accuracy than the fixed threshold system, as expected.

Fig. **??** compares word recognition accuracy for several of the algorithms discussed in the paper. 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.

9.6.2 Experimental results when there are three randomly-positioned interfering speakers

9.6.3 Experimental results for natural noise

In our third 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. 9.15 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

9.7 Conclusions

In this work, we present a speech separation algorithm, PDCW, based on ITD that is inferred from phase information. The algorithm uses gammatone weighting and longer analysis windows. This algorithm is quite computationally efficient and shows significant improvement in recognition accuracy under practical environmental conditions of noise and reverberation. We also presented 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. When PDCW is combined with this automatic ITD threshold selection algorithm, we refer this algorithm to PDCW-AUTO. PDCW-AUTO shows much more robustness than PDCW-FIXED which uses an fixed ITD threshold empirically obtained.

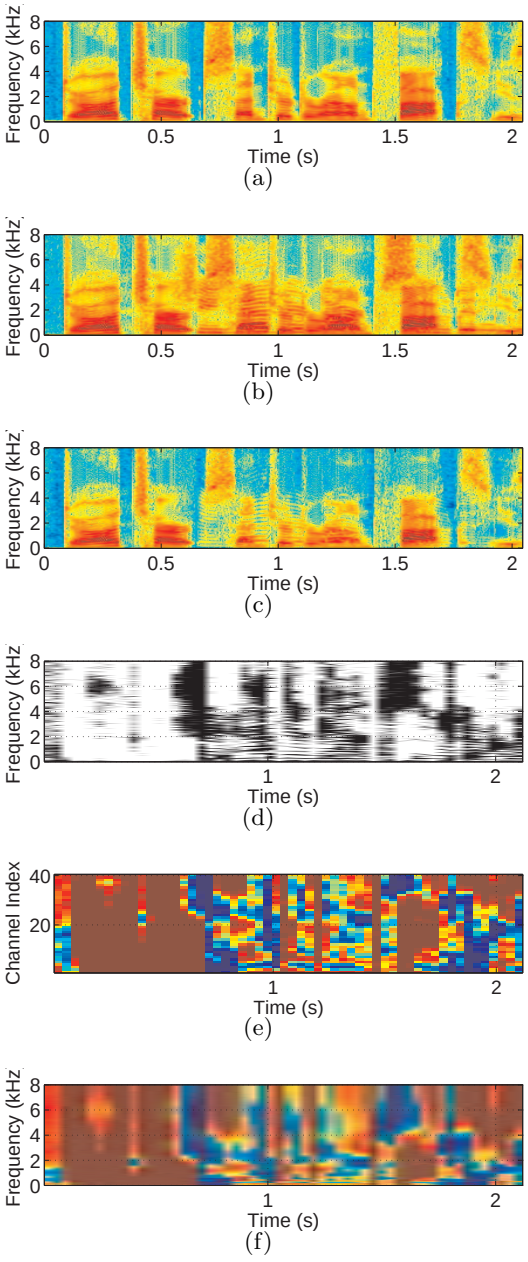
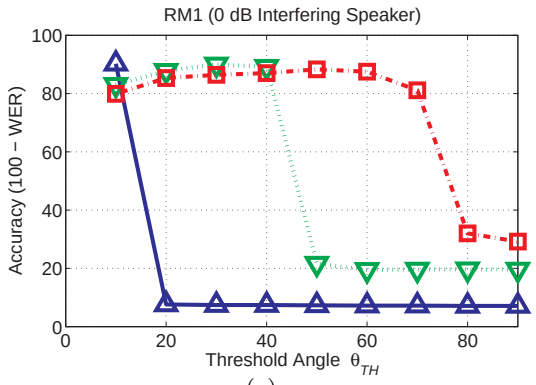
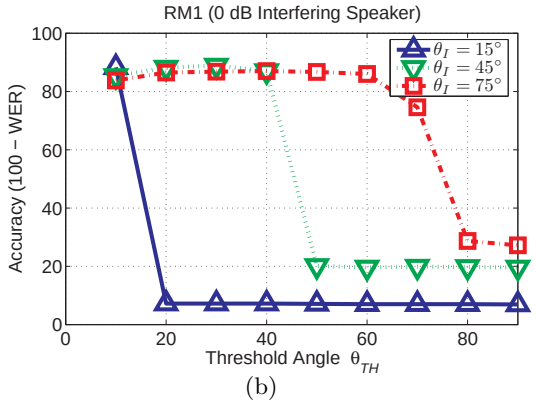


