Phase Spectrum Based Speech Processing and Spectral Energy Estimation for Robust Speech Recognition

Phase Spectrum Based Speech Processing and Spectral Energy Estimation for Robust Speech Recognition

Anthony Stark

BEng (Hons), BInfTech

Griffith School of Engineering

Science, Environment, Engineering and Technology

Griffith University

Submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

June 2010

Statement of originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by any other person except where due reference is made in the thesis itself.

Anthony Stark

Student number: 2094545
Date of Submission: 16th June 2010
Principal Supervisor: Professor Kuldip Paliwal
Associate Supervisor: Doctor Stephen So

Acknowledgements

First, I would like to thank my thesis supervisor Kuldip Paliwal. I am extremely grateful for the guidance and feedback that made this thesis possible. I would also like to extend my gratitude to the other members of the Griffith University Signal Processing Laboratory; particularly James Lyons, Kamil Wójcicki and Stephen So, for the assistance provided throughout my dissertation research. Last but not least, I would like to thank those close to me who provided encouragement and support through my years of study.

Abstract

Speech is the dominant mode of communication between humans; simple to learn, easy to use and integral for modern life. Given the importance of speech, development of a human-machine speech interface has been greatly anticipated. This challenging task is encapsulated in the digital speech processing research field. In this dissertation, two specific areas of research are considered: 1) the use of short-time Fourier spectral phase in digital speech processing and 2) use of the minimum mean square error spectral energy estimator for environment-robust automatic speech recognition.

In speech processing and modelling, the short-time Fourier spectral phase has been considered of minor importance. This is because classic psychoacoustic experiments have shown speech intelligibility to be closely related to short-time Fourier spectral magnitude. Given this result, it is unsurprising that the majority of speech processing literature has involved exploitation of the short-time magnitude spectrum. Despite this, recent studies have shown useful information can be extracted from the spectral phase of speech. As a result, it is now known that spectral phase possesses much of the same intelligibility information as spectral magnitude. It is this avenue of research that is explored in greater detail within this dissertation. In particular, we investigate two phase derived quantities – the short-time instantaneous frequency spectrum and the short-time group delay spectrum. The properties of both spectra are investigated mathematically and empirically, identifying the relationship between known speech features and the underlying phase spectrum. We continue the investigation by examining two related quantities – the instantaneous frequency deviation and the group delay deviation. As a result of this research, two novel phase-based spectral representations are proposed, both of which show a high degree information applicable to speech processing.

In the second part of this dissertation, the problem of robust automatic speech recognition (ASR) is considered. Contributions are focused on the problem of estimating speech features that are robust to additive noise. Specifically, we investigate the use of the minimum mean square error spectral energy estimator.

The spectral energy estimator is a simple and efficient noise suppressor. However, it often performs poorly when used for environment-robust ASR. To remedy this, two methods are investigated. In the first method, we investigate a set of simple heuristic modifications to the basic spectral energy estimator. One heuristic adaptation uses the speech presence uncertainty framework, while another adaptation considers a direct noise suppression modification. Both heuristic adaptations are shown to overcome the limitations of the spectral energy estimator, allowing for effective, efficient and flexible noise suppression. For the second method, we investigate the direct stochastic estimation of the Mel-frequency cepstral coefficient (MFCC) feature vector. Since a stochastic estimator is only as useful as the statistical assumptions that underpin it, we start with a well known spectral-domain noise distortion model. Using this framework, statistical models for estimating spectral energies have previously been derived. In the present work, we detail the mathematical transformations required to transform the spectral energy models into filterbank, log-filterbank and cepstral domain noise distortion models. Using this cepstral framework, we present a novel MFCC estimator. The proposed MFCC estimators are evaluated on the Aurora2 and RM speech recognition tasks. Results show significant improvement in robustness over both baseline results and several contemporary noise suppression algorithms.

Contents

List of Figures

List of Tables

List of Acronyms

AMS	Analysis-modification-synthesis (framework)
ASR	Automatic speech recognition
CDF	Cumulative (probability) distribution function
CMS	Cepstral mean subtraction
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DSTFT	Discrete short-time Fourier transform
GD	Group delay
GMM	Gaussian mixture model
GS	Gerchberg-Saxton (algorithm)
HMM	Hidden Markov model
IAGDD	Inverse absolute group delay deviation
IAIFD	Inverse absolute instantaneous frequency deviation
IF	Instantaneous frequency
IID	Independent and identically distributed
LSA	Short-time MMSE log-spectral amplitude filter
MAP	Maximum a posteriori
MFCC	Mel-frequency cepstral coefficient
ML	Maximum likelihood
MMSE	Minimum mean square error
PESQ	Perceptual evaluation of speech quality (score)
PDF	Probability density function
RV	Random variable
SA	Short-time MMSE spectral amplitude filter
SE	Short time MMSE spectral energy filter
SNR	Signal to noise ratio
SW	Spectral Wiener filter
VTS	Vector Taylor series
WER	Word error rate

Chapter 1

Introduction

Spoken language is the primary form of communication between humans; it is fast, simple and intuitive. It is learned from an early age and is critical for functioning within society. Given the importance of speech, the development of a human-machine speech interface has long been an engineering goal. The demand for such an interface has grown considerably in recent years, due to the increasingly prevalent and integrated nature of modern electronic technology. Unfortunately, the ease with which humans produce and process spoken language belies a deceptively difficult engineering problem. As a result, the field of digital speech processing has enjoyed considerable research attention in the past few years.

Digital speech processing is a broad field that encompasses several overlapping research topics. This includes speech modification, speech enhancement, speech synthesis, speech recognition and voice biometrics. In this dissertation two specific areas of research are investigated: 1) the use of short-time Fourier spectral phase in speech processing and 2) use of the minimum mean square error spectral energy estimator in environment-robust automatic speech recognition. This dissertation has been structured into the two aforementioned parts, both of which are summarized below.

1.1 Research Part I: Use of the short-time Fourier phase spectrum for speech processing

In Part I of this thesis, the role of short-time Fourier phase spectra in speech processing is investigated. Currently, the vast majority of speech processing algorithms operate on the short-time Fourier magnitude spectrum. While short-time spectral phase is implicitly required to produce natural sounding speech, it is considered of minor importance in terms of speech intelligibility. In addition to this, the phase spectrum suffers from many tractability and signal processing issues. As a result, it is often overlooked in many speech applications.

However, spectral magnitude techniques have become relatively mature. Because of this, the use of short-time phase spectra is being revisited by several researchers. While current research in this area has not been able to topple the primacy of the short-time magnitude spectrum, it has challenged the belief that short-time spectral phase lacks useful speech information. Encouraged by these results, this dissertation investigates several specific areas of the short-time phase spectrum. In particular, the short-time instantaneous frequency spectrum (time derivative of the short-time phase spectrum) and short-time group delay spectrum (frequency derivative of the short-time phase spectrum) are mathematically and empirically characterized for speech signals. Particular focus is given to measuring how far the instantaneous frequency (and corresponding group delay) deviate from expected centre frequencies. It was found that these deviation measurements contained many useful characteristics, reflecting a wide degree of speech information. As a result, it was shown that the short-time phase spectrum contains many characteristics useful for speech processing applications. In addition to investigating the phase spectrum derivatives, several other research avenues were pursued. This includes a brief study on the direct mathematical relationships between the phase and magnitude spectra, as well as a speech enhancement algorithm based on phase spectrum manipulation. A chapter by chapter overview of specific research is given in Section 1.3.

1.2 Research Part II: Environment-robust estimation of Mel-frequency cepstral coefficients

Part II of this thesis aims to contribute to the field of automatic speech recognition (ASR). ASR is the process by which the acoustic signal (of spoken language) is processed and transcribed into a word sequence. The primary goal is to convert natural language into a parsed formant machines can understand. A brief listing of current and potential applications of ASR is given below:

- Voice enabled portals for command and control.
- Systems for the disabled.
- Automatic dictation.
- Automatic translation between languages.

Currently, ASR exists in many telephony systems, consumer grade computers and portable electronic devices. The goal of these systems, is to provide a fast, reliable and natural interface to an underlying electronic system. However, several large challenges remain that are hindering adoption of the technology. Of these challenges, increasing robustness and reliability is seen to be one of the most important.

Current state-of-the-art ASR can exhibit extremely good recognition accuracy in the laboratory setting. Unfortunately, this isn't the case for real-world deployment settings such as office environments, a busy warehouse or in a car. In these environments, it is difficult to maintain acceptable recognition accuracy due to interfering background noise. Recognition accuracy is itself, negatively impacted by any variation in the acoustic environment. As well as direct acoustic interference, variations also arise from the use of different recording equipment, differing speaking styles and even different speakers. Taken together, even mild variation can reduce recognition accuracy to unacceptable levels. Since variability is often impossible to eliminate from the operating environment, increasing ASR robustness is extremely important.

Contributions in this thesis are focused on the problem of estimating 'noise-free' speech features from speech that has been corrupted with additive background noise. There are many ways in which such an estimate may be performed. However, the quality

of these estimators is highly dependent on the statistical assumptions that underpin them. To avoid the problem of high-level statistical assumptions, we investigate the estimation problem from a fundamental statistical viewpoint. In particular, we begin with a well known spectral noise distortion model. Such a model has been used previously to derive many popular enhancement algorithms, including the spectral Wiener filter, spectral amplitude filter and log-spectral amplitude filter. However, these algorithms were originally designed to improve the subjective listening quality of an acoustic signal. Because the objectives of subjective human listening and machine recognition are not the same, several mathematical and heuristic design changes are investigated. In particular, we investigate the use of the spectral energy estimator for deriving the Mel-frequency cepstral coefficient (MFCC) speech feature set. Two specific modifications to the spectral energy estimator are examined in this thesis: 1) a set of heuristic adaptations and 2) a mathematical conversion to enable direct stochastic estimation of the MFCC vector. The proposed estimators are evaluated on the Aurora2, RM and OLLO speech databases and can be shown to significantly improve additive noise robustness. A more detailed overview of the proposed estimators is given in Section 1.3.

1.3 Thesis chapter overview

- **Chapter 2** provides a general literature review of the digital speech processing field. Here the basic properties of the human speech production and perception systems are explained. We continue by describing the speech system from a signal processing point of view, including the broad engineering requirements needed for its analysis and manipulation. Finally, we provide a short synopsis regarding the role of short-time phase spectra in speech processing.

- **Chapter 3** is a literature review of ASR. Here we discuss the main components required for building a generic state-of-the-art ASR system. In addition to this, we also discuss several weaknesses of current ASR – including the degradation in recognition performance caused by additive background noise. Current methods in the literature aimed at tackling this problem are also summarized.

Part I: Investigation of short-time spectral phase for use in speech processing

- **Chapter 4** examines the relationship between the short-time magnitude and short-time phase spectra from a mathematical perspective. Here methods for regenerating one spectrum from the other are investigated. In addition, a more efficient algorithm for the phase spectrum regeneration problem is proposed and tested.

- **Chapter 5** describes research into the short-time instantaneous frequency (IF) spectrum. Mathematical properties of the IF spectrum are investigated, and its mathematical and empirical relationship to a speech signal is determined. Particular focus is given to the effect of the analysis window on the calculation of the IF spectrum. To investigate the IF spectrum behaviour further, we then shift attention to the related quantity of IF deviation – a measure of how far the IF 'deviates' from a centre frequency. We show that this deviation quantity contains many useful traits applicable to speech processing. Using this knowledge, a new spectral representation based on the short-time IF deviation spectrum is proposed.

- **Chapter 6** describes research into the short-time group delay (GD) spectrum. In a similar approach to Chapter 5, the properties of the GD spectrum are analysed and a spectral representation based on the group delay deviation is proposed. The differences between the IF and GD spectra and their associated spectral representations are investigated.

- **Chapter 7** details a novel speech enhancement algorithm that achieves noise suppression via manipulation of the short-time phase spectrum. Previous research has shown that noise suppression can be achieved by exploiting the phase relationships between Fourier analysis conjugate pairs. In the current approach, this noise suppression mechanism is extended for the purposes of online speech enhancement. To accomplish this, we derive a noise-driven

heuristic to control the phase-based noise suppression. The proposed algorithm is tested across a number of different noise types and its subjective quality is examined.

Part II: Investigation into stochastic estimation of the Mel-frequency cepstral coefficient feature set

- **Chapter 8** investigates modifications to the short-time spectral energy estimator for use in robust ASR. Traditionally, this estimator has been shunned because of its tendency to under-suppress additive noise. However, we show the spectral energy estimator is closely related to the optimal MFCC estimator. With some simple heuristic modifications, we show that the spectral energy estimator can exhibit greatly improved performance for robust ASR. Two specific modifications are examined: 1) a heuristic to modify speech presence uncertainty and 2) a direct heuristic modification to the noise suppression profile. Recognition results for both modifications are given for the OLLO, RM and Aurora2 speech recognition tasks.

- **Chapter 9** investigates the minimum mean square estimation of the MFCC vector for use in robust automatic speech recognition. Since a stochastic estimator is only as good as the statistical assumptions that underpin it, we start with a well understood spectral-domain noise distortion framework. In order to use this framework for MFCC estimation, the spectral-domain noise distortion models must be modified into cepstral-domain models. The mathematical transformations required for converting the spectral-domain models into spectral energy, filterbank energy, log-filterbank energy and cepstral models are derived and discussed. Results for the proposed estimator are presented for the Aurora2 and RM speech recognition tasks.

1.4 Contributions

- An efficient algorithm for the regeneration of spectral phase from spectral magnitude (Chapter 4).
- Properties of the short-time instantaneous frequency spectrum are mathematically and empirically quantified. A novel spectral representation is proposed based on the findings (Chapter 5). Properties of the short-time group delay are similarly studied (Chapter 6).
- A novel, noise-driven heuristic is developed for a phase spectrum based speech enhancement algorithm (Chapter 7).
- An investigation into the properties and application of short-time spectral energy estimation for robust speech recognition is presented. A heuristic (Chapter 8) and a stochastic (Chapter 9) approach is taken to modifying this estimator for use in robust automatic speech recognition.

1.5 Publications

1.5.1 Publications resulting from dissertation research

1. A. Stark and K. Paliwal, "MMSE estimation of log-filterbank energies for robust speech recognition", *Speech Communication, accepted.*

2. A. Stark and K. Paliwal, "Minimum mean square error estimation of the Mel-frequency cepstral coefficient vector for robust speech recognition", *IEEE Trans. on Audio Speech and Language Processing, under review.*

3. A. Stark and K. Paliwal, "Instantaneous-frequency deviation based spectral representation of speech", *Speech Communication, under review.*

4. A. Stark and K. Paliwal, "Use of speech presence uncertainty with MMSE spectral energy estimation for robust automatic speech recognition", *Speech Communication, accepted.*

5. A. Stark and K. Paliwal, "Group-delay-deviation based spectral analysis of speech", *International Conf. on Spoken Language Processing*, 2009, pp 1083-1086.

6. A. Stark, K. Wójcicki, J. Lyons and K. Paliwal, "Noise driven short-time phase spectrum compensation procedure for speech enhancement", *International Conf. on Spoken Language Processing*, 2008, pp 549-552.

7. A. Stark and K. Paliwal, "Speech analysis using instantaneous frequency deviation", *Proc. International Conf. on Spoken Language Processing*, 2008, pp 2602-2605.

1.5.2 Other publications

1. S. So, K. Wojcicki, J. Lyons, A. Stark and K. Paliwal, "Kalman filter with phase spectrum compensation algorithm for speech enhancement", *Proc. IEEE*

Intern. Conf. on Acoustics, Speech and Signal Processing, Taipei, Apr. 2009, pp. 4405-4408.

2. K. Wójcicki, M. Milacic, A. Stark, J. Lyons and K. Paliwal, "Exploiting conjugate symmetry of the short-time Fourier spectrum for speech enhancement", *IEEE Signal Processing Letters*, 2008, Vol.15, pp 461-464.

3. J. Lu, A. Stark and D. Thiel, "Switched Parasitic Patch Antenna Array Using Thirteen Hexagonal Shaped Elements", *ISAPE2008, the 8th International Symposium on Antennas, Propagation, and EM Theory*, Kunming 2008.

4. A. Stark and K. Paliwal, "An empirical comparison of three initialization techniques for categorical K-means algorithms", P*roc. Griffith School of Engineering Research Conference, Brisbane*, Australia, Oct. 2007

5. A. Stark and J. Lu, "An electronically steerable patch antenna for indoor wireless communications", *Tenth Australian Symposium on Antennas*, Sydney 2007.

Chapter 2

Speech processing fundamentals

2.1 Speech production

The production of speech is a complicated process involving both physical and psychological characteristics. Physically, the vocal chords, vocal tract, nasal cavity, tongue and teeth all play a role in the production of speech. Given the large number of physical variables, it is useful to develop a simpler mathematical analogue. In this work, we consider the source-filter model of speech production [69]. The source aspect of the model describes two types of speech, *voiced* speech and *unvoiced* speech. Voicing arises from the state of the vocal chords. In the case of voiced speech, the vocal chords are stretched taut. Air delivered from the lungs causes pressure behind the vocal chords to rise. This continues until the building air pressure reaches a threshold, escaping through the vocal tract and producing a *glottal pulse*. With the pressure released, the vocal chords close and the cycle begins anew. The frequency of this cycle leads to the perceptual quantity of *pitch*. An example of voiced speech includes the 'a' in car and the 'e' in bee. In the case of unvoiced speech, the vocal tract is left relaxed. This allows turbulent air to pass through the vocal tract unaltered. Unvoiced sounds lack the time-frequency regularity present in voiced

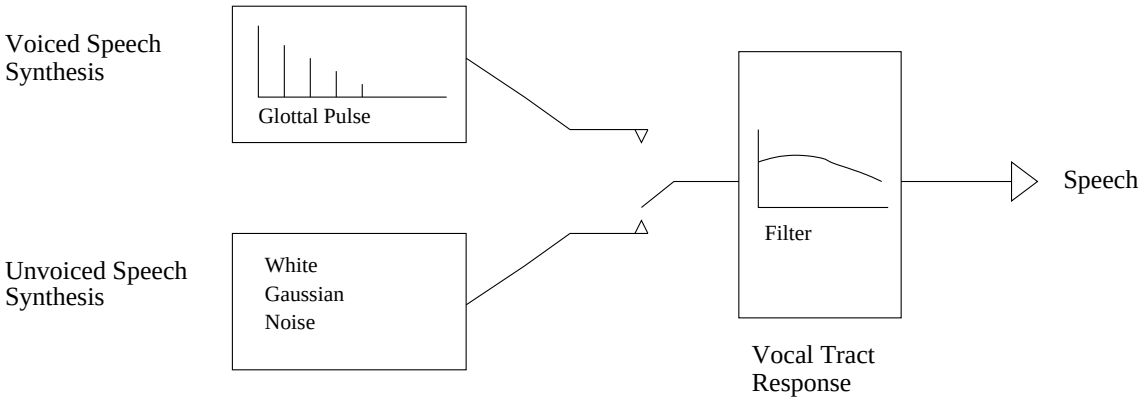

Figure 2.1: Illustration of the source-filter voice synthesis model.

sounds, and typically have less acoustic energy [69]. Unvoiced speech includes the 's' in stop and 't' in hat. Mathematically, voiced speech excitation is described as a periodic glottal pulse train while unvoiced speech excitation is modelled with white Gaussian noise.

The filter aspect of the source-filter model describes acoustic properties of the mouth, nasal cavity, tongue, lips and velum. Collectively they act as a time varying filter; enhancing certain acoustic frequencies, while suppressing others. The resonant frequencies of this filter system are termed formants, and are responsible for conveying the majority of speech intelligibility. Mathematically, the complete source-filter model of speech production can be described with the following convolutional relationship

$$x(t) = e(t) * h(t), \tag{2.1}$$

where $x(t)$, $e(t)$ and $h(t)$ are the time-domain speech signal, excitation signal and filter response respectively. The convolution operator is given as $*$. Fig. 2.1 shows a block diagram of the source-filter speech production model.

2.2 Speech perception

Like speech production, the hearing process is a product of many processes. Physically, the ear plays a dominant role – converting acoustic energy into an electrical signal. These signals are then relayed to the brain, where they are interpreted into a meaningful message. The ear itself is a complex organ, and can be classified into three main regions: the outer, middle and inner ear [47, 69].

1. The outer ear, consisting of auricle and external auditory canal, capture sound waves, amplifying them and channelling them inwards toward the eardrum.

2. The middle ear, consists of the ear drum (tympanic membrane) and ossicles – small bone structures. When sound energy strikes the eardrum, it vibrates, and these vibrations are passed through to the ossicles where they are amplified before being passed into the cochlea.

3. The inner ear cochlea is a fluid-filled, spiral shaped organ. Within the cochlea, tiny hairs (cilia) pick up vibrations, converting them to electrical energy. These electrical signals are finally relayed to the brain via the auditory nerve.

While there has been considerable research into the physical hearing system, our knowledge of the brain and its higher level functionality is much less complete. The remainder of this section describes some empirically determined properties of the physical hearing system.

2.2.1 Loudness perception

Perceived loudness is related to the logarithm of acoustic sound pressure [47, 69]. In more precise terms, loudness perception is a non-linear function of frequency as well as acoustic energy. Unsurprisingly the most sensitive frequency region – 500 Hz to 5 kHz corresponds to the typical domain of human speech. Fig. 2.2 shows the empirically determined relationship between perceived loudness (phons), frequency and sound pressure level (SPL). In terms of physical quantities, human

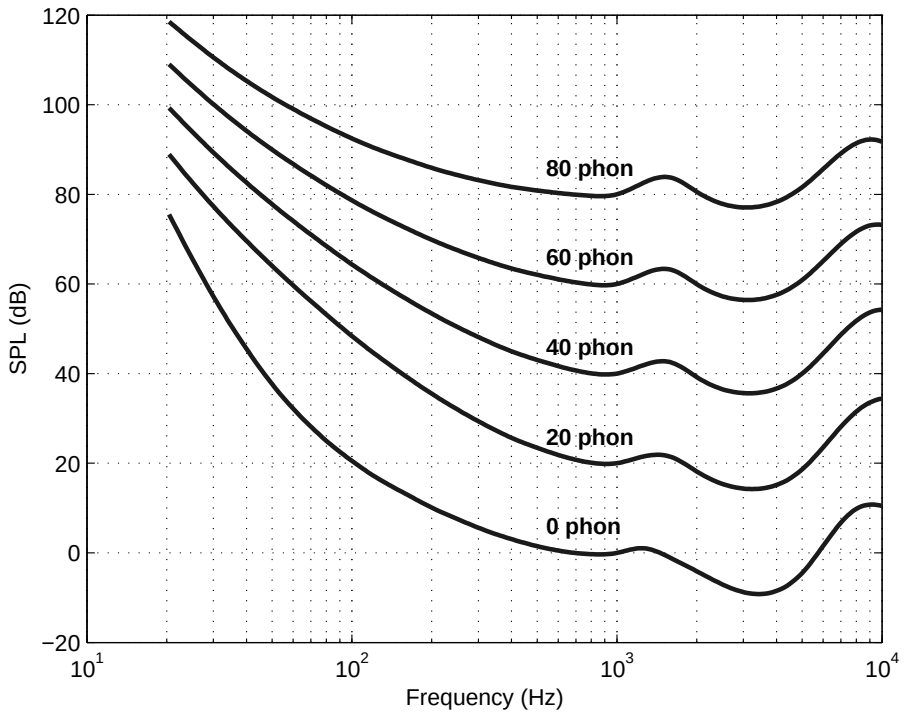

Figure 2.2: Perceptual loudness (phons) as a function of frequency and sound pressure level (SPL). Greatest sensitivity is between 500 Hz and 5000 kHz. Curves are based on the ISO226 standard [1].

hearing ranges from 0 dB SPL ($10^{-12}W.m^{-2}$, threshold of hearing), to 120 dB SPL ($1W.m^{-2}$, painfully loud).

2.2.2 Frequency perception

The healthy ear can detect frequencies between 20 Hz and 20 kHz. Like loudness perception, frequency (pitch) perception is roughly logarithmic in nature. The cochlea itself can be thought of as a biological spectrum analyser. This allows it to be modelled as a series of overlapped, logarithmically spaced filterbanks. The properties of these filterbanks are derived from *critical bands* – a set of experimentally determined psychoacoustic quantities [47]. Two frequency scales, the Bark scale and Mel scale are based on these critical bands, and offer a nonlinear frequency scale that is motivated by the perceptual properties of the human ear.

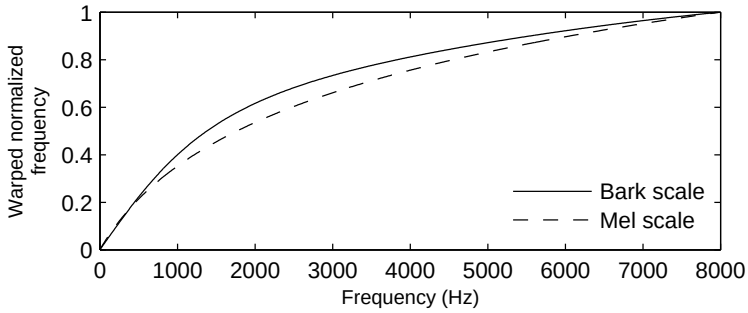

Figure 2.3: The normalized Bark and Mel frequency warping functions. Both warping functions are roughly logarithmic in nature.

The two scales, given as a function of frequency f (in Hertz) are [69]:

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

$$M(f) = 1125 \log(1 + f/700). \tag{2.3}$$

Fig. 2.3 illustrates the frequency warping achieved by the Bark and Mel scales. Both scales have similar properties.

2.3 Digital processing of speech signals

The majority of speech information is concentrated within the frequency regions 100 Hz to 4 kHz [69]. Thus, a sampling rate of 8 kHz (sampling time of 125 ms) is generally sufficient to provide telephone quality speech digitization. However, quality and naturalness are improved by sampling at 16 kHz and higher.

While the time sampled representation of the speech is adequate for recording and playback, it is more typical to analyse speech in the *spectral*-domain. In order to satisfy the stationarity requirements of spectral transforms, speech is often assumed to be quasi-stationary over short-time (up to 32 ms) intervals. This facilitates the need to decompose the entire speech signal into separate and often overlapping

short-time frames [115]. Using the discrete short-time Fourier transform (DSTFT), each of these frames are transformed to the spectral-domain. For a discrete-time sampled speech signal $x(n)$, the DSTFT signal $X(m,k)$ is given by

$$X(m,k) = \sum_{n=-\infty}^{\infty} x(n)w(mS-n)\exp(-j2\pi kn/K), \tag{2.4}$$

where k denotes the k'th discrete-frequency of K uniformly spaced frequencies, $w(n)$ is an analysis window function, m is the DSTFT frame index and S is the frame-shift in samples. For speech analysis, $w(n)$ often takes the form of a Hamming window

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) & 0 \leq n < N \\ 0, & \text{otherwise,} \end{cases} \tag{2.5}$$

where N is the length of the window in samples. The tapered edges of the Hamming window help reduce the phenomenon of spectral leakage, albeit at the loss of some spectral resolution [64, 115]. Since fine spectral resolution is generally not considered important for speech processing, windowing is a very common procedure.

The short-time spectral signal $X(m,k)$ contains information from both the short-time amplitude/magnitude[1] and short-time phase spectra. The complex short-time spectral signal $X(m,k)$ may be decomposed into short-time spectral magnitude and short-time spectral phase as follows

$$\begin{aligned} X(m,k) &= |X(m,k)|\exp\left(j\angle\left[X(m,k)\right]\right) \\ &= A(m,k).\exp\left(j\theta(m,k)\right), \end{aligned} \tag{2.6}$$

where $A(m,k)$ and $\theta(m,k)$ are the DSTFT magnitude and phase spectra respectively. In some applications, it is common to use the DSTFT power spectrum. This is simply an alternate representation of the same spectral magnitude information.

[1] In this thesis, the terms magnitude and amplitude are used interchangeably when used to describe the absolute value of a spectral signal. Likewise, the short-time magnitude-squared, power and energy spectra are used to refer to the same information.

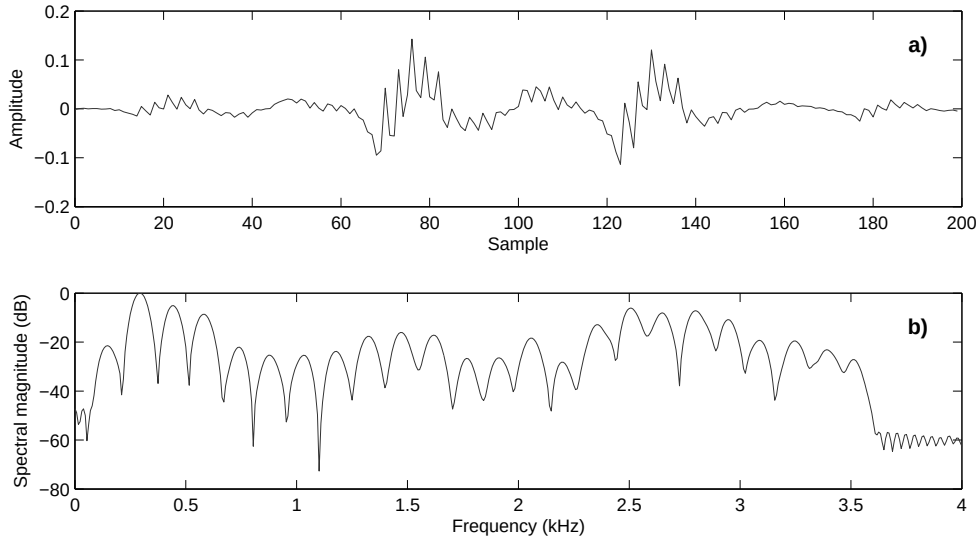

Figure 2.4: Spectral analysis for a 25 ms segment of voiced speech. Subplots: a) time domain speech with (Hamming) windowing, b) normalized magnitude spectrum (in log domain).

Whatever the particular format, spectral magnitude/power information forms the basis of most spectral-domain speech algorithms. This is because several key speech features directly manifest within the DSTFT magnitude spectrum. Fig. 2.4 shows the short-time magnitude spectrum for a single frame of speech. The evenly spaced peaks in the magnitude plot are the result of voiced speech pitch harmonics. Formants are identified as the wider frequency regions of high acoustic power.

Multiple spectral frames may be used to produce a time-frequency spectrogram representation. Figs. 2.5 and 2.6 show narrow-band spectrograms and Figs. 2.7 and 2.8 show corresponding wide-band spectrograms. Darker areas on the spectrograms show time-frequency locations of high acoustic energy. For the narrow-band spectrograms (25 ms window) we have enough frequency resolution to resolve individual pitch harmonics. Here we can see a clear difference between the male speaker (low fundamental frequency, closely spaced harmonics) in Fig. 2.5 and female speaker (high fundamental frequency, largely spaced harmonics) in Fig. 2.6. When using wide-band analysis (5 ms window), fine pitch structure is lost. However, formant regions (broader frequency regions of high acoustic energy) are retained.

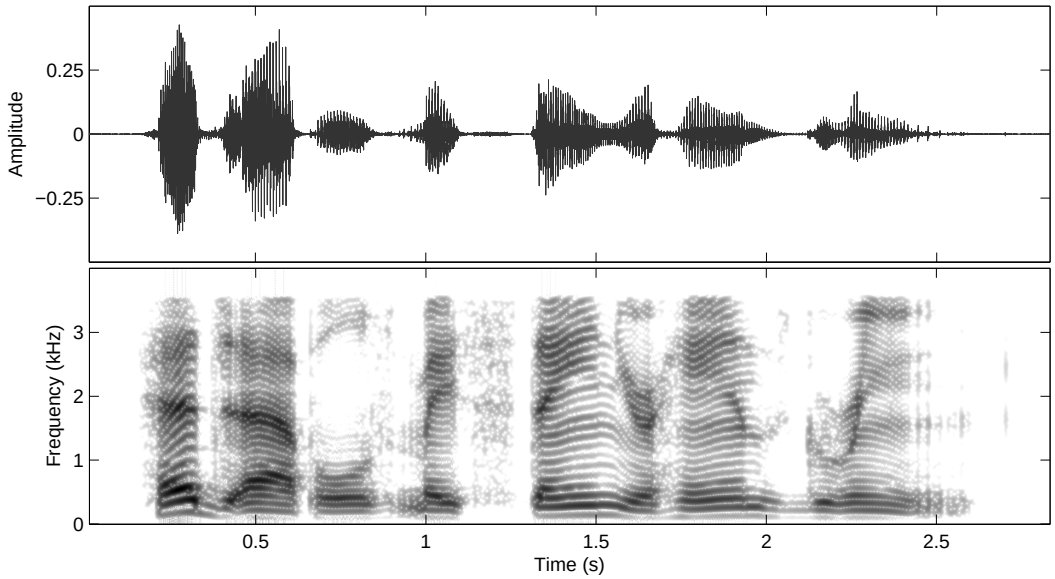

Figure 2.5: Subplots: a) time-domain speech and b), corresponding synchronous narrow-band spectral analysis. Speech is from a male speaker for the sentence: 'Hedge apples may stain your hands green'. Spectral analysis uses 25 ms segments and 50dB dynamic range.

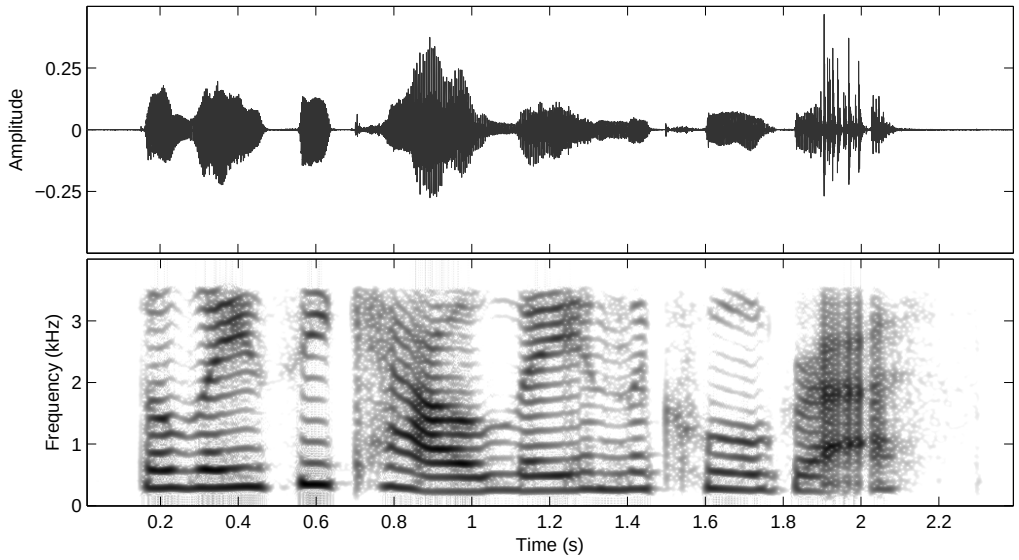

Figure 2.6: Subplots: a) time-domain speech and b), corresponding synchronous narrow-band spectral analysis. Speech is from a female speaker for the sentence: 'The lazy cow lay in the cool grass'. Spectral analysis uses 25 ms segments and 50dB dynamic range.

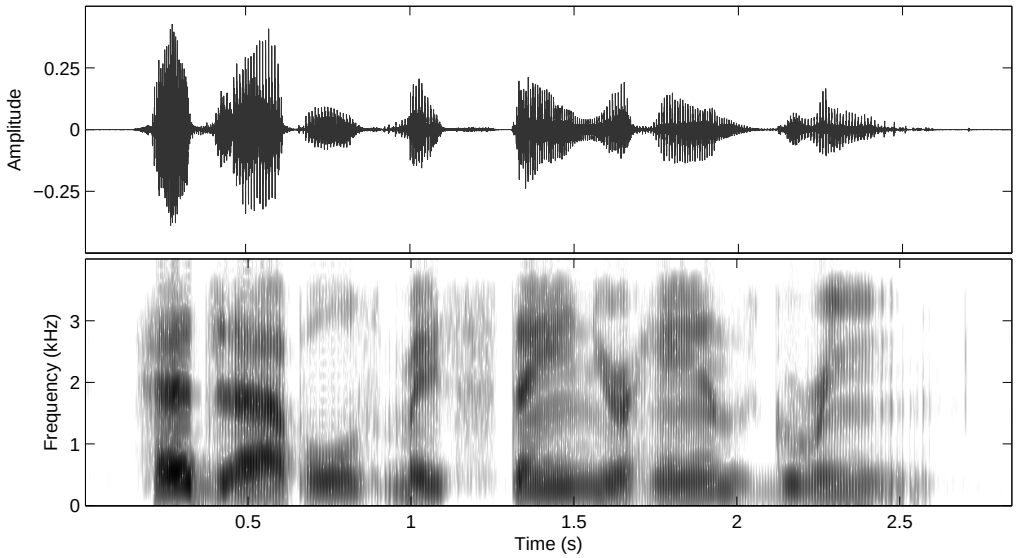

Figure 2.7: Subplots: a) time-domain speech and b), corresponding synchronous wide-band spectral analysis. Speech is from a male speaker for the sentence: 'Hedge apples may stain your hands green'. Spectral analysis uses 5 ms segments and 50dB dynamic range.

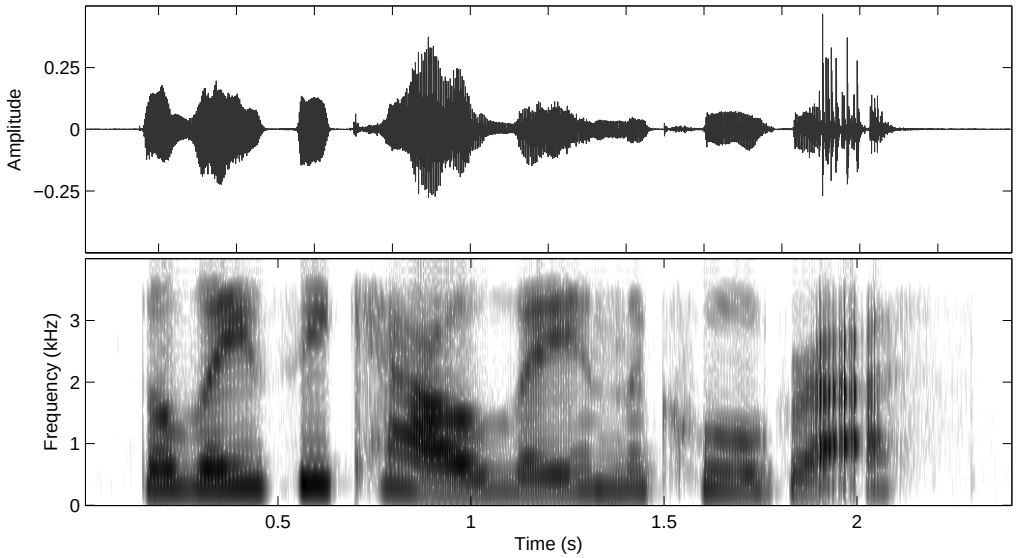

Figure 2.8: Subplots: a) time-domain speech and b), corresponding synchronous wide-band spectral analysis. Speech is from a female speaker for the sentence: 'The lazy cow lay in the cool grass'. Spectral analysis uses 5 ms segments and 50dB dynamic range.

2.4 Overview of the short-time Fourier phase spectrum

In the previous sub-sections, the short-time Fourier phase spectrum was rarely mentioned when discussing speech characteristics. Such a practice is common in the majority of speech processing and speech recognition literature. This is because most spectral speech processing methods are based on the short-time Fourier magnitude spectrum. This is true for both automatic speech/speaker recognition, as well as speech enhancement. Two primary reasons are responsible for this. Firstly, the phase spectrum is difficult to interpret and process. The phase spectrum itself requires significant processing in order to extract useful information. Furthermore it suffers from many tractability issues, including volatility and the phase unwrapping problem [6]. In contrast, the magnitude spectrum requires no such post-processing. The visual cues that manifest within the magnitude spectrum directly correlate with major speech structures. Formant and pitch information for example, are both readily seen in the magnitude spectrum. The second reason for avoiding spectral phase can be attributed to several well known perceptual experiments [79, 96, 118], which show marginal phase spectrum intelligibility over short (20-40 ms) window durations.

Contrary to these findings, it has recently been shown that stimuli constructed from the short-time phase spectrum can convey intelligibility comparable to its magnitude-only counterpart [10]. This result is supported by many studies which highlight a strong relationship between the two spectra – notably the recovery of magnitude spectra from phase [65] and vice versa [62]. The first part of this dissertation is motivated by the desire to further pursue this link.

In the remainder of this section, the general issues regarding use of the phase spectrum are discussed. In particular, the phase unwrapping and the volatility problems are examined. Later, in Chapters 4–7, specific avenues of DSTFT phase spectrum research are presented.

2.4.1 Signal processing requirements for the phase spectrum

A number of considerations must be taken into account when processing the short-time phase spectrum. On its own, the phase spectrum isn't particularly intuitive or useful. In fact, the principal phase spectrum appears to have little physical meaning. However, a more practical problem concerns its tractability issues; notably the phase unwrapping and the volatility problems. Both problems are discussed in further detail below.

The phase unwrapping problem

The true phase spectrum of short-time spectral signal $X(m,k)$ is defined on an unbounded interval; i.e., $-\infty < \angle X(m,k) < \infty$. However, calculating the phase spectrum in such a form is quite difficult. In most cases, the principal phase spectrum is calculated instead. The principle phase spectrum $ARG[X(m,k)]$, can be calculated via the four quadrant arctan function

$$ARG[X(m,k)] = \arctan\left(\frac{X_I(m,k)}{X_R(m,k)}\right), \tag{2.7}$$

where $X_R(m,k)$ and $X_I(m,k)$ are the real and imaginary components of short-time spectral signal $X(m,k)$ respectively. The principle phase spectrum is defined on a bounded interval, typically between the limits $[-\pi,\pi]$. This *wrapping* often makes principal phase values appear chaotic and lacking in structure. To address this issue, phase unwrapping is often employed – heuristic algorithms used to map the principal phase values back to an unbounded domain. The goal of these algorithms is to produce a continuous phase function that is visually more meaningful, and hopefully close to the true underlying phase spectrum. To do this a multiple of 2π can be added to any principal phase value. Of course, this leads to an infinite number of ways to actually unwrap the phase. Two simple heuristics are used most often to unwrap the phase: 1) numerical integration of the phase spectrum frequency derivative (group delay spectrum) and 2) constraining the absolute difference between adjacent phase

values to be less than some threshold (usually π). Being heuristics, these methods are not guaranteed to give a single, true unwrapped phase spectrum. However, their accuracy can be improved by using over-sampled Fourier transforms; i.e., by increasing the total number of spectral bins K in (2.4).

The phase volatility problem

Another issue regarding use of the phase spectrum is volatility. This is an undesirable trait that can severely reduce the robustness of phase-based algorithms. Two main sources of volatility exist in phase processing: temporal volatility and spectral volatility. Temporal volatility relates to the time placement of the DSTFT analysis frame. The principal phase spectrum is heavily dependent on the time location of its analysis frame, and even small perturbations are able to radically alter the phase spectrum. The magnitude spectrum on the other hand exhibits much less temporal volatility, especially when used in its narrow-band format. In order to reduce temporal volatility, it is common to use the phase spectrum derivative. Differentiation of the DSTFT phase spectrum may be performed with respect to either time or frequency, leading to the short-time instantaneous frequency and group delay spectra respectively. Both spectra exhibit much less temporal volatility than the original phase spectrum.

Spectral volatility arises from the phase estimation of low energy spectral components. As $X(m,k)$ approaches zero, (2.7) becomes numerically unstable. In fact, when $X(m,k) = 0$, the phase spectrum is mathematically undefined. Again, the magnitude spectrum does not experience this form of instability. Unfortunately, both the instantaneous frequency and group delay spectra manifest this form of instability. However, spectral phase volatility may be reduced with either smoothing/averaging or combination with spectral magnitude.

Chapter 3

Automatic speech recognition

Automatic speech recognition (ASR) is the process of enabling a machine to transcribe spoken language. The recognition problem consists of two fundamental modules: a front-end speech parameterization stage and a back-end recognition stage. Fig. 3.1 shows an overview of a typical ASR structure. For the front-end stage, we are concerned with parameterizing the speech – converting it into a format machines can both use and understand. The back-end stage consists of training and deployment. Training is undertaken so the system will learn to associate acoustic models with known transcriptions. Once trained, the back-end recognizer can be deployed to transcribe unknown speech samples. Here, the system chooses a transcription that best matches the incoming acoustic data. While there is some variety in the literature concerning front-end speech processing, stochastic algorithms form the vast bulk of back-end recognizers. Of these, the hidden Markov model (HMM) is ubiquitous in automated speech recognition.

In the stochastic approach, the hypothesized word string $\hat{\boldsymbol{W}}$ is chosen as most likely word sequence, given observational (speech data) evidence. This can be given

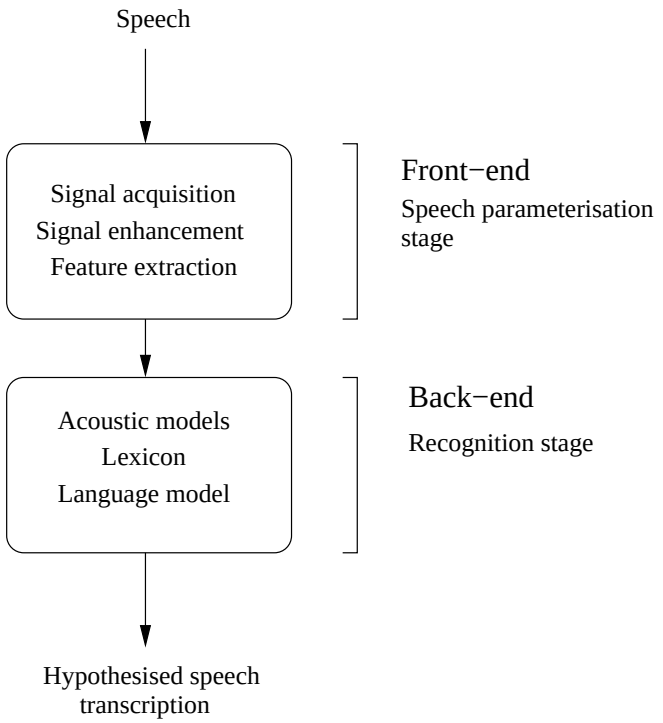

Figure 3.1: Generic ASR framework with front-end feature extraction stage and back-end recognition stage.

by the following Bayesian decision rule

$$\begin{aligned} \hat{\boldsymbol{W}} &= \underset{\boldsymbol{W}}{\operatorname{argmax}} \left[P(\boldsymbol{W}|\boldsymbol{O}) \right] \\ &= \underset{\boldsymbol{W}}{\operatorname{argmax}} \left[\frac{P(\boldsymbol{W})P(\boldsymbol{O}|\boldsymbol{W})}{P(\boldsymbol{O})} \right], \end{aligned} \tag{3.1}$$

where $\hat{\boldsymbol{W}}$ is the estimated (most likely) word sequence, $\boldsymbol{O}$ is the set of observed speech data, $P(\boldsymbol{W})$ is the *a priori* probability of a particular word sequence $\boldsymbol{W}$ and $P(\boldsymbol{O}|\boldsymbol{W})$ is the probability of observing the acoustic data $\boldsymbol{O}$ conditioned on the word sequence $\boldsymbol{W}$. In practice, we are more interested in discriminating between word sequences (to select the most appropriate one) rather than determining the exact value of likelihoods. Using the fact that $P(\boldsymbol{O})$ is invariant to $\boldsymbol{W}$, the hypothesized

word sequence can be given by the following

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} \left[\log P(\boldsymbol{W}) + \log P(\boldsymbol{O}|\boldsymbol{W}) \right]. \tag{3.2}$$

Mathematically, the use of log-likelihoods does not alter our decision rule as the logarithm is a monotonically increasing function. However, log-likelihoods are more computationally tractable than standard likelihoods, making them useful for computerized implementation.

3.1 Speech recognition with hidden Markov models

While there have been many methods proposed for recognizing speech, stochastic methods based on hidden Markov models (HMMs) are the most successful and widespread [33, 53, 69, 112, 114, 130, 142]. HMMs form the acoustic models of speech, providing a mechanism for determining the conditional probability $P(\boldsymbol{O}|\boldsymbol{W})$. When combined with a language model – which provides *a priori* knowledge of word sequences $P(\boldsymbol{W})$, a final hypothesized speech transcription may be obtained. Much of the success of HMMs comes from their ability to provide temporal modelling within an efficient and elegant framework. To do this, HMMs assume the speech process to exist in one of N finite, but hidden states. Each state is used to model a particular set of acoustic observations (a particular vowel for example), and the overall system is said to evolve between states.

While there are many types of HMM, in this dissertation we will consider the most wide spread HMM architecture in ASR – continuous density, first order, left-to-right HMMs. *Continuous density* refers to the state-conditioned observation probabilities – a continuous function, which usually takes the form of a multivariate Gaussian mixture model (GMM). *Left-to-right* refers to a linearly progressive state transition (see Fig. 3.2), and *first order* reflects a state transition that is dependent only on the immediately previous state.

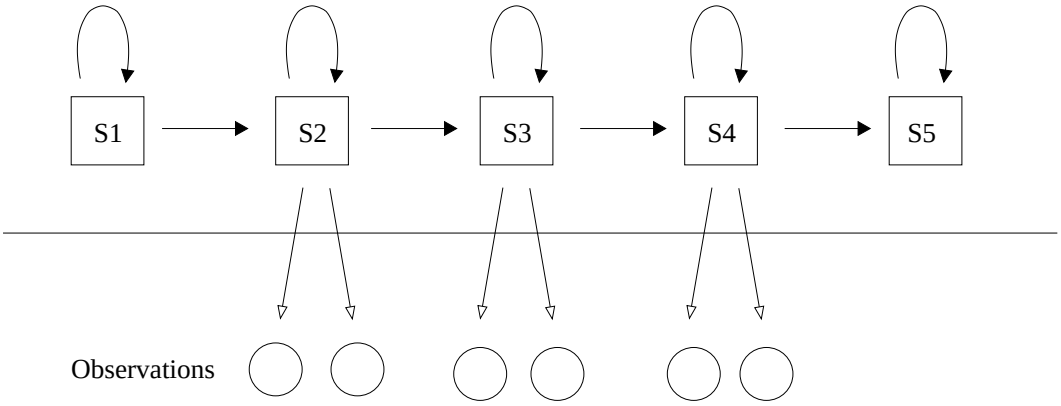

Figure 3.2: Illustration of a 5-state, left-to-right HMM typical for phoneme modelling. A series of hidden states generate the observations seen. For each observation, the state may remain the same, or transition to next state (hence left-to-right). States 2-4 are considered emitting states, while states 1 and 5 are used to tie models together into larger HMMs.

In order to provide a tractable mathematical framework, first order HMMs operate under the following statistical assumptions:

- The probability of a given observation is dependent only on the current state.
- The probability of changing states is dependent only on the immediately previous state. Transitions are considered instantaneous.

Strictly speaking, neither assumption is true for speech. In particular, the first order constraint fails to capture the highly contextual nature of speech. Nonetheless, HMMs remain popular due to their efficiency and good performance.

3.1.1 Overview of the hidden Markov model structure

Mathematically, a first-order HMM consists of the following constructs[1]:

- N states, given by $S_1, S_2....S_N$.
- A sequence of observed feature vectors $\mathbf{O}$.
- A sequence of underlying states $\mathbf{Q}$.
- HMM model $\mathbf{\Lambda}$ consisting of:
  - A state transition matrix $\mathbf{A}$, where A_{ij} is the probability of a state moving from state i to state j.
  - Observation emission probabilities $\mathbf{b}_j(O)$ - the probability that observation O was generated by state j. For speech, these are typically modelled with a Gaussian mixture model, each of which has a set of M mixture weights, mean vectors and covariance matrices.
  - π_j - the probability of being in state j at time $t = 0$.

Given these parameters, there are three fundamental goals concerning HMMs:

1. Given a model $\mathbf{\Lambda}$, determine the probability of the observation sequence $\mathbf{O}$; $P(\mathbf{O}|\mathbf{\Lambda})$, where $\mathbf{O} = [O_1, O_2, ...O_T]$.
2. Given an observation sequence $\mathbf{O}$ and a model, determine the most likely set of states $\mathbf{Q}$.
3. Given an observation sequence $\mathbf{O}$, determine the model $\mathbf{\Lambda}$ such that $P(\mathbf{O}|\mathbf{\Lambda})$ is maximized.

[1] Mathematical symbols defined for the HMM parameters are applicable to this chapter only.

3.1.2 Determining probability of an observation sequence

To address the first HMM problem, it becomes useful to define a forward variable α and backward variable β [112]. $\alpha_t(i)$ is given as the probability of a partial observation up till time t, with current state equal to state S_i, given the model $\mathbf{\Lambda}$.

$$\alpha_t(i) = P(O_1, O_2, ...O_t, Q_t = S_i \mid \mathbf{\Lambda}). \tag{3.3}$$

$\alpha_t(i)$ can then be solved iteratively as follows:

1. Initialization ($t = 1$):

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N. \tag{3.4}$$

2. Induction ($1 < t < T$):

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) A_{ij}\right] b_j(O_{t+1}), \quad 1 \leq i \leq N. \tag{3.5}$$

3. Termination ($t = T$):

$$P(\mathbf{O} \mid \mathbf{\Lambda}) = \sum_{i=1}^{N} \alpha_T(i), \quad 1 \leq i \leq N. \tag{3.6}$$

The iterative formulation of $\alpha_t(i)$ utilizes an efficient trellis structure that vastly improves performance over a brute force calculation of $P(\mathbf{O} \mid \mathbf{\Lambda})$. Although the first problem has now been solved, it becomes useful to describe a similar backward variable β to help solve the third HMM problem. Here, $\beta_t(i)$ is the probability from time $t + 1$ to the end, given state S_i and model $\mathbf{\Lambda}$.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}...O_T | Q_t = S_i, \mathbf{\Lambda}). \tag{3.7}$$

As with $\alpha_t(i)$, $\beta_t(i)$ may be solved iteratively:

1. Initialization ($t = T$):

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \tag{3.8}$$

2. Induction ($t = T-1, T-2...1$):

$$\beta_t(i) = \sum_{j=1}^{N} A_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N. \tag{3.9}$$

3.1.3 Determination of the most likely state sequence

The second HMM problem can be solved via the Viterbi algorithm [112, 142]. We can define another variable $\delta_t(i)$ that describes the probability of the best path (highest likelihood), for the first t observations, where the current state is given by S_i.

$$\delta_t(i) = \max_{Q_1, Q_2..Q_{t-1}} P\left[Q_1, Q_2...Q_t = i, O_1, O_2, ...O_t \mid \mathbf{\Lambda}\right]. \tag{3.10}$$

A further variable, $\psi_t(i)$ keeps track of the state index that maximizes (3.10). The two variables can then be solved inductively as follows:

1. Initialization ($t = 1$):

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N, \tag{3.11}$$

$$\psi_1(i) = 0, \quad 1 \leq i \leq N. \tag{3.12}$$

2. Induction ($1 < t \leq T$):

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) A_{ij}\right] b_j(O_t), \quad 1 \leq j \leq N. \tag{3.13}$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \leq i \leq N} \left[\delta_{t-1}(i) A_{ij}\right], \quad 1 \leq j \leq N. \tag{3.14}$$

3. Termination ($t = T$):

$$q_T^* = \underset{1 \leq i \leq N}{\operatorname{argmax}}[\delta_T(i)]. \tag{3.15}$$

4. Determination of optimal sequence via backtrack:

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{3.16}$$

3.1.4 Optimization of the Markov model

The third and most difficult problem can be solved by the Baum-Welch method, an iterative algorithm that utilizes the classic expectation-maximization framework [112, 142]. The variable $\gamma_t(i)$ can be defined, that describes the probability of being in state S_i at time t. Furthermore, it can be expressed in terms of the forward and backward variables defined earlier.

$$\gamma_t(i) = P(Q_t = S_i \mid \mathbf{O}, \mathbf{\Lambda}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}. \tag{3.17}$$

Likewise, the variable $\xi_t(i, j)$ describes the probability of being in state i at time t, and state j at time $t + 1$.

$$\begin{aligned} \xi_t(i,j) &= P(Q_t = S_i, Q_{t+1} = S_j \mid \mathbf{O}, \mathbf{\Lambda}) \\ &= \frac{\alpha_t(i) A_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{x=1}^{N} \sum_{y=1}^{N} \alpha_t(x) A_{xy} b_j(O_{t+1}) \beta_{t+1}(y)}. \end{aligned} \tag{3.18}$$

Model parameters can then be re-evaluated as follows. For the initial state probabilities:

$$\bar{\pi}_i = \gamma_1(i). \tag{3.19}$$

For the transition probabilities:

$$\bar{A}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{i=1}^{T-1} \gamma_t(i)}. \tag{3.20}$$

For the state conditioned observation probabilities:

$$\bar{b}_j(o) = \frac{\sum_{t=1}^{T-1}{}^{s.t.O_t=o} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)}. \tag{3.21}$$

In the case of GMM observation probabilities, the following extensions can be made.

$$\begin{aligned} b_j(o) &= \sum_{m=1}^{M} \eta_{jm} b_{jm}(o) \\ &= \sum_{m=1}^{M} \eta_{jm} \mathcal{N}(o; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}), \end{aligned} \tag{3.22}$$

where η_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ are the mixture weight, mixture mean vector and mixture covariance matrix for the j'th state's m'th mixture. Updates for the GMM parameters can be given as follows:

$$\hat{\eta}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m)}{\sum_{t=1}^{T} \sum_{n=1}^{N} \gamma_t(j,n)}, \tag{3.23}$$

$$\bar{\boldsymbol{\mu}}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) \boldsymbol{O}_t}{\sum_{t=1}^{T} \gamma_t(j,m)}, \tag{3.24}$$

$$\bar{\boldsymbol{\Sigma}}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,m) (\boldsymbol{O}_t - \bar{\boldsymbol{\mu}}_{jm})^T (\boldsymbol{O}_t - \bar{\boldsymbol{\mu}}_{jm})}{\sum_{t=1}^{T} \gamma_t(j,m)}, \tag{3.25}$$

where $\gamma_t(j,m)$ is the probability of being in mixture m, state j:

$$\gamma_t(j,m) = \gamma_t(j). \left[\frac{\eta_{jm} \mathcal{N}(o; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})}{\sum_{n=1}^{M} \eta_{jn} \mathcal{N}(o; \boldsymbol{\mu}_{jn}, \boldsymbol{\Sigma}_{jn})} \right]. \tag{3.26}$$

Given the above equations, the model parameters and state probability variables can be iteratively optimized to reach some local convergence. While global optimality cannot be guaranteed with this method, a number of practical heuristics exist to give a good parameter fit while avoiding the problem of over-training. The reader

is referred to the HTK book [142] for further ASR specific implementation detail.