Fig. 9.7: Sample spectrograms illustrating the effects of PDCW processing. (a) original clean speech, (b) noise-corrupted speech, (c) reconstructed (enhanced) speech (d) the time-frequency mask obtained with (9.16b) (e) gammatone channel weighting obtained from the time-frequency mask in (9.13) (e) final frequency weighting shown in (9.14) (f) enhanced speech spectrogram using the entire PDCW algorithm



(a)



(b)

Fig. 9.8: The dependence of word recognition accuracy (100% - WER) on the threshold angle θ_{TH} and the location of the interfering source θ_I. The target is assumed to be located along the perpendicular bisector of the line between two microphones ($\theta_T = 0°$). (a) when PD-FIXED is used. (b) when PDCW-FIXED is used.

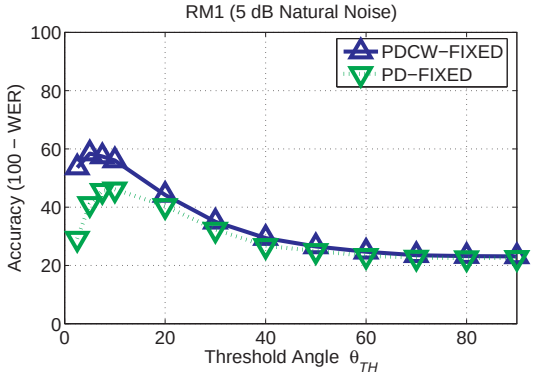


Fig. 9.9: The dependence of word recognition accuracy (100% - WER) on the threshold angle θ_{TH} in the presence of omni-directional natural noise. The target is assumed to be located along the perpendicular bisector of the line between two microphones ($\theta_T = 0°$).

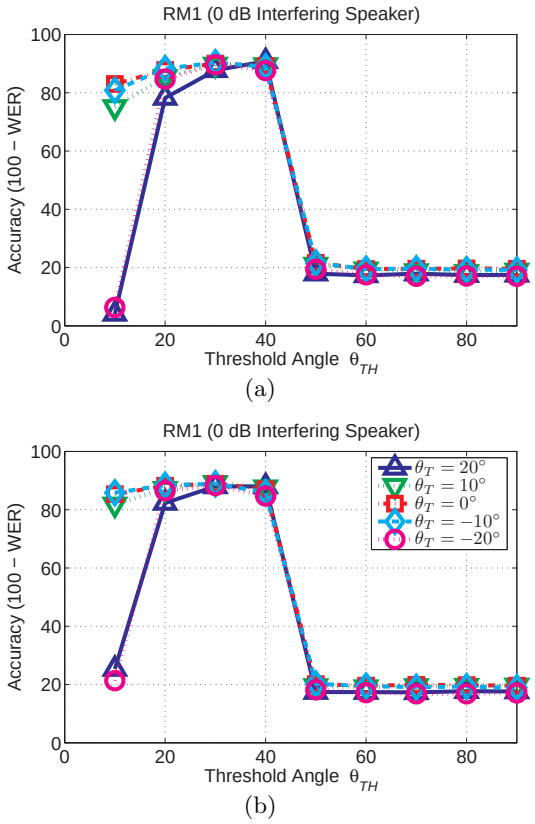


Fig. 9.10: The dependence on word recognition accuracy (100% - WER) on the threshold angle θ_{TH} and the location of the target source θ_T. (a) when PD-FIXED is used and (b) when PDCW-FIXED is used.

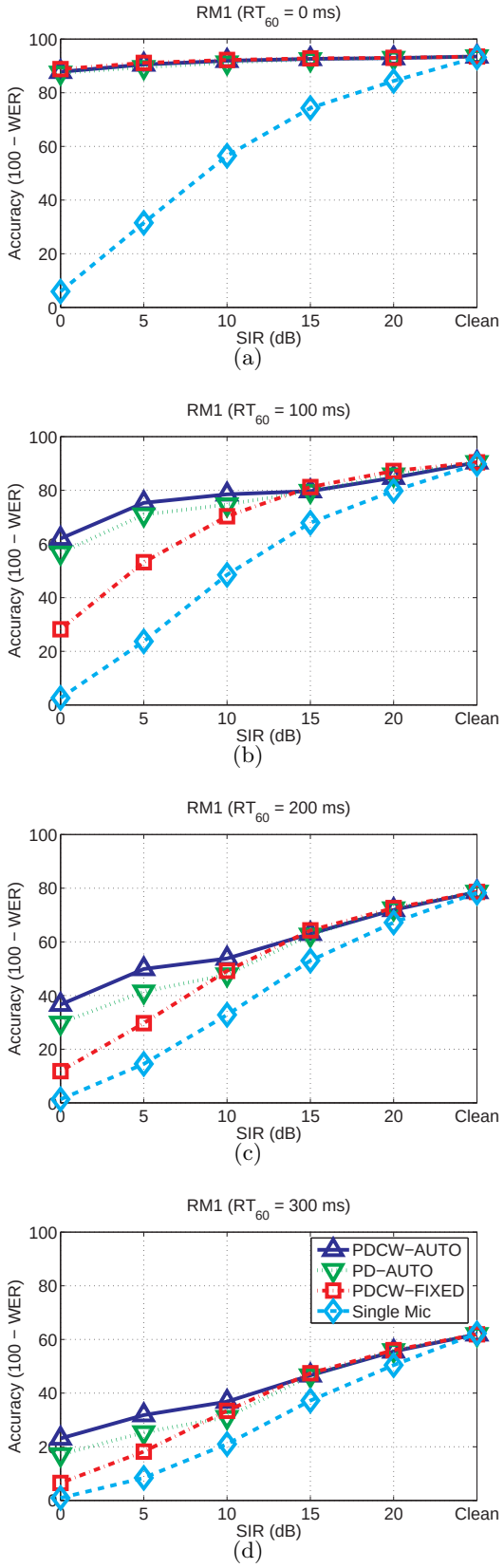


Fig. 9.11: Comparison of recognition accuracy for the DARPA RM database corrupted by three randomly placed speakers at different reverberation times (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms

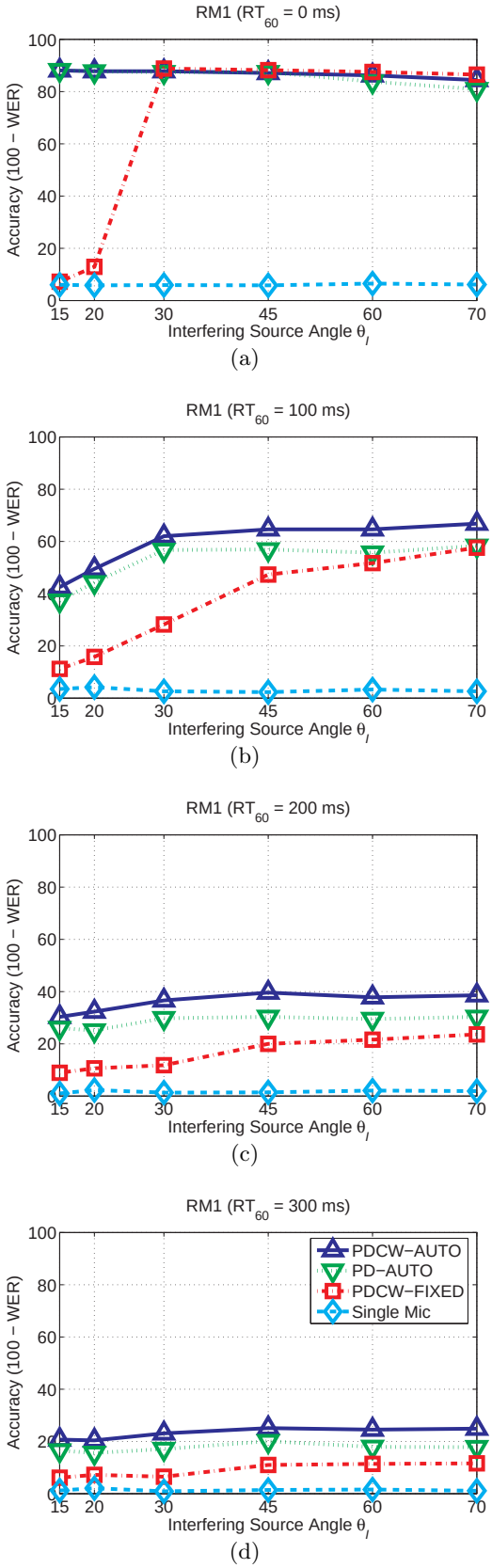
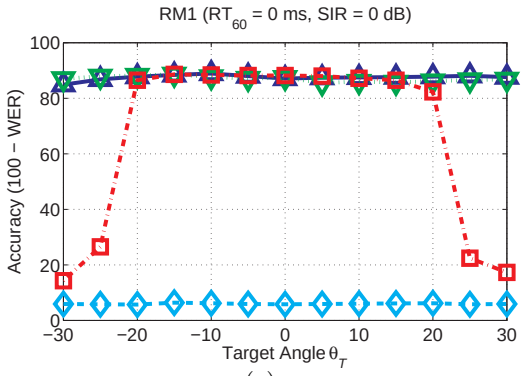