3.1.5 HMM unit modelling and lexicon

For small vocabulary recognition, individual HMMs may be trained for every possible word. Word HMMs typically have a large number of emitting states (15-20), to account for a large degree of acoustic variability in a given word [142]. Two anchor states are appended to the beginning and end of the word-HMMs. This allows individual HMMs to be tied together into larger models, allowing evaluation over word sequences.

For large vocabulary systems, word-level modelling becomes infeasible. Firstly, the number of models would be prohibitively large and secondly, there may not be sufficient data to adequately train a model for each word. In this case, it is standard practice to use sub-word *phonemes* as the modelling unit. Phonemes can be considered an atomic unit of speech – that is, the smallest unit of speech to convey some meaning. As such, they are usually modelled with a small number of states – typically 3 emitting states (with 2 non-emitting anchors). The word-to-phoneme mapping is provided by a dictionary commonly referred to as the lexicon. For example, the word *next* can be decomposed into the following phoneme sequence:

$$\text{next} \longrightarrow \mathbf{n} \;+\; \mathbf{eh} \;+\; \mathbf{kcl} \;+\; \mathbf{s} \;+\; \mathbf{tcl} \;+\; \mathbf{t}$$

This decomposition cuts down on the total number of HMMs required, and also provides a larger pool of training data for each phoneme. The lexicon allows conversion to phoneme units for training, and conversion to words after recognition. Recognition of speech in this manner is referred to as monophone recognition. It should be pointed out that the phoneme mapping for a particular word is not static and can vary between different accents.

In many cases, performance can be increased by extending the monophone models into *triphones*. Triphones are models of the preceding, current and following phoneme. For example:

next $\longrightarrow$ **sil-n-eh + n-eh-kcl + eh-kcl-s + kcl-s-tcl + s-tcl-t + tcl-t-sil**

The use of triphones allows much greater degree of context modelling over monophones. However the number of HMM models to be trained also increases significantly. This can make training problematic, since some triphones may rarely appear in the training data or not at all. To solve this problem, various clustering techniques have been proposed. Here, acoustically similar triphones are grouped together into a common training pool.

3.1.6 Language models

Acoustic models are designed to give a hypothesized word sequence given observational evidence. However, this does not guarantee a transcription that is syntactically correct or meaningful. To provide this functionality, language models are used. Here, we will consider the N-gram language model whose function is to give a probability of a hypothesized word sequence [142]

$$P(w_1, w_2, w_3, ..., w_M) \approx \prod_{i=1}^{M} P(w_i|w_{i-1}, w_{i-2}, ...wi - N). \tag{3.27}$$

The N-gram language model is an approximation that exploits language ergodicity. Specifically we have assumed the probability of observing any single word is dependent only on the preceding N words. Naturally, as N increases, the approximation improves. However, for large vocabulary systems a high-gram language model is often impractical to implement. For these reasons, N is usually kept to a maximum of 4.

The use of a language model can significantly improve automatic recognition accuracy. This is especially true for recognition systems that are highly structured, with small vocabularies or highly constrained grammar. However, even larger vocabulary systems benefit from language models. This is because most languages are themselves highly structured.

3.1.7 Evaluation of automatic speech recognition performance

To measure recognition performance, a word error rate (WER) metric is commonly used [69, 115]

$$\text{WER} = \frac{D + S + I}{N} \times 100\%, \tag{3.28}$$

where D, S and I are the number of deletion, insertion and substitution errors respectively. N is given as the total number of spoken words. Specific errors can be described as follows:

1. Deletions: the system failed to transcribe a spoken word.
2. Insertions: the system erroneously added a word that was not spoken.
3. Substitutions: the system incorrectly labelled a spoken word.

Naturally, a lower WER is better, with a WER of zero indicating perfect recognition performance. While WER is one of the more popular performance metrics, it should not be interpreted as a perfect measure of global performance. This is especially true when measuring the WER for small and/or constrained testing stimulus, where WER results may not be representative of general performance. Another issue with the WER metric concerns the use of insertion errors. In (3.28) we have treated insertion errors with the same weight as substitutions and deletions. However this may not be optimal in all applications and may lead to skewed results. In addition, since there is no limit to the number of insertion errors, it is possible for the WER to be greater than 100%.

3.2 Front-end feature extraction

The front-end stage of ASR is concerned with parameterizing speech – the process of extracting useful information from the raw acoustic signal. In any machine learning application, the feature parameterization should satisfy the following criteria:

- **Accuracy**. The features must capture the information relevant to the recognition task.
- **Compactness.** The features should not contain superfluous information.
- **Robustness.** The features should be invariant to extraneous factors, such as environmental noise and recording equipment. The features should also be reliably reproduced when given similar raw data.

Developing speech features that satisfy every criteria is a difficult task, and is currently a very active area of research. Despite this, most speech parameter sets attempt to capture the same underlying information, albeit with various approaches. While human speech contains many measurable traits, it has been found that the vocal tract shaping conveys the most intelligibility. This shaping corresponds to the filter component of the source-filter model described earlier (see Section 2.1). In spectral-domain, the convolutional relationship of the source-filter model becomes multiplicative

$$X(m,k) = X_e(m,k)H(m,k), \tag{3.29}$$

where $X(m,k)$ is the short-time spectral speech, $X_e(m,k)$ is the excitation signal (the source), and $H(m,k)$ is the vocal tract response (the filter). If we may assume independence between the vocal tract shaping and excitation, we may express the short-time spectral power of the speech as

$$P(m,k) = P_e(m,k)|H(m,k)|^2. \tag{3.30}$$

The fine structure within $P(m,k)$ can be primarily attributed to the excitation $P_e(m,k)$, while the overall spectral shape is largely a result of the vocal tract response

$H(m,k)$. Thus, the vocal tract response $H(m,k)$ can be approximated as the short-time magnitude spectrum envelope. It is this trait that the majority of feature extraction algorithms attempt to capture. The rest of this section details some common front-end speech processing algorithms.

3.2.1 Pre-emphasis

Pre-emphasis is a common speech processing tool. Higher frequency formants emerging from the glottis tend to have lower amplitudes than their lower frequency counterparts. This is due to the natural physiology of the vocal tract that results in attenuation of higher frequencies. In order to equalize the dynamic range of the signal, a simple first order FIR filter may be used as a compensator. In Z-domain, the filter is given as

$$H(z) = 1 - \alpha z^{-1} \quad \text{where } 0.9 \leq \alpha < 1. \tag{3.31}$$

The pre-emphasized speech signal $x'(n)$, is given as

$$x'(n) = x(n) * h(n). \tag{3.32}$$

3.2.2 Filterbank energies

Before the advent of efficient Fourier transform algorithms, filterbank energies (FBEs) were a popular method for spectral-domain analysis and parameterization. FBEs give an unambiguous and direct method for parameterizing the short-time spectral magnitude envelope. Despite having an clear physical relationship with the human auditory system, FBEs have mostly fallen out of favour for other, more recent feature sets. The two main reasons for this are [100]:

1. The requirement of 20-40 filterbanks increases the complexity of the ASR classification backend, slowing the recognition process.

2. The filterbank energies tend to be highly correlated. This means a full

covariance matrix is often required to model filterbank energies, vastly increasing the amount of computation needed for learning and/or recognition.

Although FBEs have been largely replaced by newer spectral analysis techniques, they form an intermediary step in many subsequent feature derivation algorithms.

3.2.3 Linear prediction

Linear prediction was first applied in the speech domain by Atal and Hanauer in 1971 [11]. Although its popularity as a feature extractor has faded over the past few years, it is still present in many audio transmission and compression algorithms. In auto-regressive (AR) linear prediction, a current sample $x(n)$ is predicted using a linear combination of the previous p samples

$$\hat{x}(n) = -\sum_{k=1}^{p} a_{p,k} x(n-k), \tag{3.33}$$

where $a_{p,k}$ is commonly referred to as the linear prediction coefficients (LPCs). The prediction error, or *residual*, is given by

$$e(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^{p} a_{p,k} x(n-k). \tag{3.34}$$

Alternatively, in the z-transform domain

$$E(z) = \left[1 + \sum_{k=1}^{p} a_{p,k} z^{-k}\right] X(z), \tag{3.35}$$

where $X(z)$ and $E(z)$ are the z-transforms of $x(n)$ and $e(n)$ respectively. If we assume the residual to be uncorrelated and *white* in nature, then $E(z)$ may be replaced by a constant. This lets us model the vocal tract response as an p'th order all-pole system $H(z)$

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_{p,k} z^{-k}}. \tag{3.36}$$

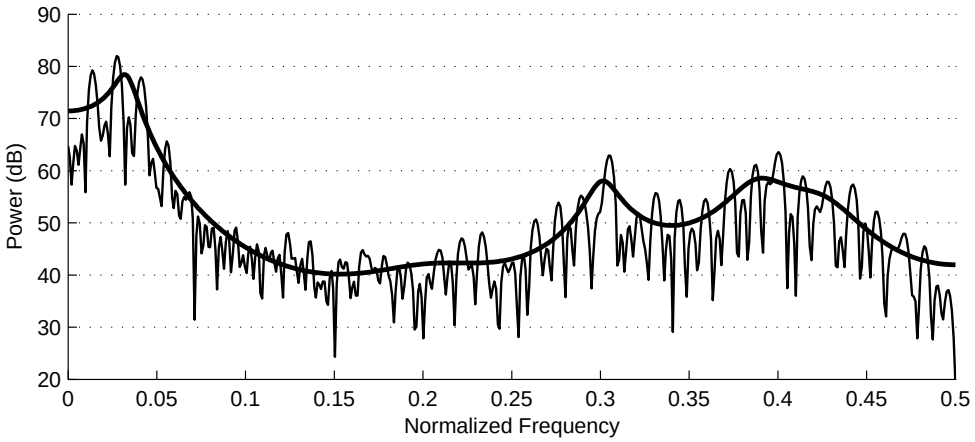

Figure 3.3: Linear prediction coefficients for modelling the short-time spectral amplitude envelope. Periodogram estimation (thin line) and LPC modelling (thick line) for a speech segment. Typical LPC analysis attempts to capture the short-time spectral envelope and not the fine detail pitch harmonics.

A number of methods exist for finding the LPCs [69], with the most popular methods mathematically minimizing the energy of the error $e(n)$. Fig. 3.3 shows the relationship between the spectral power estimation and auto-regressive model estimation. Here we can see that the LPC AR model captures the overall formant locations quite well, discarding the finer pitch information.

LPCs have seen numerous modifications since their introduction to the speech processing field, including the addition of psychoacoustic considerations [66]. Despite being largely replaced by Mel-frequency cepstral coefficients, LPCs are still present in many speech processing applications [139] and form an intermediary step in several advanced algorithms such as speech based Kalman filters [123].

3.2.4 Cepstral coefficients

The cepstrum is a useful homomorphic transformation used extensively in speech processing. In general, a homomorphic transformation is one that transforms a

convolution into an addition, i.e.

$$y(n) = x(n) * h(n) \longrightarrow \hat{y}(n) = \hat{x}(n) + \hat{h}(n) \tag{3.37}$$

Two common cepstral representations are the linear prediction cepstral coefficients (LPCCs) and Mel-frequency cepstral coefficients (MFCCs).

Linear prediction cepstral coefficients

Following the introduction of LPCs, linear prediction cepstral coefficients (LPCCs) were proposed by Atal in 1974. Given the set of LPCs coefficients $a_{p,k}$, there exists a simple algorithm for generating a set of cepstral coefficients (and vice-versa).

It is possible to produce an infinite set of cepstral coefficients for any finite set of $a_{p,k}$. However, the cepstral sequence rapidly decays toward zero, so a dozen is usually sufficient to provide a reasonable approximation. Although derived explicitly from LPCs, LPCCs tend to perform slightly better in noisy environments.

Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCCs), incorporate the perceptually based Mel scale into cepstral processing [30]. Since their introduction in 1980, MFCCs (and its variants) have become the dominant feature set for ASR. It is interesting to note that this is true for both speech and speaker recognition.

MFCCs themselves are given as the cepstrum of Mel warped, log energy filterbanks. Mel spaced filters (see Fig. 3.4) are applied to the short-time power/energy spectrum to give an energy measure for each Mel filterbank. The choice of the Mel warped frequency scale is directly motivated by psychoacoustic experiments, which shows the ear operating in a similar manner. However, for feature generation we typically use only 20-30 filterbanks instead of the hundreds hypothesized for the human cochlea. Log filterbank energy $L(m, q)$ for the m'th

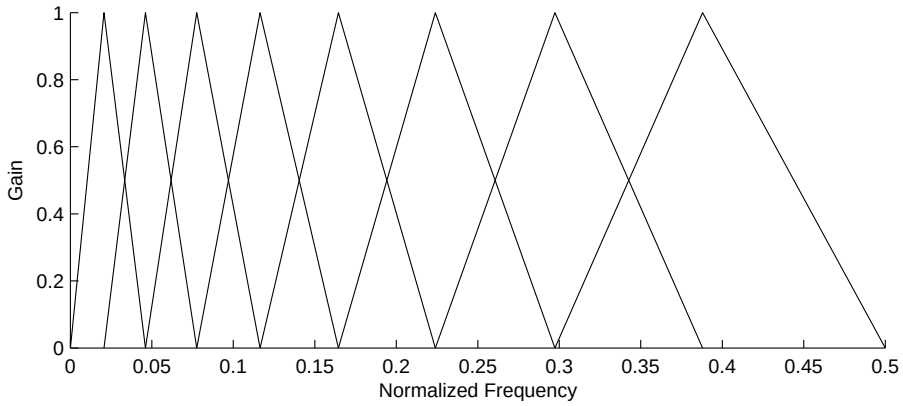

Figure 3.4: Mel-spaced filterbank for a sampling frequency of 8 kHz.

analysis frame and q'th filterbank is given by

$$L(m,q) = log\left[\sum_{k=0}^{K-1} \mid X(m,k) \mid^2 h(k,q)\right] \quad \text{for } q = 0,1,2...Q-1, \tag{3.38}$$

where $h(k,q)$ is the gain for the q'th filterbank and k'th discrete frequency bin, K is the total number of discrete frequency bins and $X(m,k)$ is the DSTFT of the speech signal. An inverse discrete Fourier transform is used to convert the filterbank energies into the real cepstrum. However, since the power spectrum is an even sequence, a DCT-II is generally used instead. The q'th Mel-frequency cepstral coefficient of the m'th frame, $c(m,n)$ is given as

$$c(m,q) = \sum_{q'=0}^{Q-1} L(m,q')cos\left(\pi n(q'-1/2)/Q\right), \tag{3.39}$$

where Q is the total number of filterbanks. A block diagram of the MFCC derivation process is given in Fig. 3.5.

While not psychoacoustically motivated, the DCT has a number of mathematical benefits. Firstly, filterbanks are heavily overlapped, meaning higher order coefficients tend to have very small energies. Because of this it is usual to

[3]Pre-emphasis and liftering are optional procedures.

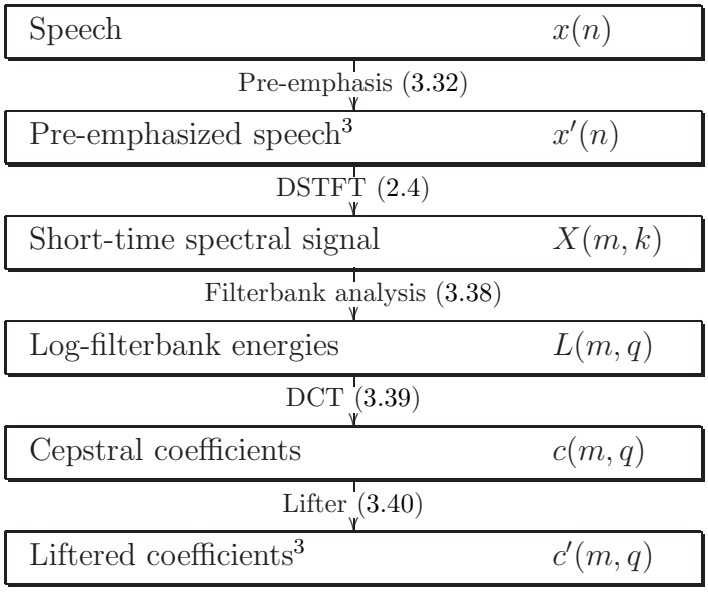

Figure 3.5: Overview of typical MFCC derivation.

simply discard the higher order coefficients, leading to a more compact feature set. Furthermore, it is the lower order cepstral coefficients that are more sensitive to overall spectral shape, while higher order coefficients tend to pick up highly varying, noise-like features. By removing higher order coefficients we can improve the accuracy and robustness of the feature vector. The second advantage of the DCT is that the coefficients it produces have very little cross-correlation. This allows us to approximate MFCC covariances with diagonal matrices – vastly improving the efficiency of recognition algorithms. Figs. 3.6a) and 3.6b) illustrate the covariance diagonalization. For the log-filterbank energies (Fig. 3.6a)), there is significant cross-correlation between filterbanks – indicated by large R^2 in the off-diagonal elements. This occurs even when filterbanks are not directly overlapped. For the MFCCs however (Fig. 3.6b)), there is much less cross-correlation.

While the DCT introduces a number of mathematical conveniences, it does remove physical meaning from individual coefficients. Being a linear combination of several log-filterbank energies means cepstrum coefficients no longer represent a localized frequency region of the spectrum.

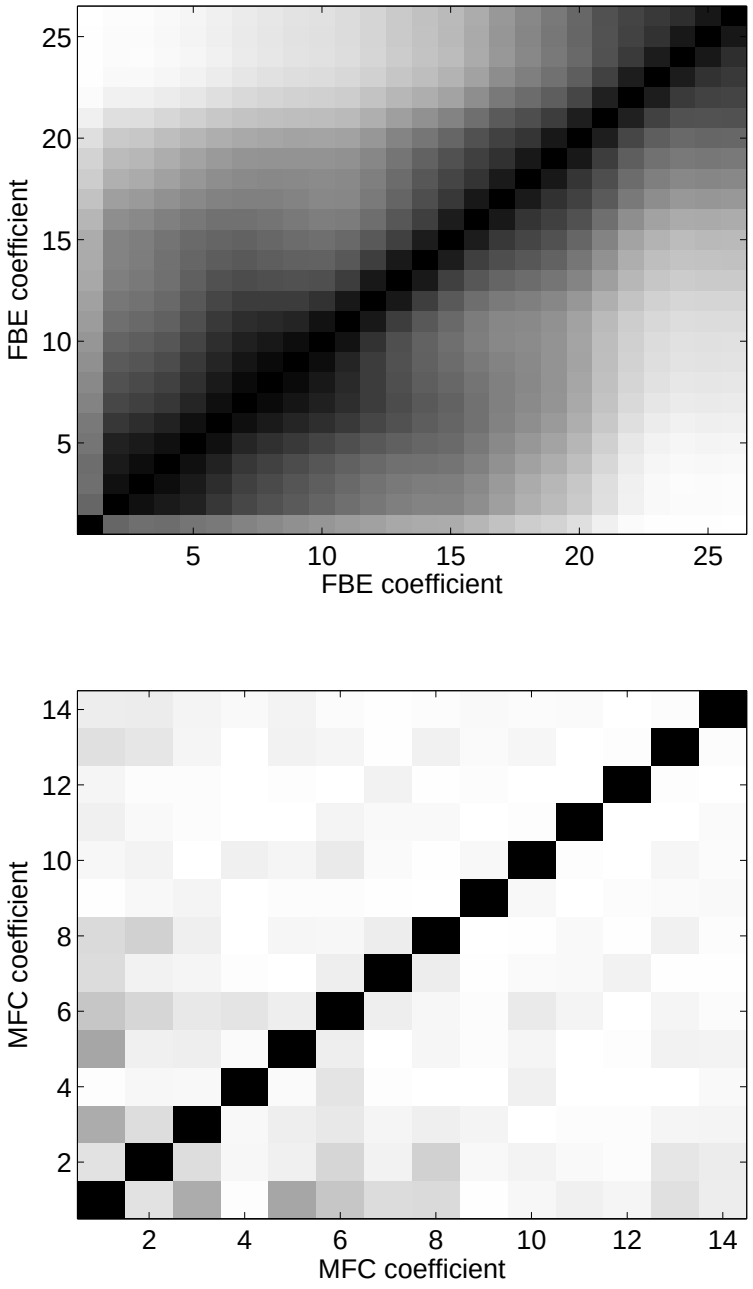

Figure 3.6: Top figure: R^2 coefficient matrix for 26 log-filterbank energies, bottom figure: R^2 coefficient matrix for 14 Mel-frequency cepstrum coefficients. White pixels represent zero correlation, while black pixels represent perfect ($R^2 = 1$) correlation. R^2 coefficients were generated from 16 kHz sampled clean speech from the RM speech corpus [110].

Cepstral mean normalization

As stated earlier, convolutional effects within the spectral-domain become additive in the cepstral domain. This gives cepstrum based features a unique advantage in compensating convolutional effects. The filter response of the recording equipment and transmission lines are generally considered to be stationary. This means that within the cepstral domain, these effects become a constant additive effect. Thus we may remove these effects by subtracting a long term cepstral average. At its simplest, we may simply remove the cepstral average of the entire utterance – cepstral mean subtraction (CMS). Cepstral mean normalization (CMN) is similar, but relies on a long term average making it suited for online, continuous speech recognition. Typically, 2-3 seconds of speech are required to generate a sufficiently accurate cepstral average. CMS and CMN have thus far proven to be very successful for increasing the robustness of ASR to channel effects. Given their quick and efficient implementation, they are commonly employed as a standard cepstral feature postprocessor.

Cepstral liftering

Liftering is filtering within the cepstral domain. Higher order cepstral coefficients tend to be smaller than the lower order coefficients [69, 114]. Because of this, we often rescale the coefficients to have similar magnitudes. Here, liftering is provided by the following formula [142]

$$c'(n) = \left(1 + \frac{L}{2}\sin\left(\frac{\pi n}{L}\right)\right) c(n), \tag{3.40}$$

where $c(n)$ are the original coefficients, $c'(n)$ are the liftered coefficients and $L \approx 22$, is a tuneable parameter to control the degree of liftering.

3.2.5 Dynamic coefficients

First order HMMs assume speech to be sequence of frames with instantaneous transitions. In practice, speech does not evolve instantaneously, so it can be useful to augment static features with differential features [69, 142]. While the use of dynamic coefficients clearly violates the statistical assumptions of the HMM, they can offer large performance gains for many speech recognition tasks. This is especially the case for large vocabulary, speaker independent systems, since static coefficients alone are often not enough to adequately separate larger dictionaries.

A number of methods exist for calculating dynamic coefficients, though linear regression is most common. First order *delta* coefficients can be given by the following [142]

$$\Delta c(t) = \frac{\sum_{\tau=-D}^{D} t\,[c(t+\tau) - c(t-\tau)]}{2\sum \tau^2}, \tag{3.41}$$

where D is a parameter that specifies the regression window size. These coefficients are then appended onto the existing static feature vector. Second order *delta-delta* or *acceleration* coefficients can be gained by applying the above equation to the delta coefficients instead of the static coefficients. Higher order coefficients may be generated in a similar fashion, though doing so yields diminishing performance gains at the cost of higher dimensionality feature vectors.

3.3 Improving robustness of ASR

State of the art ASR can exhibit impressive recognition performance when operated in ideal laboratory conditions. Unfortunately, performance tends to degrade substantially when ASR is used in real world environments. This degradation is caused by acoustic model mismatch. Here, we use the term mismatch to describe any difference between the acoustic environment the ASR system was trained on, and the acoustic environment the ASR system is actually deployed in. Mismatch can involve several phenomena:

- Additive uncorrelated noise: background noises such as cars, other people

talking, electrical induction noise.

- Additive correlated noise: echoes and reverberation.
- Convolutional mismatch: different responses for the microphones/transmission channels used.
- Inter-speaker variability: variability arising from different people pronouncing the same words.
- Intra-speaker variability: changes to speech based on speed, inflections and emotional state (such as stress).
- Lombard effect: changing of speaking style in response to the environment – such as talking louder when it is noisy.

Since we do not have complete control over the testing environment, a degree of mismatch is often unavoidable. Given the severe performance penalty associated with mismatch, increasing robustness is a very important ASR problem. In this dissertation, we use a simplified noise distortion model that encompasses only additive background and convolutional mismatch. Fig. 3.7 is a block diagram of this simplified model. To limit the performance degradation caused by acoustic model mismatch, several approaches have been adopted in the literature:

1. Choosing a feature set that is inherently robust. For example, the MFCC parameterization gives greater robustness than LPC parameterization.
2. Enhancing the test speech prior to classification. Typically this enhancement is performed prior to the feature extraction stage, though additional processing may be performed directly on the feature level.
3. Removing/regenerating features deemed corrupted.
4. Adapting the back-end recognizer to better match the perceived operating environment.
5. Training the recognizer on noisy stimulus. This approach is often referred to as multi-style training.

These approaches are covered briefly in the remainder of this section.

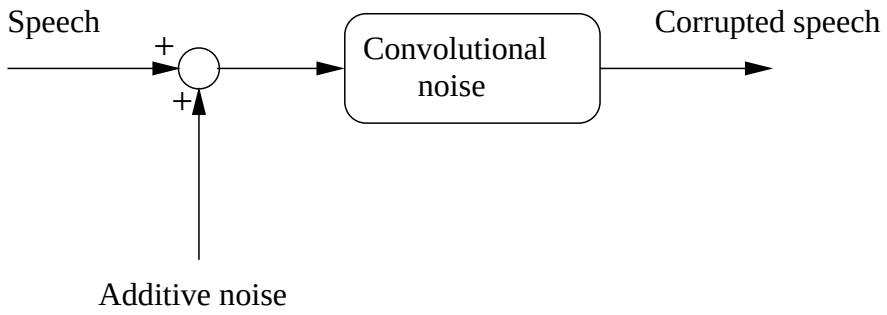

Figure 3.7: Simplified noise distortion model incorporating additive and convolutional distortions.