Fig. 9.12: Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at different locations at different reverberation time (a) 0 ms (b)



(a)

Fig. 9.13: Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at 45 degrees ($\theta_I = 45^\circ$) in an anechoic room. SIR level is 0 dB. Target angle is changed from $\theta_T = -30^\circ$ to $\theta_T = 30^\circ$

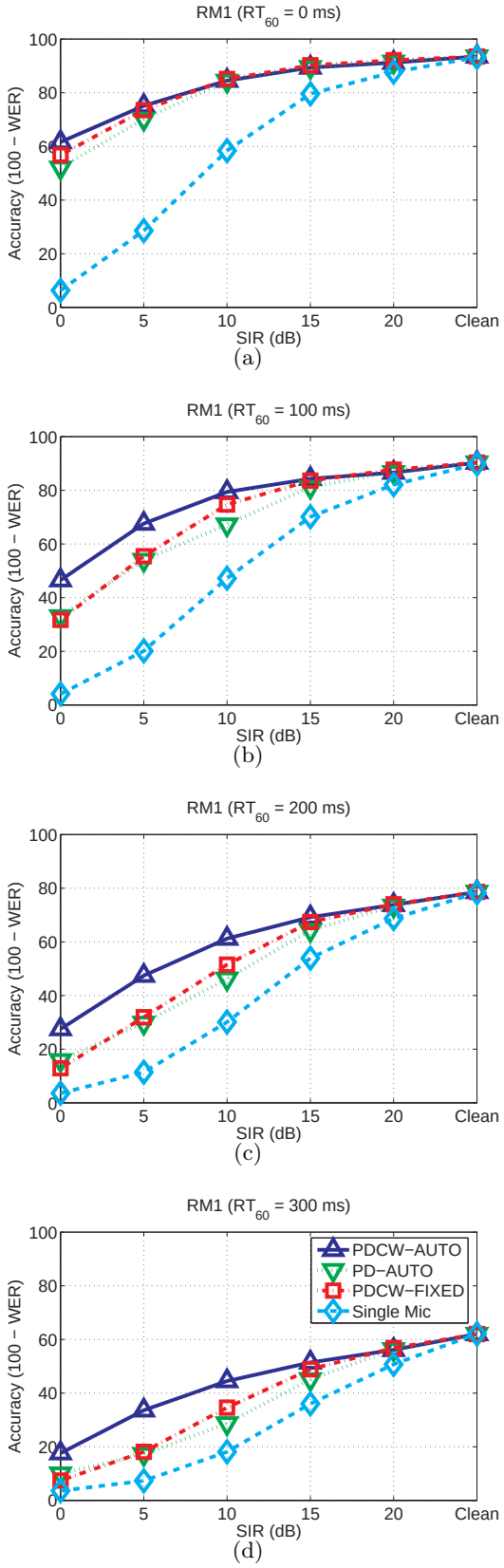


Fig. 9.14: Comparison of recognition accuracy for the DARPA RM database corrupted by an interference speaker located at 30 degrees at different reverberation times (a) 0 ms (b) 100 ms (c) 200 ms (d) 300 ms

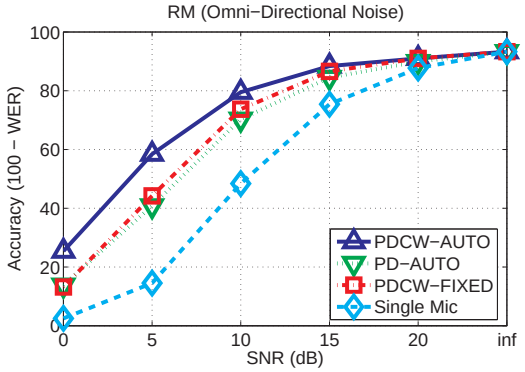


Fig. 9.15: Speech recognition accuracy using different algorithms (a) in the presence of an interfering speech source as a function of SNR in the absence of reverberation, (b,c) in the presence of reverberation and speech interference, as indicated, and (d) in the presence of natural real-world noise.

10. CONCLUSION

10.0.1 Introduction

In this thesis, we have sought to improve speech recognition accuracy in noisy environment using techniques motivated by auditory processing. Our goal in this thesis is to enhance robustness especially in more difficult environments such as non-stationary noise environment, reverberation, or interfering speaker using techniques motivated by auditory processing.

After the introduction of Hidden Markov Model (HMM), speech recognition accuracy has significantly improved. However, if the test environment is different from the training environment, then speech recognition accuracy is seriously degraded. Conventional approaches for enhancing robustness against environmental mismatch are usually based on statistical feature normalization. For example, we usually assume that the mean of each element of features is the same for all utterances. We can make a similar assumption for variance as well as mean. Cepstral Mean Normalization (CMN) and Mean Variance Normalization(MVN) are based on these assumptions, respectively. Alternatively, we can assume that the histogram is the same for all utterances. As mentioned in Chap. 2, these techniques are somewhat sensitive to silence length in each utterance, but if they are combined with a reliable Voice Activity Detection (VAD), then they usually show significant performance improvement especially for stationary noise. In the other kind of approaches, we obtain a statistical model (usually represented by a Gaussian Mixture Model (GMM)) of log spectra or features obtained from a clean training set. We set up a mathematical equation to represent the effect of noise and/or reverberation. In this mathematical model, the effect of noise is called an environmental function. Using the above mentioned statistical model obtained from training data, we obtain the environmental function, then we can reverse the effect of noise. These kinds of approaches are also successful for stationary noise, but they do not show substantial improvements in non-stationary noise or reverberation.

In this thesis, we first try to understand why human auditory systems show remarkable ability even in non-stationary noise or under reverberation. We especially focus on the characteristics of nonlinearity, temporal masking, precedence effect, binaural hearing, temporal resolution, and modulation filtering effect.

10.0.2 Summary of Findings and Contributions of This Thesis

we find that there are many auditory phenomena, which have not been employed at all or inefficiently employed in conventional feature extraction or noise compensation algorithms. Some examples include rate-intensity nonlinearity, on-set enhancement, and temporal masking. Our conclusion is that if we use a more faithful model of human auditory phenomena, then we can obtain improvements under unmatched conditions. However, exact modelling of human auditory system is prohibitive, since it is too complicated and less practical. Thus, in our work, we tried to make a simple mathematical model motivated by human auditory processing. Our objective is building simple models which can be useful for "real applications", so we also put emphasis on on-line processing and computational cost as well. Our contribution is summarized in this Subsection.

First, we observe the logarithmic nonlinearity employed in MFCC is not very robust against additive noise. The reason is that the logarithmic nonlinearity does not care about the auditory threshold as explained in Chap. 5. If power in certain time-frequency bins is below the auditory threshold level, for human listeners, it is just silence regardless of actual power level. However, in case of the logarithmic nonlinearity, power difference in inaudible range (power below the threshold level) is considered. Especially, if power in a certain time-frequency bin approaches zero, then nonlinearity output approaches the negative infinity. For these small power region, even for a very small change in the input power level, there is a very big change in the nonlinearity output, which results in vulnerability for additive noise.

The human auditory nonlinearity is free from this problem, since it exhibits a threshold behavior. However, these curves are very complicated and highly nonlinear, thus they are not very suitable for automatic speech recognition applications. So, we tried to use a simplified model, which is based on power function. From the MMSE approximation to the human

rate-intensity curve, we obtained a power coefficient between 1/10 and 1/15. As shown in experimental results in Chap. 5, this range also shows good balance in terms of trade offs between the clean performance and the noise robustness. In our Power Normalized Cepstral Coefficient (PNCC), we adopt a power coefficient of 1/15. In case of PNCC, as shown in Chap. 8 it is doing at least as good as MFCC even for clean, while obtaining much better performance than MFCC under noisy environment.

Small Power Boosting (SPB) is another approach based on this observation. As mentioned above, for these small power region, even for a very small change in the input power level, there is a very big change in the nonlinearity output. If this is the case, then we can enhance robustness by removing all small power region systematically. SPB approach results in slight degradation for clean speech, but it shows very good performance especially for music noise.