3.3.1 Signal enhancement

One of the simplest methods for additive noise robustness is to apply a speech enhancement algorithm prior to feature extraction. There are many such techniques available, each of which has differing performance / complexity tradeoffs. The following techniques assume an uncorrelated, additive noise distortion. Here the observed time-domain signal $y(n)$ is a summation of the clean signal $x(n)$ and noise signal $d(n)$

$$y(n) = x(n) + d(n). \tag{3.42}$$

Spectral subtraction

Spectral subtraction is a simple but effective technique for reducing the effects of additive noise. Assuming the clean speech and noise are uncorrelated, the maximum *a posteriori* (MAP) estimation of clean speech spectral power $|\hat{X}(m,k)|^2$ is given by

$$|\hat{X}(m,k)|^2 = U\left[|Y(m,k)|^2 - |D(m,k)|^2\right], \tag{3.43}$$

where $X(m,k)$, $Y(m,k)$ and $D(m,k)$ are the DSTFTs of clean speech, noisy speech and noise respectively. The function $U[x]$ is included to ensure the estimated power

spectrum cannot become negative

$$U[x] = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise;} \end{cases} \tag{3.44}$$

In many applications, it can be useful to define a spectral amplitude gain for the enhancement system. Here, an estimation of the clean spectrum $\hat{X}(m,k)$ is given as

$$\hat{X}(\omega,t) = G(m,k).Y(m,k), \tag{3.45}$$

where $G(m,k)$ is the spectral amplitude gain. The spectral amplitude gain for the spectral subtraction framework, $G_{SS}(m,k)$ is given by

$$G_{SS}(m,k) = \sqrt{\frac{U\left[|Y(m,k)|^2 - |D(m,k)|^2\right]}{|Y(m,k)|^2}}. \tag{3.46}$$

Perhaps the biggest drawback of the spectral subtraction method is the appearance of musical noise. Musical noise manifests as a series of randomly distributed, sharp spectral peaks in the noise residual. This produces perceptually annoying acoustic artefacts, severely degrading subjective speech quality. Despite its shortcomings, spectral subtraction remains a popular enhancement method due to its simplicity and ability to integrate well with other enhancement methods. In addition, the basic spectral subtraction framework has seen numerous modifications and additions, both to decrease the appearance of musical noise, and improve its performance as an ASR front-end. Common additions include over-subtraction and spectral flooring [80] – ad hoc additions that have sacrificed the MAP criterion in favour of subjective listening quality.

Spectral Wiener filtering

The spectral Wiener filter is an improvement over spectral subtraction. The spectral Wiener filter produces an estimate $\hat{X}(m,k)$, that minimizes the mean square error

to the true clean speech $X(m,k)$. While less efficient than spectral subtraction, the Wiener filter often gives higher quality results. Being linear, the Wiener may also be implemented as a time-domain filter. However, we will consider its short-time spectral representation here. The spectral amplitude gain for the spectral Wiener filter is given by

$$G_{SW}(\omega,t) = \frac{|X(m,k)|^2}{|X(m,k)|^2 + |D(m,k)|^2}. \tag{3.47}$$

The Wiener filter is a very popular enhancement regime that has found use in a variety of fields. However, like the spectral subtraction method it is often prone to musical noise.

Other minimum mean square error estimators

The Wiener filter is a minimum mean square error (MMSE) estimator in the spectral-domain. Since spectral phase is considered of minor importance (for speech intelligibility), we may choose instead to derive MMSE estimates in the short-time spectral amplitude domain [39]. On a similar track, the MMSE short-time log-spectral amplitudes may also be estimated [40], bringing the estimation criteria closer in line to current psychoacoustic hearing models. Both the spectral amplitude estimator and log-spectral amplitude estimators are given as non-linear filters operating in the spectral-domain. Though having a much higher computational cost than either the spectral subtraction and spectral Wiener filter methods, the spectral amplitude and log-spectral amplitude estimators are very popular enhancement methods. This popularity largely derives from their relative lack of musical noise artefacts. This class of enhancement techniques is investigated in greater detail in Chapters 8 and 9.

Feature enhancement using speech models

Model based feature enhancement techniques use prior trained speech models to adapt noisy speech features to better suit the trained acoustic model [70, 87, 88, 130]. One model based approach involves direct linearization of a log-filterbank/cepstral

domain noise model with vector Taylor series (VTS) expansion [87]. VTS uses a linearized noise distortion model in conjunction with an *a priori* clean speech model to enhance speech features. For the *a priori* speech model, a Gaussian mixture model (GMM) trained on a clean speech corpus is generally used. With a sufficiently large number of mixtures, VTS has been shown to give good recognition performance for several speech tasks.

Similar GMM based feature enhancement algorithms also exist. One particular implementation is missing feature theory (MFT) imputation [28]. Here, noisy/unreliable features are first identified and flagged for imputation. The unreliable features are imputed/regenerated using a combination of an *a priori* clean speech model and the remaining, reliable features. This method of feature enhancement prevents highly corrupt features from overly biasing the recognition back-end. MFT imputation works most naturally with log-filterbank energies, since narrow-band noise will only corrupt a small number of filterbanks. Cepstral coefficients on the other hand, leverage information from the entire frequency axis. This means narrow-band additive noise will corrupt the majority of the cepstral coefficient feature vector. Of course, the cepstral domain still offers significant advantages when dealing with convolutional distortion.

In HMM model based feature enhancement [130], an *a priori* HMM is used for enhancement. Here clean speech estimates are constructed as a weighted sum of individual, state-conditioned estimates. Structurally, this form of model adaptation is similar to the parallel model combination (PMC) method. However, the authors have noted that much simpler models are required for the front end enhancement. This makes HMM feature enhancement a more scalable approach, since complex recognition back-ends are often required for large vocabulary systems.

3.3.2 Model based adaptation

In this section we briefly cover several HMM adaptation based techniques used for robust ASR. These algorithms directly modify the parameters of the HMM

back-end recognizer such that they more closely resemble the operating acoustic environment. The parameters adapted can include state distributions (the GMM observation probability density functions), or the entire HMM topology (number of states, number of mixtures, transition probabilities etc) [3, 53, 56]. Model adaptation based techniques have been shown to increase ASR robustness across a number of recognition tasks. However, use of these techniques is not always practical as they usually scale poorly. This can make them ill-suited for large and/or dynamic vocabulary tasks.

Maximum a posteriori adaptation

Maximum *a posteriori* (MAP) adaptation [56] seeks to adapt the GMM parameter set for continuous density HMMs. Let $\boldsymbol{\theta}$ denote the parameter vector for the acoustic model and $\boldsymbol{x}$ denote the labelled adaptation data. The MAP estimate is defined as the mode of the posterior PDF; i.e.,

$$\begin{aligned} \boldsymbol{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\, P(\boldsymbol{\theta}|\boldsymbol{x}) \\ &= \underset{\theta}{\operatorname{argmax}}\, P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta}). \end{aligned} \tag{3.48}$$

Given a trivial (constant) prior PDF $P(\boldsymbol{\theta})$, this estimate becomes equivalent to the more familiar maximum likelihood (ML) approach. The MAP estimate can be shown to be a weighted sum of the prior parameter set and the ML evidence based parameter estimate. Given a large amount of adaptation data, this means the MAP estimate converges to the ML estimate, while for limited training data, the MAP estimate will leverage more of the prior PDF. This allows the MAP estimate to cope with comparatively small amounts of training data. Of course, this benefit requires an accurate estimate of the prior PDF; a task that is often difficult. Another drawback of this technique is the requirement of labelled adaptation data for every word/phoneme model. If labelled data is not available, transcriptions can be autonomously estimated online with preliminary ASR.

Maximum likelihood linear regression

Like MAP adaptation, in maximum likelihood linear regression (MLLR), we seek to adapt the GMM state models to suit a new acoustic environment. If we only consider the GMM means, the adaptation is given by the linear transformation

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\xi}, \tag{3.49}$$

where $\hat{\boldsymbol{\mu}}$ is the estimate of the noise corrupted means, $\boldsymbol{A} \in \mathbb{R}^{N\times(N+1)}$ is the transformation matrix to be learned and $\boldsymbol{\xi}$ is the extended clean-condition mean vector given by

$$\boldsymbol{\xi} = \begin{bmatrix} 1 \\ \boldsymbol{\mu_c} \end{bmatrix}. \tag{3.50}$$

The transformation matrix $\boldsymbol{A}$ is learned in the standard ML approach using speech from the new acoustic environment. A similar transformation may also be used to adapt the model variances to gain another (albeit smaller) performance increase. One disadvantage of this approach is that it uses a linear model to describe what is normally non-linear (in the cepstral or log-filterbank domains) noise distortion. However the MLLR approach does have some advantages over MAP. The main advantage being able to adapt the entire parameter set via a single transformation matrix. This means all parameters may be adapted, regardless of whether they are present in the adaptation data. However, like the MAP adaptation, the requirement for labelled data in the new environment can be impractical.

Parallel model compensation

Parallel model compensation [53] seeks to directly modify cepstral feature HMM parameters. PMC departs from the MAP and MLLR approach by dropping the requirement for speech adaptation data and using noise data instead – the latter being a simpler quantity to obtain. In PMC, clean-speech HMMs are combined with noise HMMs to generate a noisy speech HMM that is more suitable for the new

acoustic environment. Since background noise is considered additive in the spectral-domain (and not the cepstral domain), several conversions must take place to adapt the cepstral feature HMM parameters. Exact adaptation formulas were given as a set of numerically integrated solutions. However, closed form approximations for adapting static and dynamic model parameters were also derived. To arrive at these solutions, a number of mathematical simplifications were made. Most notable of these was the assumption that the addition of two log-normally distributed variables yields another log-normally distributed variable.

The model compensation can involve significant computational overhead if there are large numbers of Gaussian mixtures used to describe the clean speech and noise. To address some of these issues, a data-driven version of PMC has been proposed. Here random samples from the clean speech training set are combined with noise models to produce simulated noisy samples. These samples are then used to estimate the noisy speech HMMs.

3.3.3 Multi-style training

In the case where we can pre-determine the operating acoustic environment, multi-style training has been shown to be very effective [31]. Here, training speech is degraded with the same noise that would appear in the operating environment, eliminating a large degree of acoustic mismatch. Obviously, this technique performs best when we have good knowledge of the operating environment.

For less constrained systems, it is common to train a multi-style system on multiple noise types, each at several intensities. Such a process does increase the robustness of the system to slightly mismatched and/or novel noises. However the larger data collection and training requirements can be onerous. In some cases it may simply be impractical to model all possible operating environments. Furthermore, while additional robustness is gained, the increased generality of the acoustic models may harm clean condition performance.

3.3.4 Other techniques

In most ASR back-ends, performance is degraded substantially when a corrupt feature is present. Intuitively, we would only want quality data to be used in the recognition decision, comparing relevant speech features, while minimizing/ignoring the unreliable data. Missing feature theory (MFT) marginalization addresses this problem [12, 28, 116]. Like MFT imputation, unreliable features are identified based on their local SNR and dealt with prior to recognition. At its simplest, we may simply ignore the unreliable features during recognition. Mathematically, the output probability of feature vector $\boldsymbol{y}$ using a particular GMM mixture M (with diagonal covariance), becomes a marginal probability over the reliable feature dimensions.

$$p(\boldsymbol{y}|M) = \prod_{d=0}^{D-1} \left[\gamma_d \, \mathcal{N}\left(y_d|\mu_{m,d}, \sigma^2_{m,d}\right) + (1-\gamma_d)\right], \tag{3.51}$$

where y_d is the d'th of D total features in feature vector $\boldsymbol{y}$. γ_d is the mask variable that is equal to 1 when the feature y_d is reliable, otherwise 0. While simple to implement, MFT marginalization does suffer some drawbacks. Most notable is an increase in word insertion errors for ASR tasks. This is because of the differing number of reliable/unreliable features on a frame-to-frame basis. A few alternate approaches to feature removal exist. Firstly, instead of removing the unreliable features, we may regenerate/impute them with something more suitable [72]. The imputation approach is similar to using an *a priori* speech model for enhancement, though with a simplified binary hypothesis to decide which features are regenerated.

In another related approach, we may attribute a finite non-zero variance to all features. The feature variance is then incorporated in the back-end recognizer. Here, features with high variance contribute comparatively less to the recognizer decision. This process is often referred to as uncertainty decoding [35, 130].

Part I

Use of the short-time Fourier phase spectrum for speech processing

Chapter 4

Reconstruction of magnitude and phase spectra

4.1 Introduction

The characterization of signals from partial specification in the time-domain, frequency-domain, or combinations thereof has been extensively studied in the past. In general, the Fourier phase and magnitude spectra are independent functions, with both being required to uniquely specify a time-domain signal. However, when certain properties of the signal are known, the coupling between the magnitude and phase spectra can be quite strong. This is especially the case when we choose to over-sample the Fourier transform; that is, padding a signal with zeros prior to spectral transformation. In the following sections we investigate the direct mathematical relationships between the magnitude and phase spectra, and how these relationships may be used for signal reconstruction. In particular, we revisit the problem of regenerating spectral magnitude information from spectral phase, and vice versa. The rest of this chapter is outlined as follows. In Section 4.2, we introduce the general Gerchberg-Saxton spectra regeneration algorithm. In Section 4.3, we investigate

the specific problem of regenerating magnitude spectra from phase information. The alternate problem of regenerating phase spectra from magnitude information is then visited in Section 4.4. In the process of this investigation, we highlight the shortcomings of the standard Gerchberg-Saxton algorithm. We examine these shortcomings and propose a modified algorithm to speed up the phase retrieval process. Lastly in Section 4.5, we present some concluding remarks.

4.2 Iterative reconstruction

One of the most profoundly studied methods for signal reconstruction is the iterative Gerchberg-Saxton (GS) algorithm. While originally proposed in the field of optics, the GS algorithm is readily generalized to numerous speech regeneration applications. The GS algorithm works by iteratively solving two sets of constraints – spectral-domain constraints and time-domain constraints. Spectral-domain constraints include forcing a signal to match a known magnitude or phase specification, while time-domain constraints are used to enforce causality and duration. In general, it is not possible to simultaneously solve both the spectral-domain and time-domain constraints. The GS algorithm works by iteratively solving each constraint in an alternating fashion. Provided the mathematical requirements for reconstruction have been met, total error (w.r.t both the spectral and time-domain constraints) will monotonically decrease with each iteration until a local optimum has been found. The basic algorithm is outlined below.

The iterative framework allows signal reconstruction under a number of different time and frequency specifications. For a comprehensive list of mathematical relationships between the magnitude and phase spectra, the reader is referred to the following papers [62, 65, 68, 96, 111].

Algorithm 1 Generalized Gerchberg-Saxton signal reconstruction algorithm

1: $k \leftarrow 0$
2: Set error threshold η
3: Provide functions $f(.)$ and $g(.)$ that map to spectral subspaces. The subspace which $f(.)$ maps to must satisfy the time-domain constraints, while the subspace that $g(.)$ maps to must satisfy spectral-domain constraints[1].
4: Provide an initial guess of the spectral-domain signal: $\hat{\mathbf{X}}_k$
5: Enforce time-domain constraint: $\hat{\mathbf{X}}'_{k+1} \leftarrow f(\hat{\mathbf{X}}_k)$
6: Enforce spectral-domain constraint: $\hat{\mathbf{X}}_{k+1} \leftarrow g(\hat{\mathbf{X}}'_{k+1})$
7: If convergence met: $\| \hat{\mathbf{X}}_{k+1} - \hat{\mathbf{X}}_k \|^2 < \eta$, exit
8: $k \leftarrow k + 1$.
9: Go to step 5

4.3 Regeneration of magnitude spectra from phase spectra

Despite the ubiquity of the magnitude spectrum in speech processing, a time-domain signal can be described (within scale) solely from phase information. While the Gerchberg-Saxton algorithm provides a general framework for signal reconstruction, a more interesting linear relationship exists between the time-domain signal and its Fourier phase spectrum. This linear relationship was originally detailed within [65]. In the following section, this relationship is recast into a more modern and tractable nullspace form.

For a time-domain signal $\mathbf{x} \in \mathbb{R}^{N\times 1}$, and corresponding spectral-domain signal $\mathbf{X} \in \mathbb{C}^{L\times 1}$ where $L > N$, we have the following mathematical relationship

$$\mathbf{F}^{-1}\mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{F}$ is the discrete L-point discrete Fourier transform (DFT) matrix. The appended zero column vector in (4.1) accounts for DFT over-sampling. We can now construct a diagonal windowing matrix $\mathbf{S}$, such that $\mathbf{S}_{k,k} = 1$ for $k = 0, 1, ..., N-1$, and zero otherwise. Thus, in order for $\mathbf{X}$ to satisfy the finite time duration

[1] Functions $f(.)$ and $g(.)$ must satisfy convergence criteria.

requirements, the following mathematical relation must hold

$$\mathbf{FSF}^{-1}\mathbf{X} = \mathbf{X}, \tag{4.2}$$

or alternatively

$$\mathbf{F}\left[\mathbf{S} - \mathbf{I}\right]\mathbf{F}^{-1}\mathbf{X} = \mathbf{0}. \tag{4.3}$$

This ensures that when an inverse DFT is applied to $\mathbf{X}$, all energy is contained within the first N time-domain samples. The vector $\mathbf{X}$ may be split into magnitude and phase components as follows

$$\mathbf{X} = \boldsymbol{\theta}\mathbf{m}, \tag{4.4}$$

where $\mathbf{m} \in \mathbb{R}^{L\times 1}$ is the spectral magnitude vector and phase matrix $\boldsymbol{\theta} \in \mathbb{C}^{L\times L}$ is given by,

$$\boldsymbol{\theta} = \begin{bmatrix} \exp\left(i\,\mathrm{ARG}[X(0)]\right) & 0 & \cdots & 0 \\ 0 & \exp\left(i\,\mathrm{ARG}[X(1)]\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \exp\left(i\,\mathrm{ARG}[X(L-1)]\right) \end{bmatrix}, \tag{4.5}$$

We now have a clear linear relationship between the phase matrix $\boldsymbol{\theta}$ and unknown magnitude vector $\mathbf{m}$

$$\mathbf{F}\left[\mathbf{S} - \mathbf{I}\right]\mathbf{F}^{-1}\boldsymbol{\theta}\mathbf{m} = \mathbf{0}. \tag{4.6}$$

With the additional constraints of $\mathbf{m}$ being real and non-negative, the above equation can be solved as a nullspace solution.

$$\begin{aligned} &\Re\{\mathbf{F}\left[\mathbf{S} - \mathbf{I}\right]\mathbf{F}^{-1}\boldsymbol{\theta}\}\mathbf{m} = \mathbf{0}, \\ &\mathbf{Bm} = 0 \quad \text{s.t } \mathbf{m} > \mathbf{0}, \end{aligned} \tag{4.7}$$

where $\Re$ is the real component modifier and $\mathbf{B} = \Re\{\mathbf{F}\left[\mathbf{S} - \mathbf{I}\right]\mathbf{F}^{-1}\boldsymbol{\theta}\}$. Generally, with $L > 2N$ (padding with N or more zeros), we can produce a matrix $\mathbf{B}$ such that it has a nullity of 1. This allows for a direct, scaled reconstruction of the magnitude spectrum $\mathbf{m}$. With it, a scaled version of the time-domain sequence can be obtained via use of the known phase (the matrix $\boldsymbol{\theta}$) spectrum and the IDFT.

4.4 Regeneration of phase spectra from magnitude spectra

Regeneration of phase spectra from magnitude spectra has several practical applications. While the magnitude spectrum alone is insufficient to uniquely characterize a signal, it can nonetheless be used to regenerate a (perceptually) useful phase spectrum [62]. This can be useful in cases where we have radically altered the magnitude spectrum and require a consistent phase spectrum to match it. Such a need might arise in the case of heavy speech enhancement and/or modification. Unfortunately, unlike the previous magnitude spectra regeneration problem, there is no simple closed-form method for regenerating spectral phase corresponding to a given magnitude spectrum. While the standard GS iterative algorithm may be used to regenerate the phase spectrum, it tends to be quite slow. Because of this, we investigate the limitations of the GS algorithm, and propose a method for increasing the efficiency of phase retrieval.

4.4.1 Phase regeneration with the Gerchberg-Saxton algorithm

We begin this section by describing the standard Gerchberg-Saxton constraints required for the phase regeneration problem. As with the previous sub-section we start with an unknown time-domain signal $\mathbf{x} \in \mathbb{R}^{N\times 1}$, and corresponding frequency domain signal $\mathbf{X} \in \mathbb{C}^{L\times 1}$ where $L > N$. The time-domain constraint is the same as the spectral magnitude retrieval problem, i.e.

$$\mathbf{X} = \left[\mathbf{FSF}^{-1}\right]\mathbf{X}, \tag{4.8}$$

This constraint forces the IDFT of $\mathbf{X}$, to be a finite, length N time-domain signal. The second, non-linear spectral magnitude constraint is given by

$$|\mathbf{X}| = \mathbf{m}, \tag{4.9}$$

where $|\mathbf{X}|$ refers to element-wise absolute values. This is the constraint that forces the spectral signal $\mathbf{X}$ to match the desired magnitude spectrum $\mathbf{m}$. The deviation from these constraints can be modelled as two root mean square error (RMSE) measures

$$\epsilon_1 = \| \mathbf{FSF}^{-1}\mathbf{X} - \mathbf{X} \|, \tag{4.10}$$

and

$$\epsilon_2 = \| \mathbf{m} - |\mathbf{X}| \| . \tag{4.11}$$

The usual method for minimizing both errors is the Gerchberg-Saxton iteration. This algorithm works by alternately zeroing the errors ϵ_1 and ϵ_2. If we enforce each constraint in a least-norm fashion, we can guarantee convergence. The function $g(\mathbf{X})$ that enforces the magnitude specification constraint is given by

$$g(\mathbf{X}) = \mathbf{m} \circ \exp\left(j\mathrm{ARG}\left[\mathbf{X}\right]\right), \tag{4.12}$$

where $\circ$ refers to an element-wise multiplication. The exponential and ARG function are likewise performed on an element-wise basis. In simple terms, we keep the old phase spectrum of $\mathbf{X}$, but force the updated signal $g(\mathbf{X})$ to conform to the desired magnitude spectrum. The function $f(\mathbf{X})$ that enforces the time-domain duration constraint is given by

$$f(\mathbf{X}) = \mathbf{FSF}^{-1}\mathbf{X}. \tag{4.13}$$

Using $f(\mathbf{X})$ and $g(\mathbf{X})$, the GS algorithm may be used to derive a spectral signal $\hat{\mathbf{X}}$ that has a consistent phase spectrum (see algorithm 1). An illustration of a single GS phase retrieval iteration, is shown in Fig. 4.1. Starting at point $\mathbf{X}_a$ which satisfies

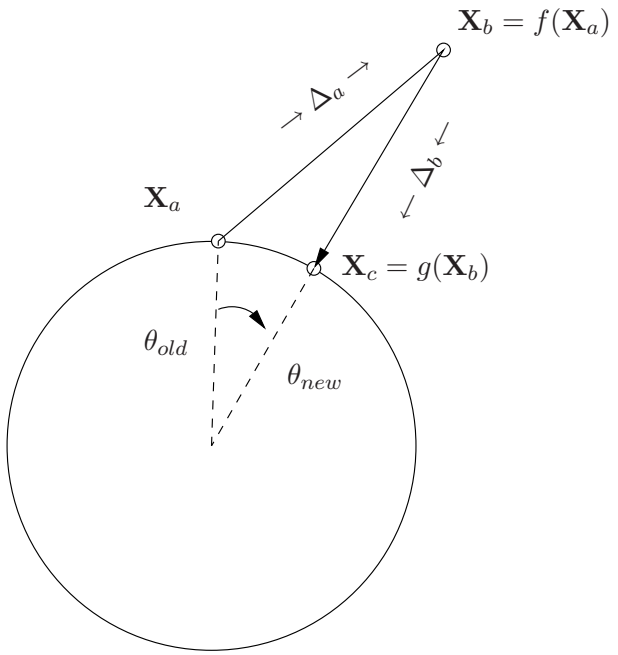

Figure 4.1: Illustration of Gerchberg-Saxton phase retrieval iteration for a single spectral bin. The circle represents the desired spectral magnitude for this spectral bin. In each GS iteration we alternate between satisfying the time-domain constraints ($\mathbf{X}_a \rightarrow \mathbf{X}_b$) and the spectral magnitude constraints ($\mathbf{X}_b \rightarrow \mathbf{X}_c$). At the end of each iteration, the spectral phase $\boldsymbol{\theta}$ is pushed toward a local optima.

the magnitude constraint, we move to position $\mathbf{X}_b = \mathbf{FSF}^{-1}\mathbf{X}_a$ which satisfies the time-domain constraint. This reduces the time-domain constraint error from $\| \Delta_a \|$ to zero. The spectral magnitude error can *at most*, increase by $\| \Delta_a \|$ (and can only do so if Δ_a is normal to the magnitude circle). This means total error cannot increase. From $\mathbf{X}_b$, we move to $\mathbf{X}_c = \mathbf{m} \circ \exp(j\text{ARG}[\mathbf{X}_b])$ to satisfy the magnitude constraint. This reduces the spectral magnitude constraint error from $\| \Delta_b \|$ to zero. The time-domain error can *at most* increase by $\| \Delta_b \|$ (and can only do so if Δ_b is orthogonal to the subspace $\mathbf{F}^{-1}\mathbf{SF}$). Once again, total error cannot increase.

The iterative method outlined above has two primary advantages; first it is guaranteed to converge – meaning the total squared error is strictly non-increasing. Secondly, each iteration is relatively fast thanks to the fast Fourier transform – an $\mathcal{O}(L \log L)$ operation.

4.4.2 An improvement to the Gerchberg-Saxton algorithm

The main drawback of the GS algorithm comes from its slow convergence. Initially, the algorithm reduces error quite rapidly. However, after several iterations the error reduction slows dramatically. An illustration of this behaviour is shown in Fig. 4.2. On the left is an illustration of the standard GS algorithm. Here we have two vector updates: Δ_a which enforces the time-domain constraint, and Δ_b which forces the signal back to the desired magnitude spectrum. While each update is guaranteed to reduce error, there is no motivation to reduce total error rapidly. As a result, the net update of a single iteration ($\Delta_a + \Delta_b$) tends to be very small. That is, in the process of reducing the error of one constraint to zero, we induce almost as much error in other; giving negligible net benefit. On the right of Fig. 4.2, we show a more ideal update policy. Here, a time-domain constraint update Δ_c would move tangentially to the required magnitude circle. While this no longer guarantees the update vector Δ_c to be a least norm update, moving tangentially to the magnitude circle induces very little error in the spectral magnitude constraint. The net result is a larger update (and error reduction) per iteration.