Second, we observe that the on-set enhancement plays an important role for enhancing robustness especially for reverberation. This is also highly related to the actual human auditory nerve response. If we observe the rate-intensity response, then we observe that the on-set rate is much higher than the sustained rate. Additionally, unlike the sustained rate, on-set rate does not show a saturation behavior. In the reverberant environment, the effect of reverberation usually less affect the onset portion but more affects the trailing portion. Thus, we can understand that emphasizing the onset and suppressing the trailing portion is helpful for reverberant environment. This idea is realized as SupprSSF.

In some sense, temporal masking is similar to on-set enhancement, but they are not exactly the same. In case of on-set enhancement, we relatively emphasize the onset portion (or relatively de-emphasizes the trailing portion). In case of temporal masking, we think about some kinds of masking region, so after some sufficiently large peak in sound pressure, we cannot perceive smaller peaks. We also designed a simple mathematical model for achieving this objective. Applying temporal masking also showed improvements for reverberation and interfering speaker noise.

Thirdly, we observed that temporal resolution plays an important role in robust speech recognition. For human listeners, it has been well known that we largely ignore slowly varying spectral components. If we try to remove such a slowly varying component, then it is better to use a longer window than a short window, which has been usually used in speech

analysis. The issue is to model slowly varying noisy components, we need to use a longer window, but for speech analysis, we still need to use a shorter window. To tackle this problem, in our approach, we proposed a two-stage window system. It was realized either using Medium-duration Analysis Synthesis (MAS) approach or Medium-duration. This two stage window length system has been incorporated in many algorithms such as Power Normalized Cepstral Coefficient (PNCC), Power-function-based Power Distribution Normalization (PPDN), Phase Difference Channel Weighting (PDCW), etc.

As mentioned in the above paragraph, human auditory systems pay less attention to slowly varying component. Motivated by this, researchers have developed many different types of modulation filtering approach. In our work, we propose a new technique based on asymmetric filtering. Compared to many existing techniques, our approach has the following characteristics. First, we use medium-duration window for obtaining better temporal resolution, as mentioned in the previous paragraph. Second, we apply the filtering before nonlinearity, since the additive noise can be more easily removed. Third, unlike conventional filtering, the filter output contour tracks the low level envelope, which is especially useful for estimating slowly varying noise component.

For human auditory systems, due to binaural hearing, we can obtain cues regarding where sound source is located. Motivated by this, we developed an efficient sound source separation algorithm called PDCW, which is explained in Chap. 9. In this approach, we calculate ITD from phase difference. We observe a smoothed weighting scheme using the gammatone frequency response is much more effective than using binary masks for each frequency index.

In the PDCW approach, we still do not know what would be the optimal threshold to select the target. However, human has an ability to automatically clustering sound from different direction and selecting only the target. Motivated by this, we proposed a ITD threshold selection algorithm using the cross correlation of nonlinearity-applied power from the target and everything except the target. This approach, combined with PDCW, is called PDCW-AUTO. We discussed this approach in very detail in Chap. 9.

10.0.3 Directions for Further Research

In this thesis, we explored various We considered various aspects of human auditory processing, but still there are remaining things to be investigated into. First, we haven't explored into masks other than temporal masking. We need to consider effects like two tone suppression and other masking effects.

Second, we calculated the ITD in the frequency domain. We also need to compare this approach with correlation-based approach, which is closer to the actual human auditory processing in more detail.

Third, in our work here, we used a fixed nonlinearity, but we think variable nonlinearity might be also an interesting idea. Additionally, time-varying filter bandwidth (which is used in Zhang and Carney's model) might be also an interesting idea to try.

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] B. C. J. Moore and B. R. Glasberg, "A revision of Zwicker's loudness model," *Acustica - Acta Acustica*, vol. 82, no. 2, pp. 335–345, 1996.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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*, Apr. 1997, pp. 851–854.

[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, no. 11, pp. 1486–1501, Nov. 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 Arrys, 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. Volkmann, 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] H. Hermansky, "Perceptual linear prediction analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[23] A. V. Oppenheim and R. W.Scafer, with .J. R. Buck, *Discrete-time Signal Processing*. Englewood-Cliffs, NJ: Prentice-Hall, 1999.

[24] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood-Cliffs, NJ: Prentice-Hall, 1978.

[25] S. S. Stevens, "On the psychophysical law," *Psychological Review*, vol. 64, no. 3, pp. 153–181, 1957.

[26] 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.

[27] 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/

[28] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques or robust speech recognition," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 1995, pp. 153–156.

[29] B. E. D. Kingsbury, N. Morgan, and, S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, vol. 25, no. 1-3, pp. 117–132, Aug. 1998.

[30] 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.

[31] ——, "Effect of reducing slow temporal modulations on speech recognition," *J. Acoust. Soc. Am.*, vol. 95, no. 5, pp. 2670–2680, May 1994.

[32] 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.

[33] ——, "Power function-based power distribution normalization algorithm for robust speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Dec. 2009, pp. 188–193.

[34] 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.

[35] 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.

[36] 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.

[37] M. C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky , P. Jain, S. Kajarekar, N. Morgan, S. Sivadas, "Robust asr front-end using spectral-based and discriminant features: experiments on the aurora tasks," in *EUROSPEECH-2001*, Sept. 2001, pp. 429–432.

[38] 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.

[39] 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.

[40] 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.

[41] J. W. Strutt (Lord Rayleigh), "On our perception of sound direction," *Philosophical Magazine*, vol. 13, pp. 214–232, 1907.

[42] 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, pp. 2495–2498.

[43] M. Slaney, "Auditory Toolbox Version 2," *Interval Research Corporation Technical Report*, vol. 1998, no. 10, 1998. [Online]. Available: http://cobweb.ecn.purdue.edu/malcolm/interval/1998-010/

[44] D. M. Green, *An Introduction to Hearing, 6th edition.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., Publishers, 1976.

[45] 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.

[46] X. Zhang, M. G. Heinz, 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.

[47] 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.

[48] H. Hermansky, "Perceptual linear prediction analysis of speech," *J. Acoust. Soc. Am.*, vol. 87(4), no. 4, pp. 1738–1752, Apr. 1990.

[49] S. S. Stevens, "On the psychophysical law," *Psychological Review*, vol. 64, no. 3, pp. 153–181, 1957.

[50] S. G. McGovern, "A model for room acoustics," http://2pi.us/rir.html.

[51] 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.

[52] 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, pp. 28–31.

[53] 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.

[54] 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.

[55] 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.

[56] P. M. Zurek, *The precedence effect*. New York, NY: Springer-Verlag, 1987, ch. 4, pp. 85–105.

[57] K. D. Martin, "Echo suppression in a computational model of the precedence effect," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 1997.

[58] Y. Park and H. Park, "Non-stationary sound source localization based on zero crossings with the detection of onset intervals," *IEICE Electronics Express*, vol. 5, no. 24, pp. 1054–1060, 2008.

[59] L. R. Rabiner and B. -H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: PTR Prentice Hall, 1993.

[60] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[61] C. Kim and R. M. Stern, "Nonlinear enhancement of onset for robust speech recognition," in *INTERSPEECH-2010*, Sept. 2010 (accepted).

[62] C. Lemyre, M. Jelinek, and R. Lefebvre, "New approach to voiced onset detection in speech signal and its application for frame error concealment," in *IEEE Int. Conf. Acoust. Speech, and Signal Processing*, May. 2008, pp. 4757–4760.

[63] S. R. M. Prasanna and P. Krishnamoorthy, "Vowel onset point detection using source, spectral peaks, and modulation spectrum energies," *IEEE Trans. Audio, Speech, and Lang. Process*, vol. 17, no. 4, pp. 556–565, May 2009.

[64] Y.-H. Chiu and R. M. Stern, "Analysis of physiologically-motivated signal processing for robust speech recognition," in *INTERSPEECH-08*, Sept. 2008, pp. 1000–1003.

[65] 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.

[66] 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.

[67] W. Grantham, "Spatial hearing and related phenomena," in *Hearing*, B. C. J. Moore, Ed. Academic, 1995, pp. 297–345.

[68] D. Halupka, S. A. Rabi, P. Aarabi, and A. Sheikholeslami, "Real-time dual-microphone speech enhancement using field programmable gate arrays," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2005, pp. 149 – 152.

[69] D. M. Green, *An Introduction to Hearing, 6th edition*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., Publishers, 1976.

[70] (2010) CMU Sphinx (Open Source Toolkit for Speech Recognition). [Online]. Available: http://cmusphinx.sourceforge.net/