Using this knowledge, we can now construct a new iterative update policy. For the proposed algorithm, we keep the majority of the GS algorithm intact, only altering the time-domain constraint function $f(\mathbf{X})$. Given a signal $\mathbf{X}$ that satisfies the spectral magnitude constraint, we now wish to apply a time-domain constraint update that moves tangentially to the magnitude circle. Our update can then be given by

$$f(\mathbf{X}) = \mathbf{X} + i\boldsymbol{\theta}\mathbf{v}, \tag{4.14}$$

where $\boldsymbol{\theta}$ is the diagonal phase matrix given by (4.5). The vector $\mathbf{v}$ provides a real (positive or negative) scaling factor for moving along the tangent. The proposed function $f(\mathbf{X})$ does not explicitly zero the time-domain constraint error ϵ_1. Obviously though, we would like to make this error as small as possible. The

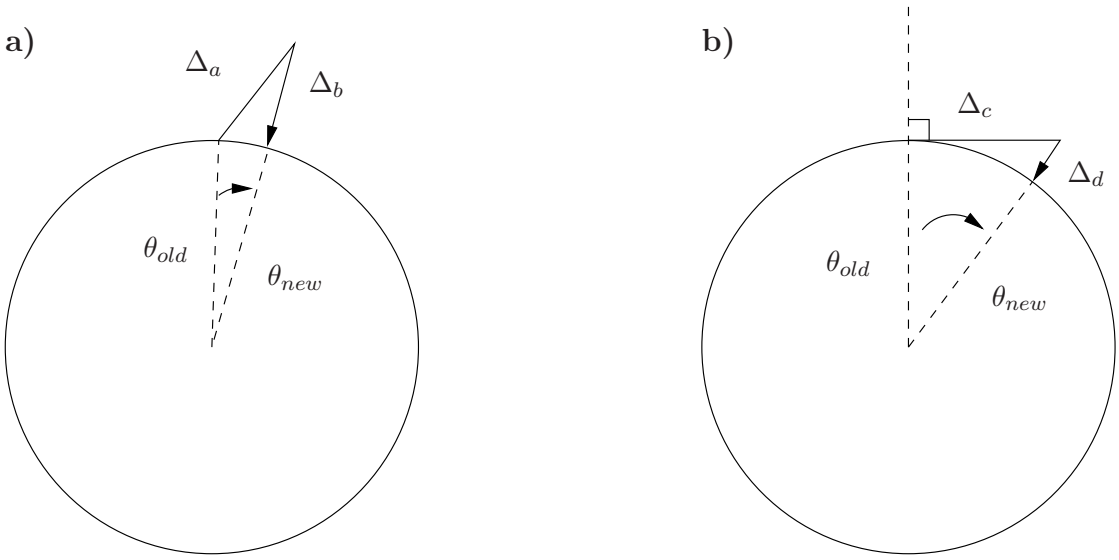

Figure 4.2: Illustration of the Gerchberg-Saxton and proposed phase retrieval algorithms. s) Gerchberg-Saxton update iteration and b) proposed update iteration on the right. In the proposed method, the time constraint update moves tangentially to the magnitude circle. This induces relatively little deviation from the desired magnitude, allowing for a larger update per iteration.

time-domain constraint error after using the function $f(\mathbf{X})$ is given as

$$\epsilon_1 = \| \mathbf{F} \left[\mathbf{S} - \mathbf{I} \right] \mathbf{F}^{-1} \left[\mathbf{X} + i\boldsymbol{\theta}\mathbf{v} \right] \| . \tag{4.15}$$

Naturally, we would also like the keep the induced magnitude constraint error ϵ_2 as small as possible as well. The simplest way to do this is via regularization; i.e., we can approximate ϵ_2 as

$$\epsilon_2 \approx \sqrt{\lambda \mathbf{v}^T \mathbf{v}}, \tag{4.16}$$

where $\lambda \approx 0.02$ is a small positive regularization constant. The total squared error $\epsilon_T^2 = \epsilon_1^2 + \epsilon_2^2$, for the update can now be given by

$$\begin{aligned} \epsilon_T^2 &= \left[\mathbf{F}(\mathbf{S} - \mathbf{I})\mathbf{F}^{-1}(\mathbf{X} + i\boldsymbol{\theta}\mathbf{v}) \right]^H \left[\mathbf{F}(\mathbf{S} - \mathbf{I})\mathbf{F}^{-1}(\mathbf{X} + i\boldsymbol{\theta}\mathbf{v}) \right] + \lambda \mathbf{v}^T \mathbf{v} \\ &= \mathbf{X}^H \mathbf{F}(\mathbf{S} - \mathbf{I})\mathbf{F}^{-1}\mathbf{X} - 2\mathbf{v}^T \Re \left[i\boldsymbol{\theta}^H \mathbf{F}(\mathbf{S} - \mathbf{I})\mathbf{F}^{-1}\mathbf{X} \right] \\ &\quad + \mathbf{v}^T \Re \left[\boldsymbol{\theta}^H \mathbf{F}(\mathbf{S} - \mathbf{I})\mathbf{F}^{-1}\boldsymbol{\theta} + \lambda \mathbf{I} \right] \mathbf{v}, \end{aligned} \tag{4.17}$$

where $\Re$ is the real component operator. The derivative of the update error w.r.t $\mathbf{v}$ is given by

$$\frac{\partial \epsilon_T^2}{\partial \mathbf{v}} = -2\Re\left[i\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\mathbf{X}\right] + 2\Re\left[\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\boldsymbol{\theta} + \lambda\mathbf{I}\right]\mathbf{v}. \tag{4.18}$$

Setting the derivative to zero and solving for $\mathbf{v}$ gives a linear equation

$$\mathbf{v} = \Re\left[\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\boldsymbol{\theta} + \lambda\mathbf{I}\right]^{-1}\Re\left[i\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\mathbf{X}\right], \tag{4.19}$$

We can solve $\mathbf{v}$ naively in $\mathcal{O}(N^3)$ time. However we choose a much quicker, albeit less precise method for calculating the required matrix inverse. The term $\left[\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\boldsymbol{\theta} + \lambda\mathbf{I}\right]$ exhibits strong diagonality. If we approximate this as a diagonal matrix, the inverse becomes trivial. The diagonal elements of $\left[\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\boldsymbol{\theta} + \lambda\mathbf{I}\right]$ are given by $-(N/L+\lambda)$, thus $\mathbf{v}$ can be approximated as

$$\mathbf{v} = -\left(\frac{L}{N+\lambda L}\right)\Re\left[i\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\mathbf{X}\right], \tag{4.20}$$

leading to the final time constraint update given by

$$\begin{aligned} f(\mathbf{X}) &= \mathbf{X} + \gamma i\boldsymbol{\theta}\,\mathrm{Im}\left[\boldsymbol{\theta}^H\mathbf{F}(\mathbf{S}-\mathbf{I})\mathbf{F}^{-1}\mathbf{X}\right] \\ &= \mathbf{X} + \gamma i\boldsymbol{\theta}\,\mathrm{Im}\left[\boldsymbol{\theta}^H\left[f_{GS}(\mathbf{X}) - \mathbf{X}\right]\right], \end{aligned} \tag{4.21}$$

where Im is the imaginary component operator, $f_{GS}(\mathbf{X})$ is the time-domain constraint function for the Gerchberg-Saxton iteration and

$$\gamma = \left(\frac{L}{N+\lambda L}\right). \tag{4.22}$$

The term $i\boldsymbol{\theta}\,\mathrm{Im}\left[\boldsymbol{\theta}^H[f_{GS}(\mathbf{X}) - \mathbf{X}]\right]$ can be interpreted as component of the Gerchberg-Saxton update vector that is tangential to the magnitude circle. The tangential component is then scaled by γ to give the proposed update vector. An illustration of this is shown in Fig. 4.3. After the proposed time constraint update is applied, the standard magnitude constraint update (4.13) is applied to complete

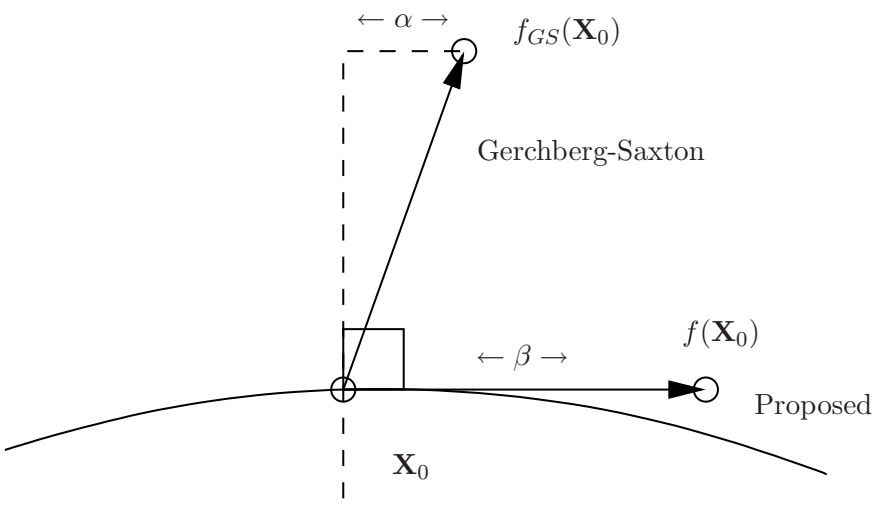

Figure 4.3: Relationship between the Gerchberg-Saxton time-domain constraint update and the proposed update. The Gerchberg-Saxton update has a tangential component, of $\alpha = i\boldsymbol{\theta}\text{Im}\left[\boldsymbol{\theta}^H[f_{GS}(\mathbf{X}) - \mathbf{X}]\right]$. The proposed update moves directly along the magnitude circle tangent with vector $\beta = \gamma\alpha$.

a full iteration. In practice, we found that the algorithm became unstable when γ was large, i.e. when $L \gg N$. In our experiments, we found that clamping γ to a maximum value of $\gamma = 1.9$ kept the iterations stable (i.e. monotonically decreasing error for each iteration).

To test the proposed algorithm, we show reconstruction performance for two audio signals; a segment of voiced speech and F16 engine noise. To measure reconstruction success, we use a signal-to-error ratio (SER). The SER measure is analogous to the more common SNR, though we now consider the energy of the reconstruction error, rather than noise. SER is defined as

$$SER(dB) = 10\log\left(\frac{\mathbf{X}^H\mathbf{X}}{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}\right), \tag{4.23}$$

where $\mathbf{X}^H$ is the Hermitian (or conjugate) transpose of $\mathbf{X}$ and $\boldsymbol{\epsilon}$ is the total error vector. Reconstructed signals were 32 ms segments of 8 kHz sampled audio, giving a 256 sample vector, to which a Hamming window was applied. We used two-fold FFT over-sampling, giving a 512 point spectral-domain vector. Magnitude spectra were

then extracted from this signal. Both the GS and proposed algorithms were then run to regenerate a phase spectrum to match the given magnitude spectra. We should point out that we are interested in synthesizing a phase spectrum that is merely consistent with the given magnitude spectrum – not producing the particular phase spectrum that corresponds to the original time-domain sequence. To bootstrap each algorithm, a random phase vector $\boldsymbol{\theta}_0$ was used provide an initial guess $\mathbf{X}_0 = \mathbf{m} \circ \exp(j\boldsymbol{\theta}_0)$, where $-\pi \leq \boldsymbol{\theta}_0 \leq \pi$. Because of the variability in the bootstrap stage, results presented here are averaged over 100 reconstructions.

Figs. 4.4 and 4.5 show the reconstruction comparisons for the speech and F16 engine noise respectively. In Figs. 4.4a), 4.5a) we show the original time-domain sequences and in Figs. 4.4b), 4.5b) we show the corresponding magnitude spectrum. Lastly, in Figs. 4.4c) and 4.5c) we show reconstruction SER as a function of iterations. Initially, both the GS and proposed algorithms drive error down very quickly. However after a dozen iterations, the slow asymptotic convergence of the Gerchberg-Saxton algorithm becomes apparent. The proposed algorithm converges approximately twice as fast – having half the error (3-4 dB improvement) after each iteration. This trend was consistent across both stimulus types. In terms of complexity, both the proposed and Gerchberg-Saxton algorithm are of $\mathcal{O}(L \log L)$ complexity per iteration. The proposed algorithm is slightly more expensive for a single iteration, requiring additional vector scalings and addition ($\mathcal{O}(L)$ operations). However since it drives down error at twice the rate of the GS algorithm, its overall convergence rate is faster.

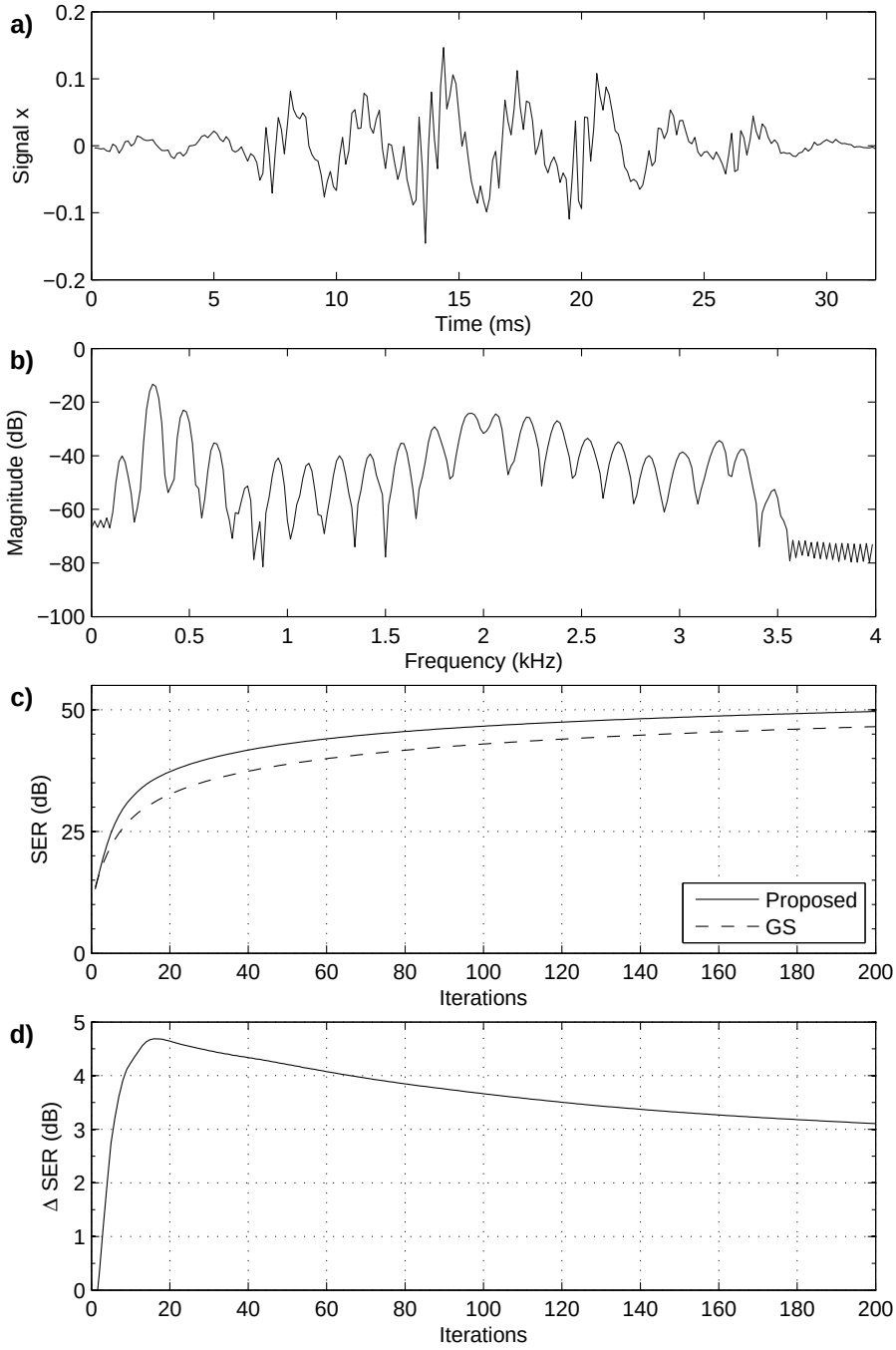

Figure 4.4: Vowel phase retrieval using proposed and the Gerchberg-Saxton methods. Subplots: a) original time-domain sequence, b) the desired magnitude spectrum (in log-domain), c) reconstruction error of the GS and proposed algorithm and d) SER improvement for the proposed algorithm.

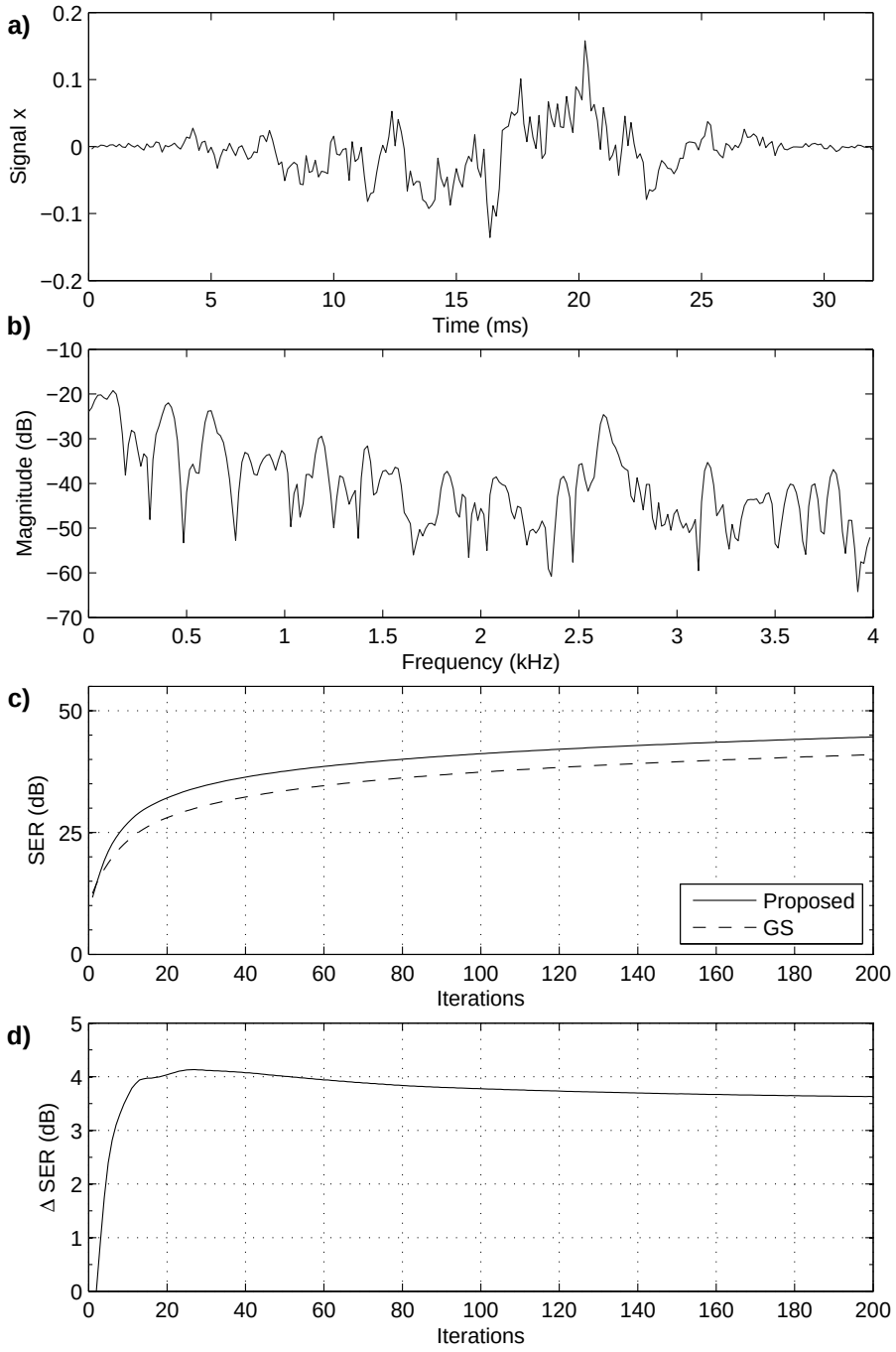

Figure 4.5: F16 engine noise phase retrieval using the proposed and the Gerchberg-Saxton methods. Subplots: a) original time-domain sequence, b) the desired magnitude spectrum (in log-domain), c) reconstruction error of the GS and proposed algorithm and d) SER improvement for the proposed algorithm.

4.5 Conclusion

In this chapter, we have provided a brief overview of some mathematical relationships that exist between the magnitude and phase spectra of a signal. In particular, we investigated the specific problems of regenerating one spectra from the other. It is of interest to note that there is a simple, closed-form linear method for regenerating the magnitude spectra from phase, but not for regenerating phase spectra from magnitude. For the phase regeneration problem, an iterative approach such as the Gerchberg-Saxton algorithm is required. However, this tends to result in a very slow phase retrieval process. The cause of this behaviour was identified and a modified Gerchberg-Saxton algorithm for phase retrieval was proposed. The proposed algorithm was tested on two audio signals, and was found to drive reconstruction error down at roughly twice the rate of the standard (Gerchberg-Saxton) algorithm.

Chapter 5

Investigation of the instantaneous frequency spectrum

5.1 Introduction

The short-time phase spectrum is a function of time as well as frequency. While there can be many ways to derive meaningful representations from the phase spectrum, two possible ways that come to mind are those obtained by taking its derivative. Differentiation along the frequency axis yields group delay and differentiation along the time axis gives instantaneous frequency (IF)[97]. Both group delay and instantaneous frequency are much more meaningful than the unprocessed phase, and both can be tied to physically relevant phenomena [16]. However, in this chapter, we choose to restrict our focus to include only the IF spectrum branch of phase processing.

The concept of IF is quite old, originally being developed as a way to characterize the phase of non-stationary signals. In its classic definition by Gabor [51], the IF of a given signal $x(t)$ is obtained as follows: the signal $x(t)$ is first converted into its analytic signal $x_a(t) = x(t)+j\mathcal{H}[x(t)] = A(t)e^{j\varphi(t)}$ (where $\mathcal{H}$ is the Hilbert transform

operator, $A(t) \geq 0$ the instantaneous amplitude and $\varphi(t)$ the instantaneous phase), and then the IF is computed as the time derivative of the instantaneous phase (i.e., the IF is given as $d\varphi(t)/dt$). Subsequently, the IF has been computed in the literature from zero-crossings and level-crossings [22, 59, 60, 74, 75, 120], AM-FM demodulation models [94, 119], Teager energy products [106], short-time Fourier magnitude spectrum [83, 99, 109, 131] and short-time Fourier phase spectrum [48, 49, 93]. The IF spectrum itself has been extensively studied in the literature for a number of speech processing applications; e.g., formant extraction [49, 107], pitch extraction [2, 23, 91, 92], speech coding [46, 108], speech recognition [34, 52, 76, 101, 135] and speaker recognition [63, 132].

A speech signal must be decomposed into a number of bandpass filtered signals before attempting to derive its IF spectrum [18]. This decomposition can be done either by bandpass filtering the speech signal in time-domain, or via a short-time Fourier transform (STFT). In this chapter, we choose to use the STFT for decomposition and restrict our focus on the IF spectrum derived via differentiation of the STFT phase spectrum (for time-domain filtering based IF spectral representations, see [63, 101, 107, 108, 135]).

Two types of STFT analysis procedures have been reported in the literature: wide-band analysis and narrow-band analysis [115]. For wide-band analysis, the duration of the analysis window is typically taken to be 2 to 4 ms. Since it is much smaller than the typical pitch period expected in normal speech, wide-band analysis is generally used in a pitch-synchronous manner, otherwise the placement of analysis window with respect to pitch epochs will introduce significant frame-to-frame variability. The IF spectrum resulting from the wide-band STFT analysis is known to provide information similar to that from the corresponding magnitude spectrum [49, 115]; i.e., both of them exhibit information about the vocal tract system, but not about the excitation source. For narrow-band analysis, the duration of the analysis window is taken to be 20 to 40 ms. Since the duration is much longer than the typical pitch period of speech, narrow-band analysis is generally used for

pitch-asynchronous analysis. It has been shown in the literature [23, 46] that the IF spectrum obtained from narrow-band analysis contains information about the excitation source, but not about the vocal tract system. As a result, the narrow-band IF spectrum has been used in the past only for pitch extraction [2, 23]. The narrow-band magnitude spectrum, on the contrary, contains information about the excitation source as well as the vocal tract system and has been used for both pitch and formant extraction [115], as well as for associated ASR tasks [69].

Wide-band pitch-synchronous STFT analysis needs *a priori* information about the location of pitch epochs. Being a difficult task, wide-band pitch-synchronous STFT analysis is rarely used in practice. Instead, pitch-asynchronous narrow-band STFT analysis is a preferred choice for most of the speech processing applications [69, 115]. In this chapter, we investigate whether the lack of vocal tract information is an inherent property of the narrow-band STFT IF spectrum. For this purpose, we explore the narrow-band STFT IF spectrum and its related quantities in greater detail. In our analysis, we have found that the STFT IF spectrum is heavily influenced by the type of analysis window function used, being far more sensitive than the corresponding magnitude spectrum. We feel that the importance of this windowing has been largely overlooked in previous IF literature. For example the use of different windows can induce a far wider range of IF behaviours than has been previously reported. In particular, we show that lack of formant structure in the narrow-band IF spectra is an artefact of the analysis window, rather than an inherent property of the IF spectrum itself. Here, an IF derived quantity, the IF deviation, can be shown to provide both formant and pitch information, much in the same way as the narrow-band magnitude spectrum. This implies that there is more speech information in the IF spectrum than has been previously reported.

In this chapter, our objective is two-fold. Firstly we seek to characterize the general properties of the STFT IF spectrum. In particular, we investigate the role of the analysis window in the computation of the STFT IF spectrum. Secondly we use this information to derive from the narrow-band STFT IF spectrum a representation

that is as rich as the corresponding magnitude spectrum; i.e., it should provide information about the source excitation as well as vocal tract system. We have developed a new representation based on the IF deviation (referred here as the IF deviation spectrum) that gives clear visual indication of underlying pitch and formant structure, and is more similar to the narrow-band magnitude spectrum than the following two existing STFT IF spectral representations [49]: the IF density spectrum and the IF spectrum. (Note that these two representations have been used in the past to generate the IF density spectrogram [49] and the pyknogram [107] as the corresponding spectro-temporal representations.)

The rest of this chapter is organized as follows. In Section 5.2, we describe the general basis for short-time Fourier transform instantaneous frequency. In Sections 5.3 and 5.4, we examine the fundamental properties of the STFT IF spectrum, investigating the properties relevant to IF densities and IF deviations respectively. Using this analysis, we propose a new IF deviation based spectral representation and show its relevance to speech analysis in Section 5.5. Lastly in Section 5.6 we present some concluding remarks.

5.2 Instantaneous frequency from the Short Time Fourier transform

The running STFT of the speech signal $x(t)$ at time t and analysis frequency ω is given by [49]

$$X(\omega, t) = \int_{-\infty}^{\infty} x(t+\tau) w(\tau) e^{-j\omega\tau} d\tau, \tag{5.1}$$

where $w(t)$ is a finite-duration window function that is symmetric. having an impulse response $w(t)e^{j\omega t}$ with the signal $x(t)$ as input. The IF $\nu(\omega, t)$ is computed as the

time-derivative of the STFT phase $\angle X(\omega, t)$ as follows:

$$\begin{aligned} \nu(\omega, t) &= \frac{\partial}{\partial t} \angle X(\omega, t) && (5.2) \\ &= \frac{\partial}{\partial t} \mathrm{Im}\left[log X(\omega, t) \right]. && (5.3) \end{aligned}$$

The plot of $\nu(\omega, t)$ as a function of the analysis frequency ω at a given time t results in the IF spectrum at time t.

By expanding the differential in (5.3), the IF can be expressed via direct differentiation of the signal $x(t)$ as follows:

$$\begin{aligned} \nu(\omega, t) &= \mathrm{Im}\left[\frac{1}{X(\omega, t)} \frac{\partial}{\partial t} X(\omega, t) \right], && (5.4) \\ &= \mathrm{Im}\left[\frac{C(\omega, t)}{X(\omega, t)} \right], && (5.5) \end{aligned}$$

where $C(\omega, t)$ is given as the running STFT of $dx(t)/dt$

$$C(\omega, t) = \int_{-\infty}^{\infty} x'(t + \tau) w(\tau) e^{-j\omega\tau} d\tau. \tag{5.6}$$

Friedman [49] has shown that the STFT IF can also be given via differentiation of the window $w(\tau)$ rather than the input signal. For this, (5.1) is altered via a change of variable as

$$X(\omega, t) = e^{j\omega t} \int_{-\infty}^{\infty} x(\kappa) w(\kappa - t) e^{-j\omega\kappa} d\kappa, \tag{5.7}$$

with derivative $\partial X/\partial t$ now given by

$$\begin{aligned} \frac{\partial}{\partial t} X(\omega, t) &= j\omega X(\omega, t) \\ &\quad + e^{j\omega t} \int_{-\infty}^{\infty} x(\kappa) \left[\frac{\partial}{\partial t} w(\kappa - t) \right] e^{-j\omega\kappa}. \end{aligned} \tag{5.8}$$

Substitution of $\partial X/\partial t$ from this equation into (5.4) yields the IF as

$$\nu(\omega, t) = \omega - \mathrm{Im}\left[\frac{\int_{-\infty}^{\infty} x(t + \tau) w'(\tau) e^{-j\omega\tau}}{X(\omega, t)} \right], \tag{5.9}$$

or alternatively,

$$\nu(\omega, t) = \omega - \mathrm{Im}\left[\frac{D(\omega, t)}{X(\omega, t)}\right], \tag{5.10}$$

where $D(\omega, t)$ is the running STFT of $x(t)$ using the differentiated window $w'(t) = dw(t)/dt$

$$D(\omega, t) = \int_{-\infty}^{\infty} x(t+\tau) w'(\tau) e^{-j\omega\tau} d\tau. \tag{5.11}$$

While the IF expressions given by (5.5) and (5.10) are useful for analytic research, they are not very helpful for implementation on a digital computer. Instead, the discrete-time version of the original IF definition given by (5.2) is used for this purpose. That is, the IF $\nu(\omega, n)$ of discrete-time (or, sampled) signals is obtained by computing the STFT phase values at discrete-time instants (or, samples) n and $n-D$, taking their difference and dividing the resulting phase difference by the time difference D; i.e.,

$$\nu(\omega, n) = \frac{1}{D}\{ARG\left[X(\omega, n)\right] - ARG\left[X(\omega, n-D)\right]\}. \tag{5.12}$$

Here, ARG denotes the operator used in the digital implementation to compute the phase. It produces a principal phase value between the limits of $\pm\pi$. Thus, the individual phases $ARG\left[X(\omega, n)\right]$ and $ARG\left[X(\omega, n-D)\right]$ in (5.12) are wrapped between the limits of $\pm\pi$. Note that the phase difference $ARG\left[X(\omega, n)\right] - ARG\left[X(\omega, n-D)\right]$ will be wrapped between the limits of $\pm 2\pi$. To produce the phase difference wrapped between the same limits (i.e., $\pm\pi$), we use Kay's method [73] to compute the IF as follows

$$\nu(\omega, n) = \frac{1}{D} ARG\left[X(\omega, n) X^*(\omega, n-D)\right], \tag{5.13}$$

This form of derivation has the added benefit of requiring only a single ARG function. However, this does not eliminate the phase wrapping problem. To reduce the

likelihood of wrapping, ω can be extracted from the ARG term

$$\nu(\omega, n) = \omega + \frac{1}{D} ARG\left[X(\omega, n)X^*(\omega, n-D)e^{-j\omega D}\right]. \tag{5.14}$$

This pushes the inner term of the ARG function toward zero – away from potential phase wrapping. To reduce the likelihood of wrapping further, we may reduce the size of time difference D in (5.14) to $D = 1$; i.e.,

$$\nu(\omega, n) = \omega + ARG\left[X(\omega, n)X^*(\omega, n-1)e^{-j\omega}\right]. \tag{5.15}$$

While (5.15) lessens the likelihood of phase wrapping, it does not solve another fundamental problem associated with the computation of the STFT phase spectrum – its instability at low spectral magnitudes. When the spectral magnitude becomes small, spectral phase is often erratic. When the spectral magnitude is zero, the phase itself is undefined. Typically this problem is resolved with some form of smoothing and/or combining with spectral magnitude.

5.3 Existing IF-based spectral representations

A commonly observed property of the short-time IF spectrum is its tendency to cluster around the dominant frequencies of a signal [22, 49, 59]. Two IF based spectro-temporal representations have been reported in the literature that take advantage of this property. These are the pyknogram [107] and the IF density spectrogram [49]. The pyknogram representation is a plot of the IF $\nu(\omega, n)$ as a function of time n for different analysis frequencies ω uniformly spaced in the interval 0 to 2π. As opposed to the spectrogram representation, which is a 3-dimensional (gray-scale) plot of $\log|X(\omega, n)|$ as a function of ω and n, the pyknogram representation is a 2-dimensional (black and white) plot of $\nu(\omega, n)$ versus n for each ω. This type of representation can be used as a means to identify dominant frequencies. But, being a binary valued (black and white) representation, the

pyknogram cannot convey relative strengths of these dominant frequencies. To derive a gray-scale representation like the spectrogram, Friedman [49] computed the IF density $p(\nu, n)$ at time n by counting the number of occurrences of IF values in the range ν and $\nu+d\nu$. The IF density spectrum is a plot of $p(\nu, n)$ as a function of ν at a given time n, while the IF density spectrogram is a 3-dimensional (gray-scale) plot of $p(\nu, n)$ as a function of ν and n.

In this section, we explore IF clustering behaviour present in the these two spectral representations (namely, the IF spectrum $\nu(\omega, n)$ and the IF density spectrum) in some detail and show that it is intrinsically linked to the type of window function used in the analysis. In particular, the sidelobe decay rate and window length appear to be the key parameters. We also show that the IF density spectrum captures only the formant frequency locations in the wide-band STFT analysis mode, and only the pitch-harmonic frequency locations in the narrow-band STFT analysis mode.

5.3.1 Role of the sidelobe decay rate of the analysis window function

In his original paper, Friedman [49] proposed the use of a Hann analysis window function in the STFT analysis for deriving the IF density spectrum. Subsequent studies reported in the literature have used either the Hann window function [23, 92] or the Blackman window function [2, 91] for the STFT IF analysis. However, no reason is provided (to the best of our knowledge) as to why the these window functions are preferred for STFT IF analysis over the Hamming window function which is perhaps the most commonly used window function for speech processing [69, 115]. Here we try to explain the reason for this as follows: the Hann and Blackman window functions are notable for their large sidelobe decay rate: 18 dB per octave [64]. This is in contrast to the 6 dB per octave for the Hamming window. To illustrate the importance of this quantity (i.e., the sidelobe decay rate), we consider the following three window functions: Hann, Hamming and Chebyshev (50 dB equiripple) window functions. The Chebyshev (50 dB equiripple) window

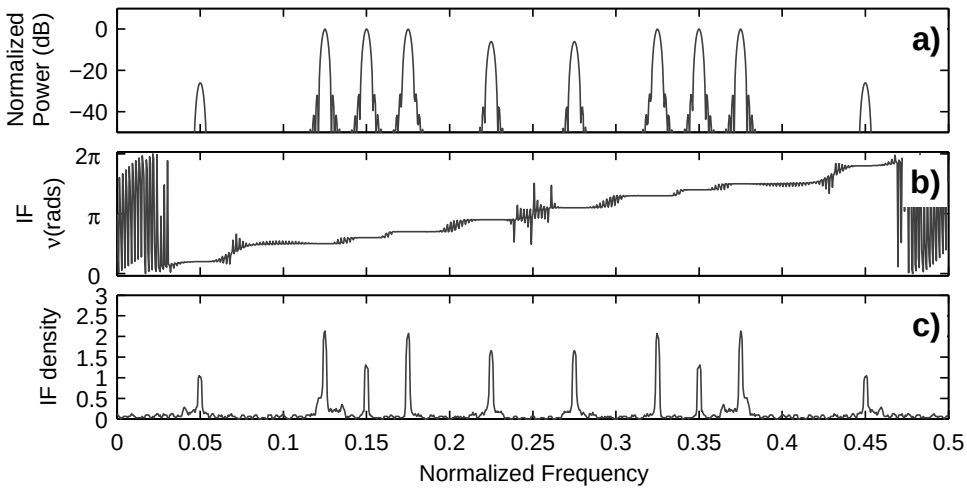

Figure 5.1: IF spectrum analysis using a high decay rate window – Hann window, 18 dB sidelobe/octave decay rate. Analysis signal is a 32 ms multi-sinusoid signal sampled at 8 kHz (5.16). Subplots: a) magnitude spectrum, b) IF spectrum and c) IF density spectrum. Frequency axis in Fig. c) refers to the normalized instantaneous frequency.

function has sidelobes that are equally-strong in its magnitude spectrum (i.e., its sidelobe decay rate is zero). We use the following signal $x(n)$ having 5 sinusoidal components for STFT analysis:

$$x(n) = \sum_{k=1}^{5} A_k \cos(\omega_k n + \phi_k), \tag{5.16}$$

where the amplitudes $\{A_k\}$ are $\{0.2, 1.0, 1.0, 1.0, 0.5\}$, the normalized frequencies $\{\omega_k/2\pi\}$ are $\{0.1, 0.25, 0.3, 0.35, 0.45\}$, and the initial phases $\{\phi_k\}$ are randomly chosen between the limits $-\pi \leq \phi_k \leq \pi$. In Figs. 5.1, 5.2 and 5.3, we show a) the magnitude spectrum, b) the IF spectrum and c) the IF density spectrum for the Hann, Hamming and Chebyshev (50 dB equiripple) window functions, respectively. For single frame analysis, the index n is dropped from our notation. In these circumstances is assumed that $n = 0$ unless stated otherwise.

Here we can see why the Hann window is preferable. For the Hann window, the IF forms a staircase structure, with each stair corresponding to a individual sinusoid

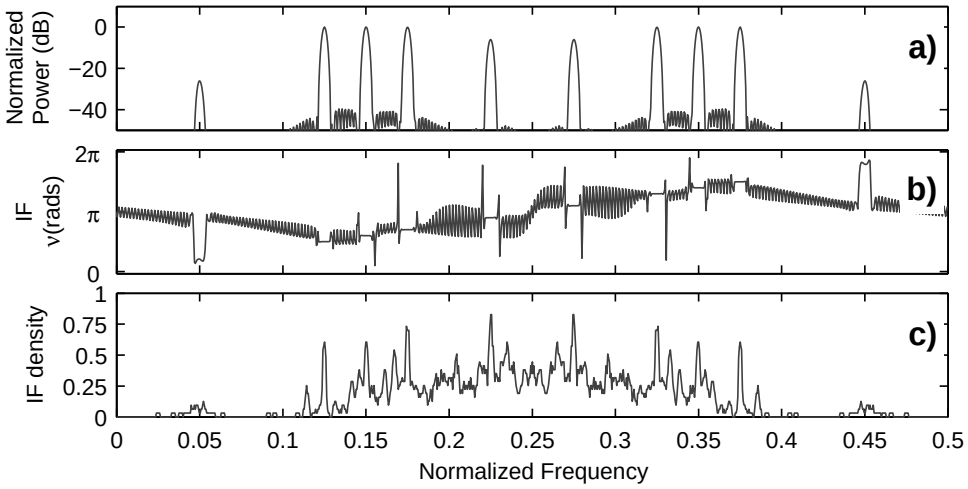

Figure 5.2: IF spectrum analysis using a moderate decay rate window – Hamming window, 6 dB sidelobe/octave decay rate. Analysis signal is a 32 ms multi-sinusoid signal sampled at 8 kHz (5.16). Subplots: a) magnitude spectrum, b) IF spectrum and c) IF density spectrum. Frequency axis in Fig. c) refers to the normalized instantaneous frequency.

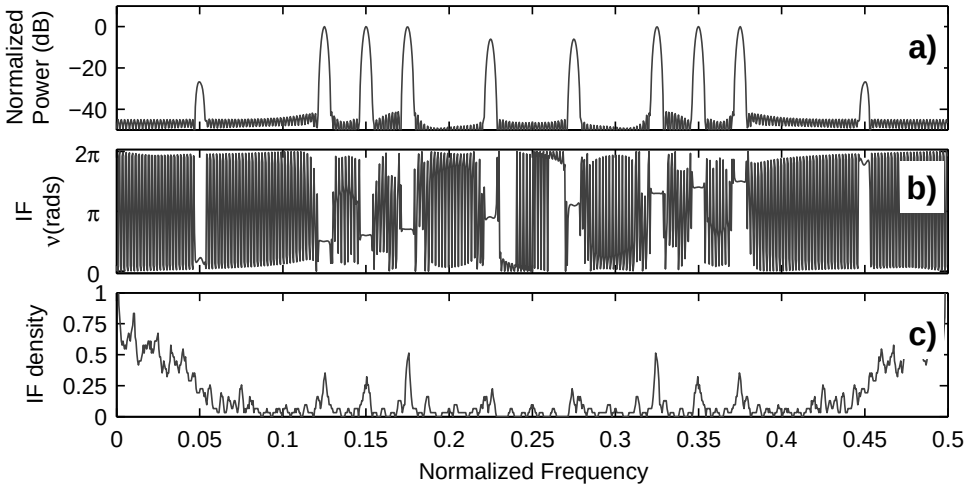

Figure 5.3: IF spectrum analysis using a zero decay rate window – Chebyshev 50 dB window, 0 dB sidelobe/octave decay rate. Analysis signal is a 32 ms multi-sinusoid signal sampled at 8 kHz (5.16). Subplots: a) magnitude spectrum, b) IF spectrum and c) IF density spectrum. Frequency axis in Fig. c) refers to the normalized instantaneous frequency.

component. As a result, the peaks in the IF density spectrum occur at the sinusoid frequencies. However, we can see the IF values becoming more chaotic between the IF stairs. This is caused by the interaction of spectral leakage. At frequencies where a particular sinusoidal component is dominant (i.e., it contributes the majority of the spectral energy), the IF is well behaved. When no single sinusoid is dominant, the resultant IF appears to be chaotic in nature. The use of a high sidelobe decay rate window function reduces the areas of high leakage interaction, allowing the wider IF staircases to appear. The opposite is true for the Hamming window. With its lower sidelobe decay rate, the spectral leakage between neighbouring sinusoidal components interact strongly across wider portions of the frequency axis. As a result, the IF spectrum and the IF density spectrum resulting from the Hamming window are dominated by the poorly behaved, unclustered IF. A more extreme case is the equiripple (zero decay rate) Chebyshev 50 dB window function. Here, spectral leakage strongly mixes across the entire frequency range, leading to very little IF spectrum clustering.

Even with the Hann (or similar) window function, the IF density spectrum displays some undesirable characteristics. In Fig. 5.1c), the largest peak in the IF density spectrum is produced by the lowest energy sinusoidal component. This is because the IF density is largely determined by the width of the IF stairs, rather than spectral energy. The sinusoidal components at 0.1 and 0.9 frequencies occupy a large portion of the frequency axis, thus produce wider IF stairs. The opposite is true for the sinusoidal components at frequencies 0.3 and 0.7. Being confined to a much smaller frequency region leads to a correspondingly smaller IF density peak. Thus, the IF density spectrum can be used to compute the dominant frequencies, but it is unable to provide any information about their relative strengths (or, magnitudes).

5.3.2 Narrow-band and wide-band properties

The behaviour of the IF spectrum and the IF density spectrum changes dramatically when switching between narrow-band and wide-band STFT analysis modes. To

illustrate this behaviour, we use a voiced speech signal and carry out the narrow-band and wide-band STFT analysis with 32 ms and 4 ms segments, respectively. In both cases, we use a Hann analysis window. Figs. 5.4 and 5.5 show a) the magnitude spectrum, b) the IF spectrum and c) the IF density spectrum obtained from the wide-band and narrow-band STFT analysis, respectively. It is clear from Fig 5.4c) and 5.5c) that we lose the formant frequency information when moving from the wide-band to the narrow-band IF density analysis. The reason for this can be seen in the narrow-band IF staircasing structure of Fig. 5.5b). Here the staircase is quite uniform – a result of individual harmonics being equally spaced across the frequency axis. Since each harmonic occupies roughly the same amount of the frequency axis, the IF density produces uniform peaks at each harmonic location. This behaviour is not seen in the wide-band analysis since there is insufficient frequency resolution to resolve individual harmonics. Instead, the wider formant structure is captured. Unsurprisingly sidelobe characteristics play a lesser role in wide-band analysis. In this case, mainlobe interactions become a more dominant effect.

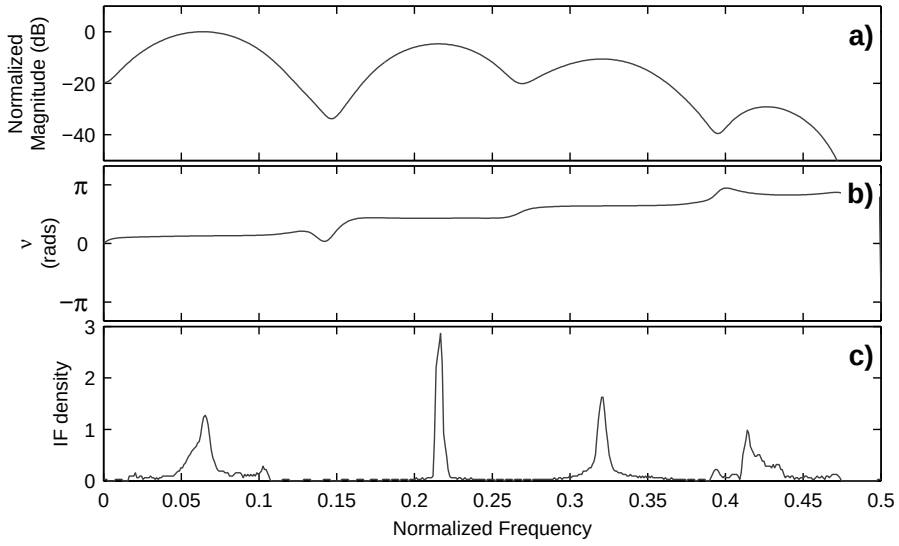

Figure 5.4: Wide-band IF spectrum analysis for a 4 ms segment of voiced speech, digitized at 8 kHz. A Hann analysis window is used to derive a) magnitude spectrum, b) IF spectrum and c) IF density spectrum. Frequency axis in Fig. c) refers to the normalized instantaneous frequency.

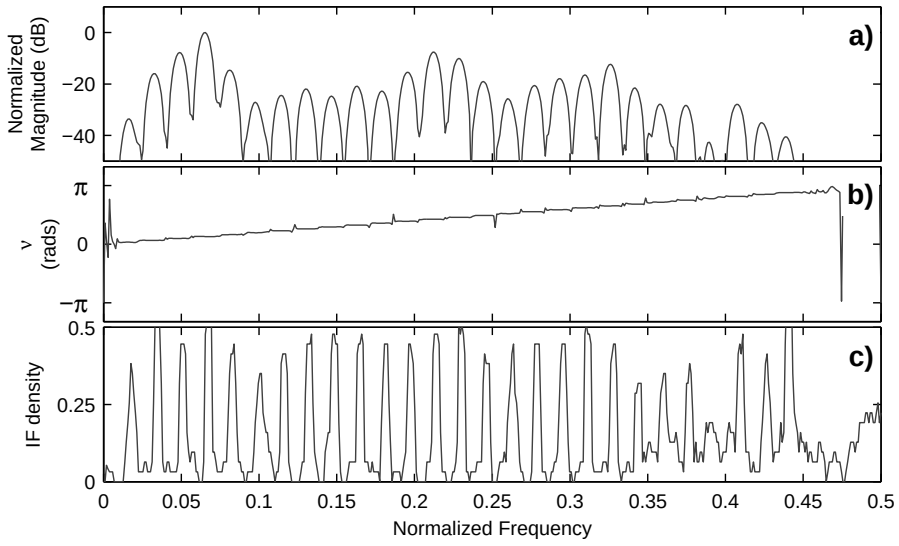

Figure 5.5: Narrow-band IF spectrum analysis for a 32 ms segment of voiced speech, digitized at 8 kHz. A Hann analysis window is used to derive a) magnitude spectrum, b) IF spectrum and c) IF density spectrum. Frequency axis in Fig. c) refers to the normalized instantaneous frequency.

5.4 Proposed spectral representation using STFT IF deviation

As mentioned earlier, our aim in this chapter is to derive a more meaningful representation from the STFT IF spectrum. Like the narrow-band magnitude spectrum, this representation should display the pitch as well as the formant structure. The narrow-band IF density spectrum described in the preceding section does not exhibit formant structure. It captures the frequencies of pitch harmonics, but not their relative strengths. In order to overcome these deficiencies, we introduce in this section a new representation based on IF deviation. We define the IF deviation $\psi(\omega, t)$ as

$$\psi(\omega, t) = \nu(\omega, t) - \omega. \tag{5.17}$$

For discrete-time signals, it becomes

$$\psi(\omega, n) = \nu(\omega, n) - \omega. \tag{5.18}$$

Combining (5.15) and (5.18), we have

$$\psi(\omega, n) = ARG\left[X(\omega, n)X^*(\omega, n-1)e^{-j\omega}\right]. \tag{5.19}$$

We have seen earlier from Figs. 5.1(b) and 5.5(b) that the IF spectrum behaves like a stair-case, with horizontal stairs occurring at the pitch harmonic peaks. The IF values track the frequencies of pitch harmonic peaks. We will show in this section that the accuracy of the tracking (the IF deviation magnitude) is inversely proportional to the spectral magnitude; i.e., the higher the magnitude of a harmonic peak, the better the IF value tracks its corresponding harmonic frequency. This relationship can be described mathematically as follows:

$$|\psi(\omega, n)|^{-1} \propto |X(\omega, n)|, \tag{5.20}$$

or in log-domain

$$-\log|\psi(\omega,n)| \propto \log|X(\omega,n)|. \tag{5.21}$$

We use the function $|\psi(\omega,n)|^{-1}$, the inverse absolute IF deviation (IAIFD) to define a new spectral representation. Note that the proposed IAIFD representation differs from the IF density representation in the following manner: instead of grouping the IF values into a density function (and losing information of individual IF values), we directly plot transformed IF values at each time-frequency location in the proposed representation. In the rest of this section, we mathematically characterize some of the properties of the IF deviation quantity.

5.4.1 Using STFT IF deviation to capture locations of spectral components

In this section, we use some simple synthetic signals to show how the STFT IF deviation may be used to identify the locations of the dominant spectral components. Given a simple sinusoidal signal $x(t) = A_0 e^{j(\omega_0 t+\phi_0)}$, where A_0 and ϕ_0 are the magnitude and initial phase of the sinusoid, it is easy to show that the IF is constant and equal to ω_0. This leads to a ramped IF deviation that crosses the axis through zero at ω_0. Thus, IAIFD $|\psi(\omega,t)|^{-1}$ will present a peak at the sinusoid frequency ω_0. To illustrate this, we take a complex 800 Hz sinusoid sampled at 8 kHz. We show the magnitude, IF, IF deviation, absolute IF deviation, log-IAIFD and IF density spectra in Fig. 5.6a) through f), respectively. Here we use a 5-point mean filter to smooth the absolute IF deviation $|\psi(\omega,t)|$ (shown in Fig. 5.6d)) prior to calculating the log-IAIFD in Fig. 5.6e). This smoothing removes zero values in the absolute IF deviation, making the IAIFD and log-IAIFD spectra better behaved. In Fig. 5.6f), the IF density spectrum clearly displays a peak at the sinusoid location. The log-IAIFD spectrum is not as neat, but still displays a peak at the correct frequency. While IF analysis is trivial for a single sinusoidal signal, this is certainly not true for more complex signals. As an example, let us take a signal $x(t)$ having two sinusoidal

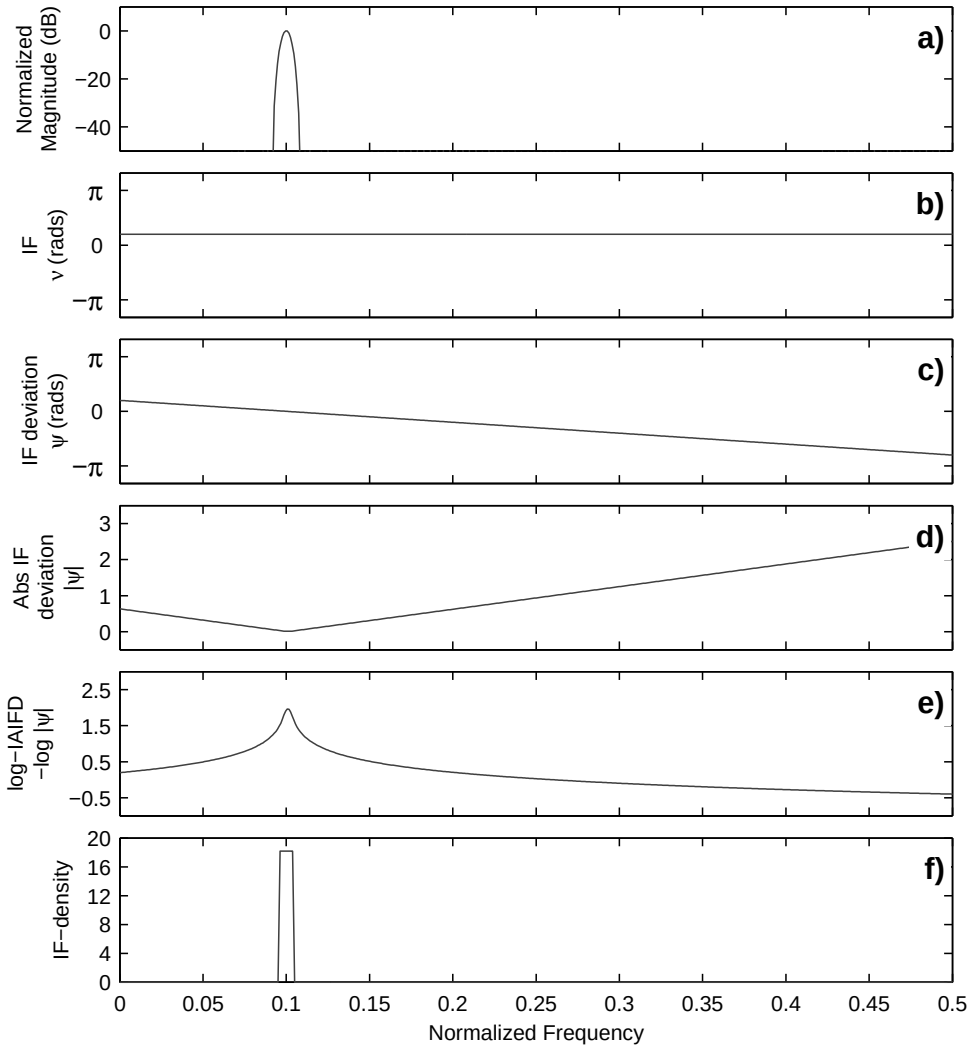

Figure 5.6: Narrow-band IF spectrum analysis for a 32 ms 800 Hz complex sinusoid sampled at 8 kHz. a) Magnitude spectrum, b) IF spectrum, c) IF deviation spectrum, d) absolute IF deviation, e) log-IAIFD spectrum and f) IF density spectrum. IF density spectrum was generated via a Hann window, remaining spectra used a 50 dB Chebyshev window. The frequency axis of figure f) refers to normalized instantaneous frequency.

components,

$$\begin{aligned} x(t) &= x_0(t) + x_1(t) \\ &= A_0 e^{j(\omega_0 t + \phi_0)} + A_1 e^{j(\omega_1 t + \phi_1)}, \end{aligned} \tag{5.22}$$

where A_k and ϕ_k are the magnitude and initial phase of the k'th sinusoid, respectively. After STFT analysis, this becomes

$$\begin{aligned} X(\omega,t) &= X_0(\omega,t) + X_1(\omega,t) \\ &= A_0 W(\omega-\omega_0)e^{j(\omega_0 t+\phi_0)} + A_1 W(\omega-\omega_1)e^{j(\omega_1 t+\phi_1)}, \end{aligned} \tag{5.23}$$

where $W(\omega)$ is the Fourier transform of the analysis window function. For notational convenience, we drop the ω and t subscripts from $X_0(\omega,t)$ and $X_1(\omega,t)$ and dependence on these subscripts is implicitly assumed hereafter. For the purposes of this analysis, we consider the first sinusoid to be dominant, such that $|X_0| > |X_1|$ for the time-frequency location of interest. Combining (5.3) and (5.23) gives the instantaneous frequency as

$$\begin{aligned} \nu(\omega,t) &= \frac{\partial}{\partial t}\mathrm{Im}\log X(\omega,t) \\ &= \omega_0 + \frac{\partial}{\partial t}\mathrm{Im}\log\left[1+\frac{B_1(\omega)}{B_0(\omega)}e^{j(\omega_1-\omega_0)t}\right]. \end{aligned} \tag{5.24}$$

where $B_k(\omega)$ is given as $A_k W(\omega-\omega_k)e^{j\phi_k}$. It should be further noted that $|B_k(\omega)| = |X_k(\omega,t)|$. In (5.24) the IF of the two-sinusoid signal can be interpreted as a constant term (belonging to the dominant sinusoid) and an oscillating term with period $T = 2\pi/(\omega_1-\omega_0)$. The average value of the IF, $\langle\nu(\omega,t)\rangle$ with respect to time is given by

$$\begin{aligned} \langle\nu(\omega,t)\rangle_t &= \frac{1}{T}\int_0^T \nu(\omega,t)dt \\ &= \omega_0 + \left[\frac{1}{T}\mathrm{Im}\log\left[1+\frac{B_1(\omega)}{B_0(\omega)}e^{j(\omega_1-\omega_0)t}\right]\right]_0^T \\ &= \omega_0. \end{aligned} \tag{5.25}$$

In other words, the average instantaneous frequency is the dominant spectral component – in this case the sinusoid $X_0(\omega,t)$ with centre frequency $\omega = \omega_0$. For the case of when $|X_1| > |X_0|$, phase wrapping occurs between the integration limits.

Accounting for the phase wrap, the average IF is now given by

$$\begin{aligned}\langle \nu(\omega,t)\rangle_t &= \omega_0 + \frac{2\pi}{T} \\ &= \omega_1.\end{aligned} \tag{5.26}$$

This shows the IF oscillating now around the dominant ω_1 frequency. In order to characterize the IF behaviour further, we require (5.24) to be simplified. Again assuming $|X_0| > |X_1|$, (5.24) can be simplified to

$$\nu(\omega,t) = \omega_0 + \mathrm{Re}\left[\frac{(\omega_1-\omega_0)}{1+\frac{B_0(\omega)}{B_1(\omega)}e^{j(\omega_0-\omega_1)t}}\right]. \tag{5.27}$$

Fig. 5.7 shows the instantaneous frequency of the two-sinusoid signal (5.22) computed using (5.27). Here, the two sinusoidal components, $x_0(t)$ and $x_1(t)$, of the signal (5.22) are located at normalized frequencies $\omega_0/2\pi = 0.7$ and $\omega_1/2\pi = 0.2$, respectively, with $A_0 = 1.6$, $A_1 = 1$ and $\phi_0 = 5.9$, $\phi_1 = 0$. A 16-samples long Hamming window was used for STFT analysis. We plot the IF versus time for three normalized analysis frequencies $(\omega/2\pi)$ equal to $\{0.75, 0.46, 0.38\}$. The magnitude ratio $|X_1(\omega,t)/X_0(\omega,t)|$ at these frequencies was $0.1, 0.6$ and 1, respectively. Since the sinusoidal component of frequency $\omega_0 = 0.7$ is dominant at the first two of these analysis frequencies, the IF is seen to oscillate around 0.7 for these analysis frequencies. Furthermore, the magnitude of the oscillations increases as the ratio $|X_1/X_0|$ approaches to 1. At the analysis frequency $\omega/2\pi = 0.38$ (where the two sinusoidal components are of equal magnitude), the IF becomes equal to the average of two sinusoidal component frequencies $(= (0.7+0.2)/2 = 0.45)$, superimposed with periodic delta pulses.

We have explored these properties further in Fig. 5.8. Here we show the IF properties of the same two-sinusoid signal (5.22) across the entire frequency axis. In Fig. 5.8a), we have shown the separate magnitude spectra for component sinusoids $X_0(\omega,t)$ and $X_1(\omega,t)$. The lighter line represents the $\omega_0/2\pi = 0.7$ sinusoid, while the darker line represents the $\omega_1/2\pi = 0.2$ sinusoid. In Fig. 5.8b), we have shown

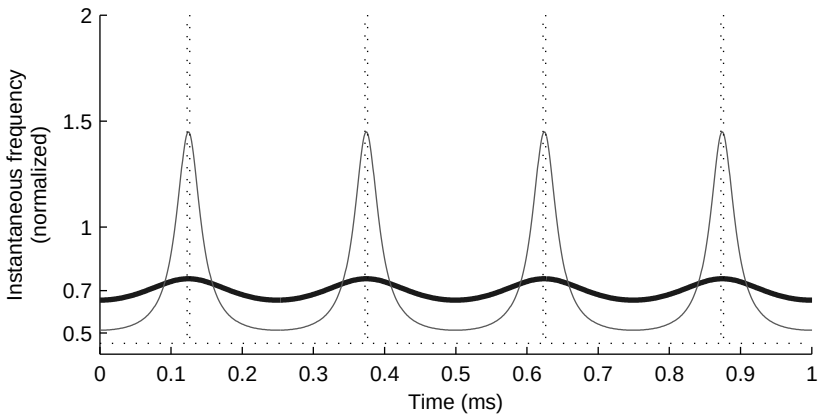

Figure 5.7: Instantaneous frequency analysis of a two-sinusoid signal, $X(\omega) = X_0(\omega, t) + X_1(\omega, t)$, sampled at 8 kHz. Sinusoid components $X_0(\omega, t)$ and $X_1(\omega, t)$ are located at normalized frequencies 0.7 and 0.2 respectively. Dark line represents magnitude ratio $|X_1/X_0| = 0.1$, light gray line represents magnitude ratio $|X_1/X_0| = 0.6$, dotted line represents magnitude ratio $|X_1/X_0| = 1$. When $|X_0| > |X_1|$ IF measurements oscillate around the dominant spectral component located at $\omega_0 = 0.7$. As the magnitude ratio approaches unity, the magnitude of oscillations becomes greater.

the IF properties of the combined signal. The dark line in Fig. 5.8b) represents the average instantaneous frequency (with respect to time), while the shaded region represents the range of the IF oscillations. When the $X_0(\omega, t)$ sinusoid is dominant, the IF oscillates around its frequency of 0.7. The opposite occurs when the $X_1(\omega, t)$ sinusoid is dominant. Furthermore, we can see that the magnitude of the oscillations is tied to sinusoid dominance. At the mainlobe regions of either sinusoid, there is very little oscillation in the IF. This leads to a correspondingly small IF deviation. In the frequency regions where there is strong interaction ($|X_1/X_0| \approx 1$), the IF (and IF deviation) can vary wildly.

The IF oscillation behaviour described above is intimately tied to the IF deviation quantity. Again assuming $|X_0| > |X_1|$, (5.27) can be altered to give the IF deviation

$$\psi(\omega, t) = (\omega_0 - \omega) + \mathrm{Re}\left[\frac{(\omega_1 - \omega_0)}{1 + \frac{B_0(\omega)}{B_1(\omega)} e^{j(\omega_0 - \omega_1)t}}\right]. \tag{5.28}$$

Here we have two terms; a linear difference term $(\omega_0 - \omega)$ and the oscillation term

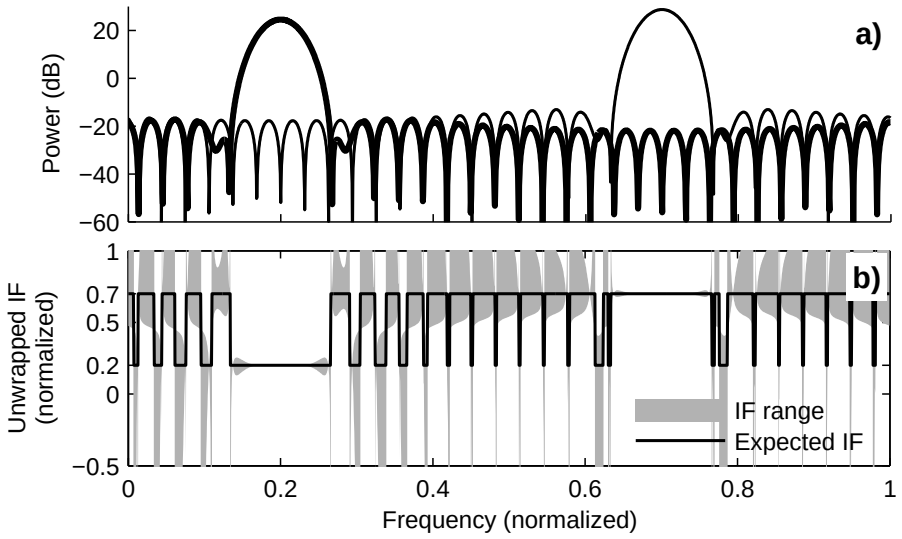

Figure 5.8: Instantaneous frequency analysis of a two-sinusoid signal, $X(\omega) = X_0(\omega, t) + X_1(\omega, t)$, sampled at 8 kHz. Sinusoid components $X_0(\omega, t)$ and $X_1(\omega, t)$ are located at normalized frequencies 0.7 and 0.2 respectively. Subplots: a) magnitude spectrum for component sinusoids: light line is $X_0(\omega, t)$ and dark line is $X_1(\omega, t)$. b) IF properties of the combined two-sinusoid signal $X(\omega)$. Dark line is the expected IF (w.r.t time), while the shaded region represents the range of values as the IF oscillates over time. The expected IF value reflects which sinusoid is dominant, while the range of IF reflects the degree of dominance. When one sinusoid is dominant, the IF is pushed toward that sinusoids centre frequency.

characterized earlier. Taken together, we now expect the IF deviation to be small when 1) a particular sinusoid is dominant, and 2) we are near the centre frequency of that sinusoid. Thus, we can use small IF deviations to indicate centre frequencies of dominant spectral components.

5.4.2 Capturing the relative magnitudes of spectral components

In the previous sub-section, we have shown that IF deviation may be used to find the frequency location of dominant spectral components. In this sub-section, we show that the IF deviation may also be used to capture the relative magnitudes of each spectral component. It is this property that motivates us to examine the IF deviation spectrum in preference to the IF density spectrum. To show how this property can be obtained, we take a more qualitative approach to our analysis.

Using (5.10) and (5.17), we can calculate IF deviation as

$$\psi(\omega, t) = -\text{Im}\left[\frac{D(\omega, t)}{X(\omega, t)}\right]. \tag{5.29}$$

We note that the denominator of (5.29) is simply the short-time Fourier transform of $x(t); X(\omega, t)$. Given a standard window function (with the maximum of its magnitude spectrum at $\omega = 0$), we expect peaks to form in $|X(\omega, t)|$ at dominant sinusoid frequencies. The numerator term behaves slightly differently. From (5.11), $D(\omega, t)$ is the running STFT of $x(t)$ when using the differentiated window $w'(t)$ for short-time Fourier analysis. In spectral-domain the differentiated window becomes $j\omega W(\omega)$, a function that becomes zero at $\omega = 0$. Thus, sinusoid peaks are not present in $|D(\omega, t)|$ and the function is composed mostly of leakage. If we assume this spectral leakage function to be constant, then from (5.29) we can reason that the inverse IF deviation is proportional to the magnitude spectrum. As a sinusoid increases relative to its neighbours (and their associated spectral leakage), its IF deviation should be pushed toward zero. We should point out that zeros within the IF deviation spectrum do not necessarily indicate the presence of a sinusoid. In practice, spurious IF deviation zeros are hard to predict and occur whenever the numerator leakage function $D(\omega, t)$ is zero, or $D(\omega, t)$ and $X(\omega, t)$ are out of phase by 0 or π radians. The latter condition is especially problematic when $X(\omega, t)$ becomes small – since the phase of small spectral components is often erratic and prone to computational round-off error. While spurious IF deviation zeros do occur in practice, it's usual for them to be randomly distributed, allowing them to be smoothed out by the 5-point mean filter.

Of more practical concern is the assumption that the leakage function $D(\omega, t)$ is constant. In general, this is not the case. But, we may achieve this goal approximately by selecting an appropriate window function for our STFT analysis. Ideally we would to like have a window such that $j\omega W(\omega)$ is spectrally flat. We have experimented with several existing window functions [64] and found the 50 dB equiripple Chebyshev window to give good results. In order to illustrate this, we

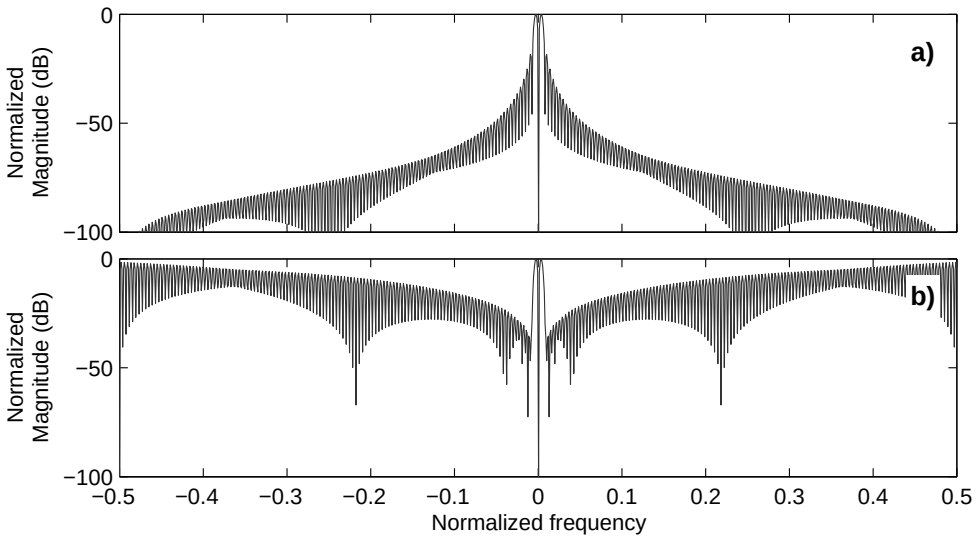

Figure 5.9: Magnitude spectrum $|j\omega W(\omega)|$ of the time-differentiated window $w'(t) = dw(t)/dt$, where $w(t)$ is a 256-pt Hann window in subplot a), and a 256-pt 50 dB Chebyshev window in subplot b).

consider the following two window functions: Hann window and 50 dB equiripple Chebyshev window. We show the magnitude spectrum $|j\omega W(\omega)|$ of the time-differentiated window $w'(t) = dw(t)/dt$ in Fig. 5.9a) for a 256-pt Hann window, and in Fig. 5.9b) for a 256-pt 50 dB Chebyshev window. It can be seen from this figure that the time-differentiated 50 dB Chebyshev window has much flatter magnitude spectrum than the time-differentiated Hann window – the (differentiated) Hann window having most of its energy concentrated about its centre frequency. In Fig. 5.10, we show the effect of the differentiated windows $j\omega W(\omega, t)$ for a given speech segment. Fig. 5.10a) shows the magnitude spectrum $|X(\omega, t)|$ of the speech segment using a Hann window. We use the Hann window to derive the leakage function $D(\omega, t)$ for the speech segment and plot its magnitude $|D(\omega, t)|$ in Fig. 5.10b). Here, the spectral leakage function retains a large degree of formant information. This in turn cancels much of the formant information present in the denominator of (5.29). When a 50 dB equiripple analysis window is used in Fig. 5.10c), the spectral leakage function is much flatter. This leads to more formant

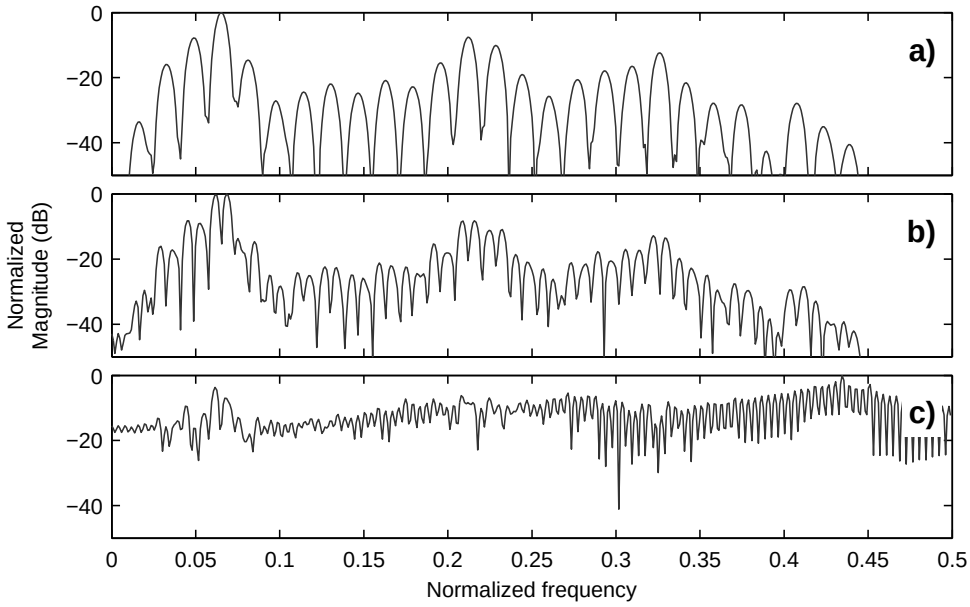

Figure 5.10: Effect of analysis window on the leakage function $D(\omega, t)$ (5.11). Subplots: a) magnitude spectrum (Hann window), b) leakage function magnitude spectrum (Hann window), c) leakage function magnitude spectrum (50 dB Chebyshev window).

information being preserved in (5.29). Appropriateness of the 50 dB equiripple Chebyshev window for the computation of STFT IF deviation is illustrated further in the next section. Interestingly, note that the choice of window function for IF deviation is the opposite (in terms of sidelobe decay rate characteristics) of the windows used for the IF density spectrum described in the previous section.

5.5 Instantaneous frequency deviation representations for speech

We demonstrate the log-IAIFD spectrum with a 32 ms speech signal, digitized at 8 kHz. The magnitude, IF, IF deviation, absolute IF deviation, log-IAIFD and IF density spectra are shown in Fig. 5.11 a) through f), respectively. Here we can see that the IF density spectrum loses formant structure, while the log-IAIFD deviation spectrum does not. Corresponding wide-band spectra are shown in Fig. 5.12. While

both the IF density and log-IAIFD spectra show formant locations, the IF density spectrum is sharper in its display.

A time-frequency IAIFD spectrogram can be developed in the same way as a magnitude spectrogram. Since speech is assumed quasi-stationary over small segments of time, we employ 32 ms analysis window and a frame-shift of 8 ms to compute the narrow-band spectrograms. In order to illustrate the spectro-temporal properties of the IF based spectral representations, we use a speech utterance of the sentence *"Wipe the grease off his dirty face."* spoken by a male speaker sampled at 8 kHz. Fig. 5.13 shows the magnitude spectrogram, IAIFD spectrogram, pyknogram and IF density spectrogram in a) through d), respectively. Here, we can see that the narrow-band pyknogram and IF density spectrograms capture fine detail harmonics, but not formant structure. The IAIFD spectrogram does not suffer this drawback, capturing both formant and harmonic details for narrow-band analysis. In Fig. 5.14, we show the corresponding wide-band spectrograms. Here we use 4 ms analysis window and a frame-shift of 1 ms. Like the wide-band magnitude spectrogram, all the three wide-band IF-based spectrograms (IAIFD, pyknogram and density) capture the formant frequency locations, but do not provide information about the pitch harmonics.

Fig. 5.15 shows the effect of analysis window types on the proposed IAIFD spectrogram. For the Hann window (Fig. 5.15c)), the IAIFD spectrogram loses formant structure becoming more similar to the IF density spectrogram. In this case, the Hann analysis window has failed to satisfy the spectral leakage criteria outlined in Section 5.4.2.

Regardless of analysis window used, a drawback of both the IAIFD and IF density spectrograms is their lack of frame-energy scaling. In the magnitude spectrograms, silence regions are shown to have very small energies, but this is not conveyed for the IF density and IAIFD spectrograms. This is because the phase spectrum of a given frame does not depend on its frame-energy. The simplest way to rectify this is to weight individual IF deviation spectra by their corresponding frame-energies.

Thus, the IAIFD representation becomes

$$10log_{10}E(n) - 20log_{10}|\psi(\omega, n)|, \tag{5.30}$$

where the frame energy $E(n)$ is given by

$$E(n) = \sum_{m=0}^{M-1} \left[x(n+m)w(m)\right]^2. \tag{5.31}$$

Fig. 5.16 shows the effect of frame-energy weighting when applied to the narrow-band IAIFD spectrograms. Corresponding wide-band frame-energy weighted spectrograms are shown in Fig. 5.17. From these figures, the frame-energy weighted IAIFD spectrogram can be seen to have a very close similarity to the magnitude spectrogram.

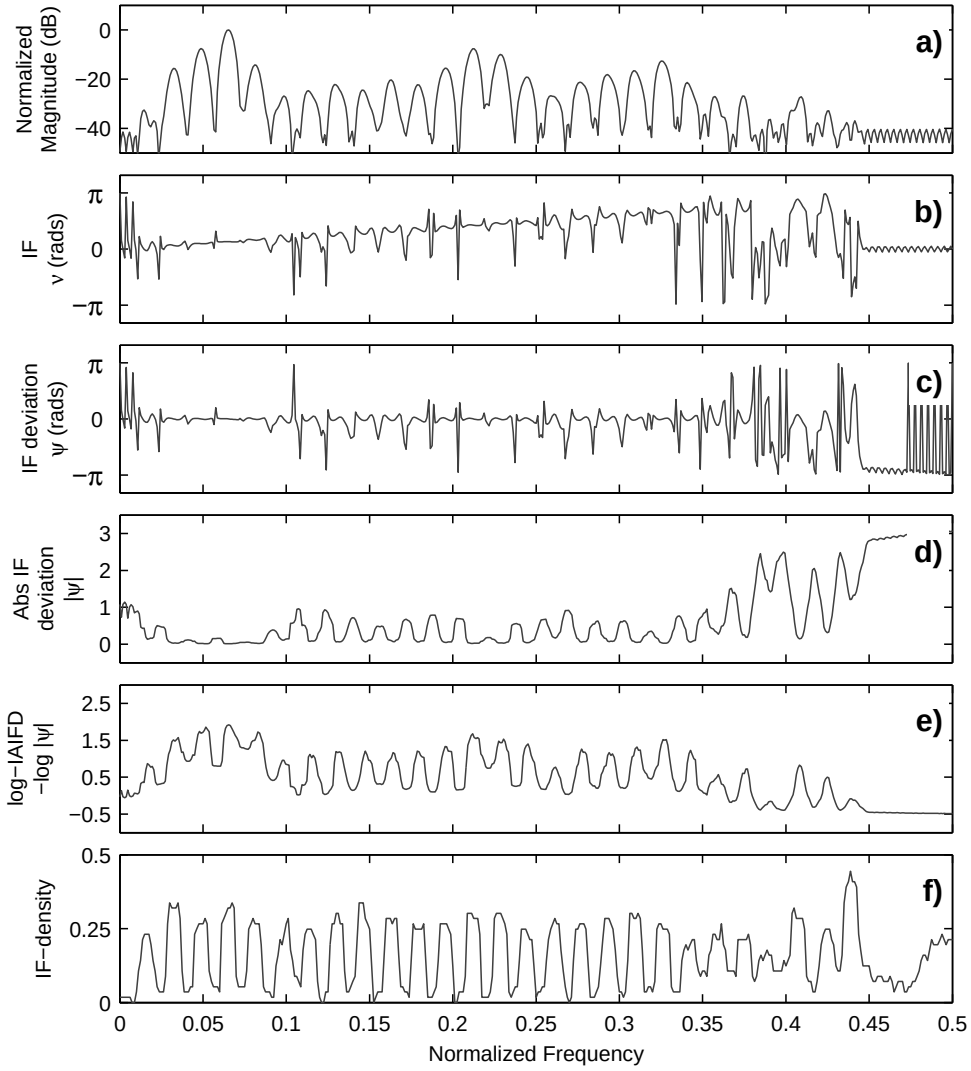

Figure 5.11: Narrow-band IF spectrum analysis for a 32 ms segment of voiced speech, digitized at 8 kHz. Subplots: a) magnitude spectrum, b) IF spectrum, c) IF deviation spectrum, d) absolute IF deviation, e) log-IAIFD spectrum and f) IF density spectrum. IF density spectrum was generated via a Hann window, remaining spectra used a 50 dB Chebyshev window. The frequency axis of figure f) refers to normalized instantaneous frequency.

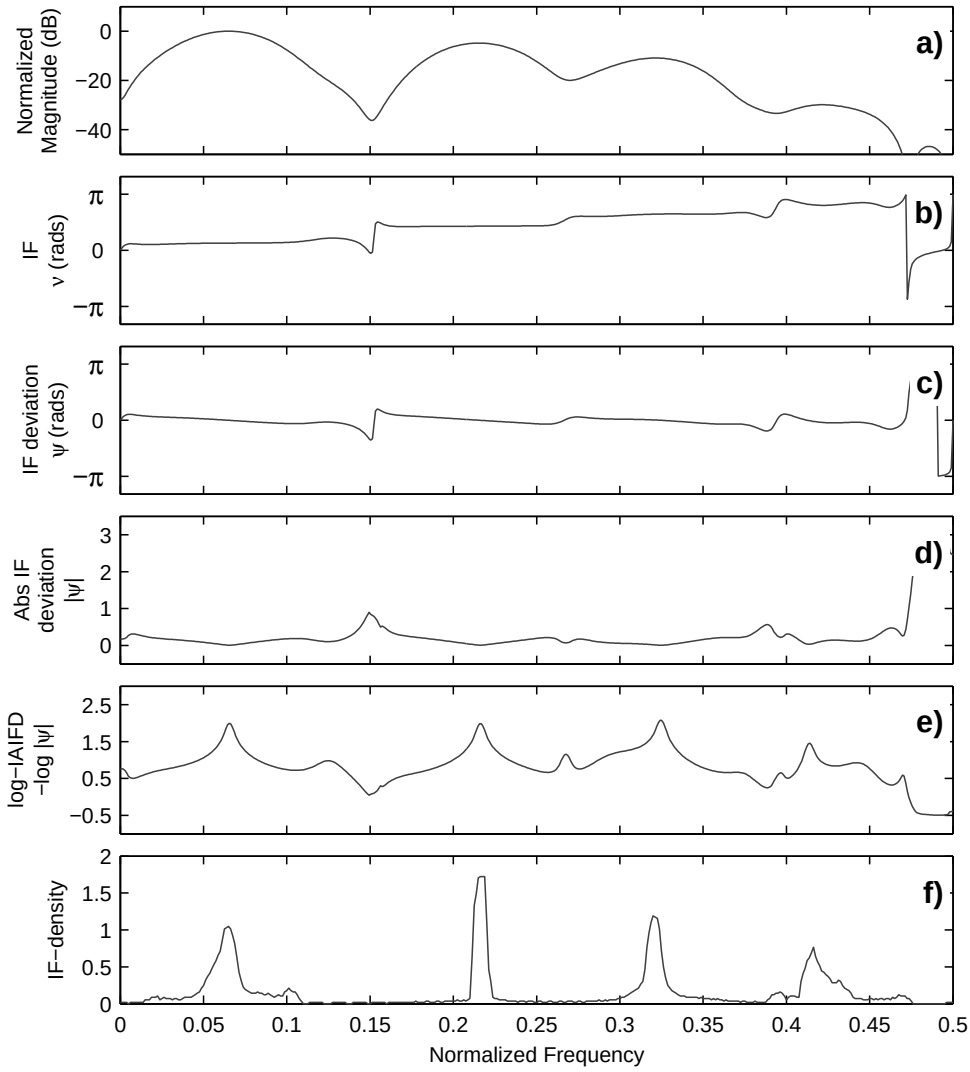

Figure 5.12: Wide-band IF spectrum analysis for a 4 ms segment of voiced speech, digitized at 8 kHz. Subplots: a) magnitude spectrum, b) IF spectrum, c) IF deviation spectrum, d) absolute IF deviation, e) log-IAIFD spectrum and f) IF density spectrum. IF density spectrum was generated via a Hann window, remaining spectra used a 50 dB Chebyshev window. The frequency axis of Fig. f) refers to normalized instantaneous frequency.

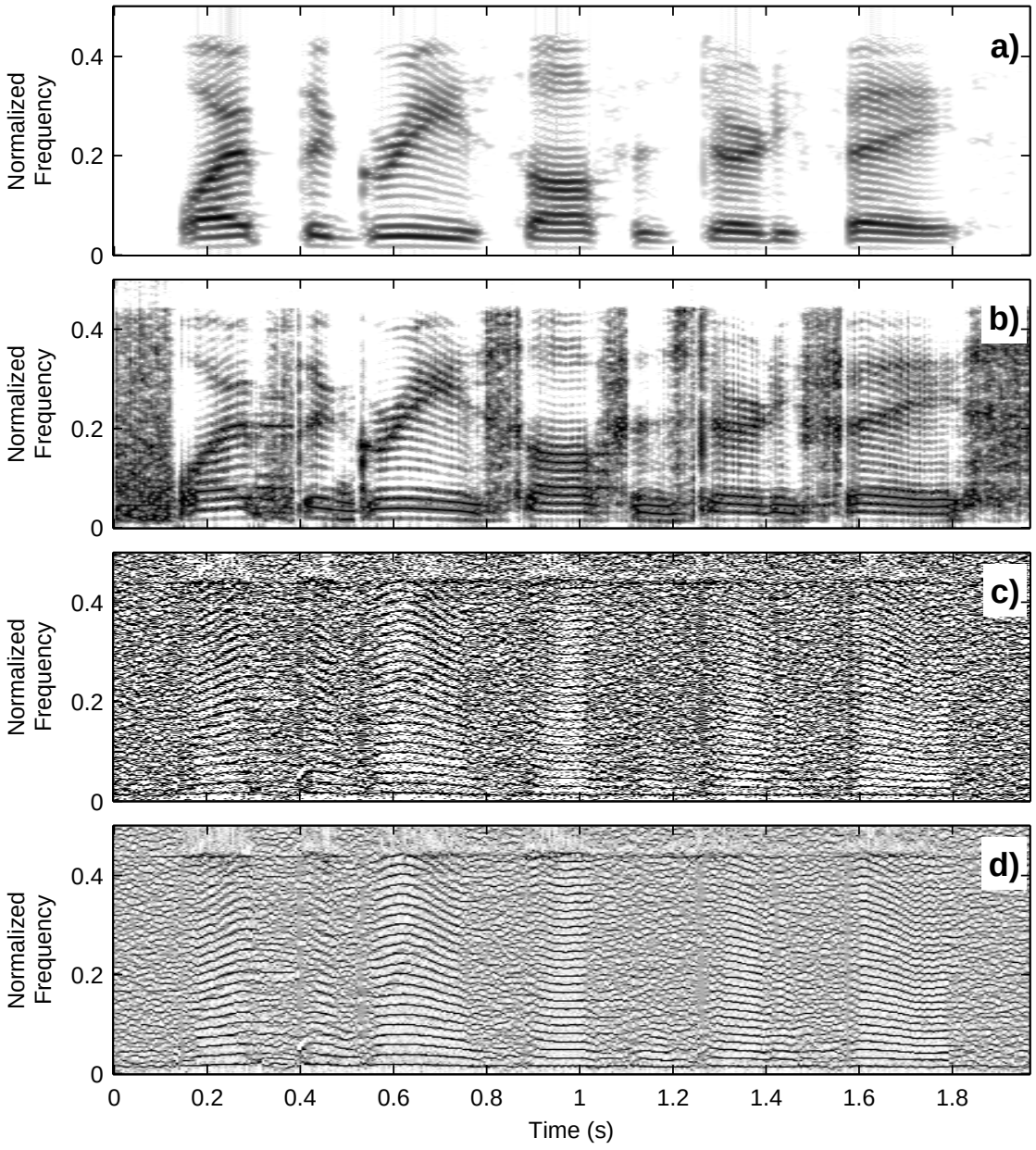

Figure 5.13: Narrow-band spectrograms for a short sentence digitized at 8 kHz. Subplots: a) magnitude spectrogram, b) IAIFD spectrogram c) IF pyknogram and d) IF density spectrogram. IF pyknogram and density spectrogram are generated via 32 ms Hann window, while remaining spectrograms use a 32 ms Chebyshev 50 dB window. Sentence is a male speaker, *'Wipe the grease off his dirty face'*. The frequency axis of Figs. c) and d) refers to normalized instantaneous frequency.

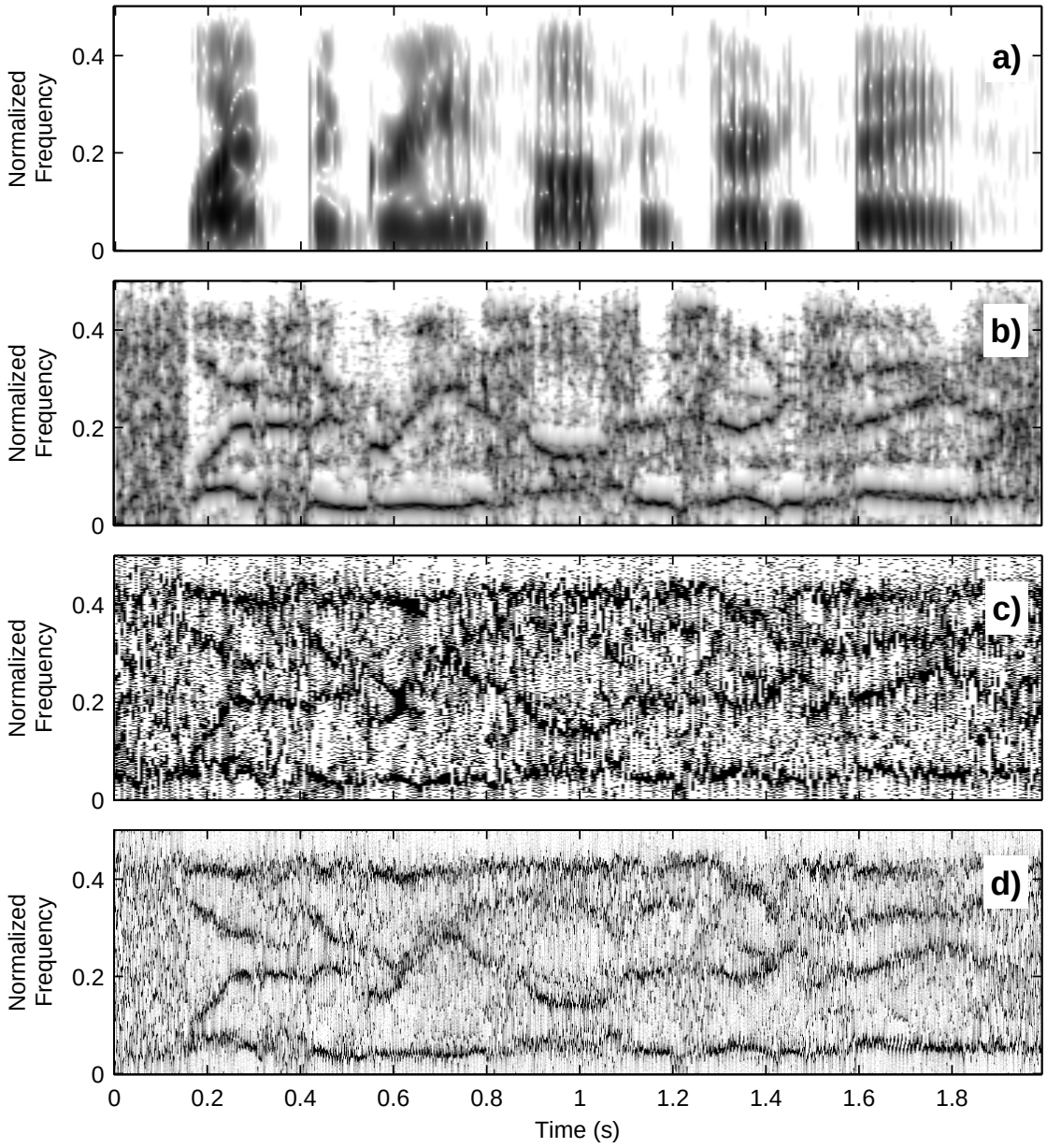

Figure 5.14: Wide-band spectrograms for a short sentence digitized at 8 kHz. Subplots: a) magnitude spectrogram, b) IAIFD spectrogram and c) IF pyknogram and d) IF density spectrogram. IF pyknogram and density spectrogram are generated via 32 ms Hann window, while remaining spectrograms use a 32 ms Chebyshev 50 dB window. Sentence is a male speaker, *'Wipe the grease off his dirty face'*. The frequency axis of Figs. c) and d) refers to normalized instantaneous frequency.

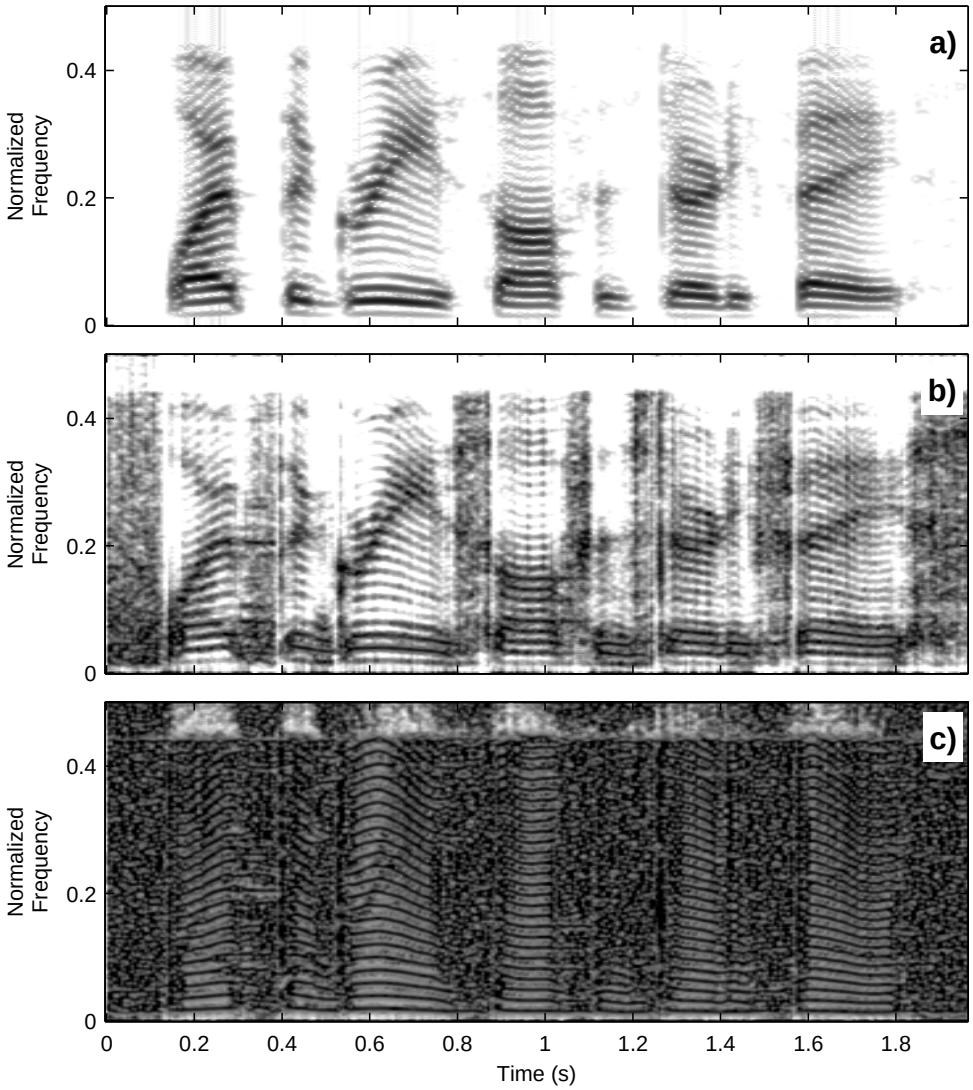

Figure 5.15: Effect of analysis window on the IAIFD narrow-band spectrogram. Subplots: a) magnitude spectrogram, 32 ms 50 dB Chebyshev window, b) IAIFD spectrogram, 32 ms 50 dB Chebyshev window and c) IF deviation spectrogram, 32 ms Hann window. Sentence is 8 kHz digitized speech, male speaker, *'Wipe the grease off his dirty face'*.

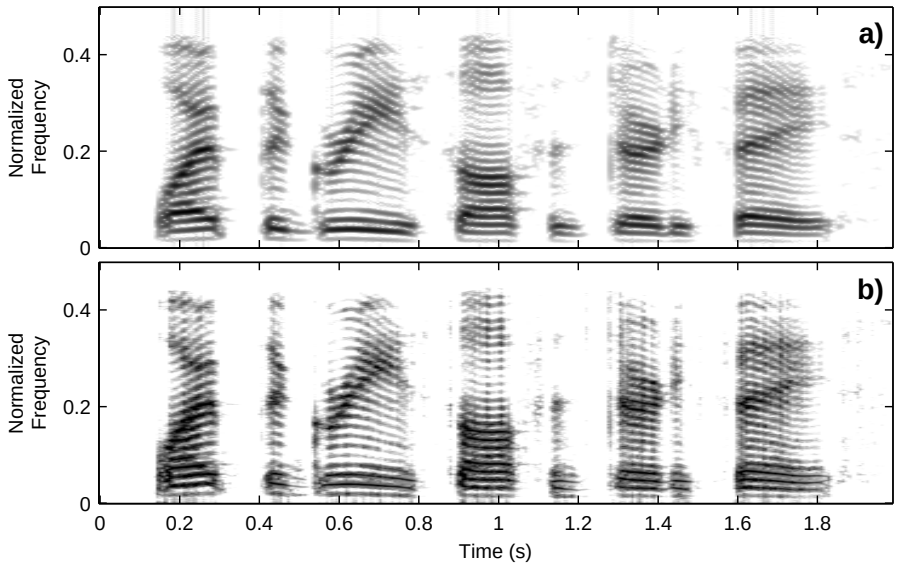

Figure 5.16: Narrow-band instantaneous frequency analysis with frame-energy weighting, for a 32 ms segment of speech sampled at 8 kHz. Subplots: a) magnitude spectrogram, b) IAIFD spectrogram with frame-energy weighting. 50 dB Chebyshev analysis window used for all plots.

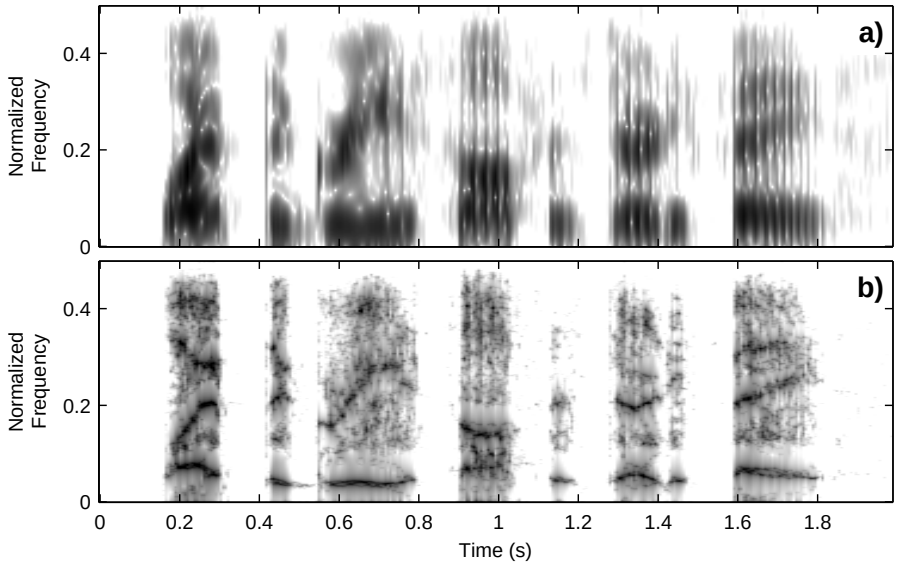

Figure 5.17: Wide-band instantaneous frequency analysis with frame-energy weighting, for a 4 ms segment of speech sampled at 8 kHz. Subplots: a) magnitude spectrogram, b) IAIFD spectrogram with frame-energy weighting. 50 dB Chebyshev analysis window used for all plots.

5.6 Conclusion

In this chapter, we have investigated the STFT IF spectrum. By characterizing the behaviour of this spectrum, we have shown that the choice of analysis window function is of critical concern. In particular, we have shown how different analysis windows can manifest vastly different IF behaviours. This is in contrast to the magnitude spectrum, which is relatively insensitive to changes in the analysis window. For past IF based representations, analysis windows were chosen to manifest behaviour suitable for deriving the IF density spectrum. In this case, high sidelobe decay rate windows (such as Hann and Blackman) were used.

However, very different IF behaviours may be induced with other window types. In this chapter, we have developed a new spectral representation – the IF deviation spectrum. We have shown that the 50 dB equiripple Chebyshev window function is a better choice for computing the IF deviation spectrum than the high sidelobe decay rate Hann window. The narrow-band IF deviation spectrum manifests both formant and pitch information similar to the narrow-band magnitude spectrum – something not seen in previous IF based representations. Visual inspection of the IF deviation spectrum suggests its potential to many ASR applications in which the short-time magnitude spectrum is traditionally used. One immediate possibility could be to use this spectrum to derive typical speech features (such as Mel-frequency cepstral coefficients) for the ASR application. However, while the IF deviation spectrum undoubtedly contains useful formant information, a more interesting task would be to extract speech features from this spectrum that are complementary to those derived from the magnitude spectrum for the ASR task. Such a goal remains a challenging problem for the future.

Chapter 6

Investigation of the group delay spectrum

6.1 Introduction

In Chapter 5, the short-time instantaneous frequency spectrum of speech was examined. In the process of this investigation, it was found that the short-time IF deviation quantity exhibited many speech features. This chapter is similarly motivated. However, focus is shifted from the IF spectrum to the group delay (GD) spectrum. The group delay spectrum is defined as the negative frequency derivative of the phase spectrum. Given the segment of speech $x(n)$, $n = 0, 1, ..., N-1$, group delay $\tau(\omega)$ can be given as

$$\begin{aligned} \tau(\omega) &= -\frac{\partial\theta(\omega)}{\partial\omega}, \\ &= -\frac{\partial}{\partial\omega}\mathrm{Im}\log\left[X(\omega)\right], \\ &= -\mathrm{Im}\left[\frac{X'(\omega)}{X(\omega)}\right], \end{aligned} \tag{6.1}$$

where $\theta(\omega)$ is Fourier phase spectrum and $X(\omega)$ is the Fourier transform of $x(n)$ given by,

$$X(\omega) = \sum_{n=0}^{N-1} w(n).x(n)e^{-j\omega n}. \tag{6.2}$$

Here $w(n)$ is the analysis window N samples in length. The term $X'(\omega)$ is the frequency derivative of $X(\omega)$ and is given as

$$X'(\omega) = \sum_{n=0}^{N-1} -jn.w(n).x(n)e^{-j\omega n}. \tag{6.3}$$

The group delay may also be given by [140]

$$\tau(\omega) = \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{|X(\omega)|^2}, \tag{6.4}$$

where $Y(\omega)$ is defined as the Fourier Transform of the signal $n.x(n)$, and $_R$ and $_I$ are the real and imaginary component modifiers. Typically, group delay is used to characterize filter and (transmission) channel characteristics. For information carrying signals such as speech, the definition is less clear cut. In addition to this ambiguity, the group delay of speech is often observed to behave poorly, having many impulsive regions. From (6.4), we can see that zeros in $X(\omega)$ correspond to poles in the group delay. In the regions near these zeros, the group delay becomes both large and chaotic – masking more useful features. While some research has gone into choosing appropriate window functions for GD speech analysis, in practice it is difficult to predict where the group delay poles will appear. Because of this, the majority of group delay methods do not use (6.4) directly, but rather compute modified spectra.

In the speech field, the group delay spectrum appears to be less studied than the corresponding IF spectrum. Nonetheless, this chapter seeks to mathematically characterize the properties of the group delay spectrum in a manner similar to Chapter 5. In particular, we investigate the use of a group delay deviation quantity, and examine whether its inverse can be made to exhibit less volatility than the

unprocessed group delay. The rest of this chapter is organized as follows. In Section 6.2, we cover the main approaches to group delay speech processing introduced in previous literature. In Section 6.3, we introduce our proposed group delay deviation function, and show its characteristics for both narrow-band and wide-band speech analysis. Finally, in Section 6.4, we present concluding remarks.

6.2 Group delay processing for speech

6.2.1 Modified group delay spectrum

Since zeros within the magnitude spectrum $|X(\omega)|$ are a direct cause of GD intractability, Yegnanarayana and Murthy [140] replaced $|X(\omega)|$ by a cepstrally smoothed magnitude spectrum $S(\omega)$. The updated group delay function is given as follows

$$\tau_p(\omega) = \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{S(\omega)^{2\gamma}}, \tag{6.5}$$

where $\gamma \approx 0.9$ is a tuning parameter. The modified group delay function (MGDF), then has its dynamic range altered

$$MGDF(\omega) = \text{sign}\left[\tau_p(\omega)\right].|\tau_p(\omega)|^{\alpha}, \tag{6.6}$$

where $\alpha \approx 0.4$. For values of $\gamma \neq 1$, the modified GD function directly incorporates magnitude spectrum information. In fact, setting $\gamma = 0$, yields a direct multiplication of the power and group delay spectra – the so called product spectrum [103]. Since the dynamic range of the magnitude spectrum is comparatively larger than that of the group delay, the MGDF and product spectrum exhibit primarily magnitude spectrum information. Even when $\gamma = 1$, the modified group delay function implicitly leverages the power spectrum through use of the smoothed magnitude $S(\omega)$. As the level of smoothing increases, the modified group delay becomes a scaled product spectrum. Because of this, we view the MGDF as leveraging group delay information to modify / enhance the magnitude spectrum,

rather than vice-versa.

6.2.2 Chirp transform group delay

Since spectral zeros lying on, or near the unit circle are the primary cause of group delay volatility, it can be beneficial to compute group delay characteristics with a modified Fourier transform. Chirp transform group delay (CGD) analysis [19] involves two stages. First, the explicit elimination of (spectral) zeros lying outside of the unit circle. This reduces a speech signal to its minimum phase form. Secondly, the Z-transform is evaluated over a circle whose radius is greater than unity (typically radius $\rho \approx 1.1$). Unlike the MGDF, the chirp group delay function does not leverage any magnitude information. However, there is a high computational cost involved for the identification and removal of spectral zeros.

6.3 Group delay deviation

Previous studies have already characterized many GD spectrum characteristics, notably its ability to detect and/or enhance useful speech features such as formants and harmonics [37, 90, 122] as well as reconstruct speech [141]. However, to use such information, special attention needs to be paid to reducing the volatility introduced by spectral zeros. In the previous sections, we have shown two methods used to overcome this problem. This study takes a different approach. Unlike the MGDF (and related product spectrum), we attempt to derive a GD function free from the influence of the power spectrum. Also, we seek a function that can process group delay without alteration to the underlying signal, as is done with CGD. To accomplish this, we direct our attention away from the group delay spectrum, and instead focus on the related group delay deviation – how much the group delay deviates from an expected value. This follows an approach used in the previous chapter, where we derived similar information from the instantaneous frequency spectrum [126].

For a non-causal symmetric window (defined between $-T/2 \leq t \leq T/2$), high

power spectral components are generally observed to have group delays close to zero. However, these group delays are often hard to see, due to the surrounding noisy components. Taking the inverse however, allows these regions to be pushed above the surrounding noise. In most practical applications, a causal, symmetric window is used for analysis. In this case, high power regions are observed to have a group delay close to $(N-1)/2$, where N is the length of the analysis frame. We can thus define a new quantity $\eta(\omega)$ as the inverse absolute group delay deviation (IAGDD) spectrum.

$$\eta(\omega) = |\tau(\omega) - \tau_w|^{-1}\,, \tag{6.7}$$

or with log compression,

$$\log \eta(\omega) = -\log |\tau(\omega) - \tau_w|\,, \tag{6.8}$$

where τ_w is the expected group delay, which for a symmetric N point window is given by $(N-1)/2$. For practical applications, it should be noted that the while the proposed function does remove some group delay noise, it does not eliminate it altogether. In speech, formant and harmonic peak regions typically produce small GD deviations. However, low power regions are generally unpredictable and noisy. Because of this, we apply smoothing to the absolute group delay deviation spectrum, i.e.

$$\hat{\eta}(\omega) = \eta(\omega) * H(\omega), \tag{6.9}$$

where $*$ is the convolution operator and $H(\omega)$ is a smoothing filter. For digital implementation, we use a simple 5-point moving average.

6.3.1 Group delay deviation for synthetic signals

A complex sinusoid signal, windowed by an N-point symmetric window can be shown to have group delay equal to $(N-1)/2$. Therefore, we start our analysis by analysing a signal that consists of two complex sinusoids. If an N-point window $w(n)$ is applied

to this signal, its short-time Fourier transform is given by

$$\begin{aligned} X(\omega) &= X_0(\omega) + X_1(\omega) \\ &= A_0 e^{j\phi_0} W(\omega - \omega_0) + A_1 e^{j\phi_1} W(\omega - \omega_1). \end{aligned} \tag{6.10}$$

Where A_k, ϕ_k and ω_k are the magnitude, initial phase and frequency of the k'th sinusoidal component respectively. $W(\omega)$ is given as the Fourier transform of the analysis window. Solving for the group delay yields

$$\begin{aligned} \tau(\omega) &= \mathrm{Im}\frac{\partial}{\partial\omega}\log\left[X_0(\omega) + X_1(\omega)\right] \\ &= \mathrm{Im}\frac{\partial}{\partial\omega}\log\left[X_0(\omega)\right] + \mathrm{Im}\frac{\partial}{\partial\omega}\log\left[1 + V(\omega)e^{j\psi}\right], \end{aligned} \tag{6.11}$$

where,

$$V(\omega) = \left|\frac{X_1(\omega)}{X_0(\omega)}\right|, \tag{6.12}$$

and

$$\psi(\omega) = \angle\left[\frac{X_1(\omega)}{X_0(\omega)}\right]. \tag{6.13}$$

Looking at the group delay of (6.11), we can see that the first term simply evaluates to $(N-1)/2$. Thus we can think of the second term as a deviation measure – how much the actual group delay is pushed off from the expected group delay. We may define the group delay deviation of the two-sinusoid signal as follows

$$\begin{aligned} \tau(\omega) - \tau_w &= \tau(\omega) - \frac{N-1}{2} \\ &= \frac{\partial}{\partial\omega}\mathrm{Im}\log\left[1 + V(\omega)e^{j\psi(\omega)}\right]. \end{aligned} \tag{6.14}$$

When the term $V(\omega) \ll 1$, the bracketed term in (6.14) is almost constant (w.r.t ω). This means the scope for introducing group delay deviation becomes small. $V(\omega)$ itself becomes small whenever $|X_0(\omega)| \gg |X_1(\omega)|$ or, rearranging (6.11), $|X_1(\omega)| \gg |X_0(\omega)|$. Because of this, the GD deviation can be thought of as a crude measure of spectral purity – the smaller the GD deviation, the more likely the energy in $X(\omega)$ originated primarily from a single sinusoid.

This assumption also holds when analysing multi-component signals, where a signal is given by K sinusoids

$$X(\omega) = \sum_{k=0}^{K-1} X_k(\omega), \tag{6.15}$$

where $X_k(\omega)$ is the k'th sinusoidal component. It can be useful to look at regions of the spectrum where a single sinusoid is dominant. In the region where $X_0(\omega)$ is dominant, the group delay deviation is then given by

$$\begin{aligned} \tau(\omega) - \tau_w &= \frac{\partial}{\partial\omega}\text{Im}\log\left[1 + \frac{\sum_{k=1}^{K-1} X_k(\omega)}{X_0(\omega)}\right] \\ &= \frac{\partial}{\partial\omega}\text{Im}\log\left[1 + V(\omega)e^{j\psi(\omega)}\right]. \end{aligned} \tag{6.16}$$

It is of particular interest to look at the ratio that comprises $V(\omega)$

$$V(\omega) = \left|\frac{\sum_{k=1}^{K-1} X_k(\omega)}{X_0(\omega)}\right|. \tag{6.17}$$

We can see that as $V(\omega)$ becomes small (as a result of $X_0(\omega)$ becoming dominant), the group delay deviation is again pushed toward zero. This means that we can expect that at frequencies where a single sinusoid is dominant (i.e. at sinusoid centre frequencies), the group delay deviation should be small. Unlike the previous IF deviation spectrum (Chapter 5), we have thus far not found any mechanism in the GD deviation to accurately convey the relative strengths of sinusoidal components.

6.3.2 Group delay deviation for voice signals

We apply the proposed group delay representation (6.8) to a short spoken sentence. We compare the proposed GD representation against an instantaneous frequency deviation function and minimum-phase chirp group delay function. Since the product spectrum and modified group delay functions are primarily magnitude-based, they are not shown here. For narrow-band analysis, the spectrograms derived

from the magnitude spectrum, the IF deviation spectrum [126], the minimum-phase chirp group delay spectrum (CGD) [19], and the proposed inverse absolute group delay deviation (IAGDD) spectrum are shown in figure 6.1 a) through d), respectively. Corresponding wide-band analysis is shown in figure 6.2 a) through d), respectively. The magnitude, IF deviation and IAGDD spectra use a 50 dB Chebyshev window while CGD spectrum uses the Blackman window suggested by the authors [19]. We can see that both the CGD and IAGDD display speech information – though less than both the magnitude spectrum and IF derived spectrum. For wide-band analysis, both group delay functions exhibit formant structure, though for narrow-band analysis, IAGDD gains harmonic structure at the expense of formant structure. In comparison to the instantaneous frequency representation proposed earlier [126], the group delay deviation spectrum does not exhibit vocal tract information as clearly. Furthermore, the GD spectrum appears to be intrinsically more volatile than the IF spectrum, especially for wide-band analysis. Here, the IF based spectrogram clearly displays formant tracks, while the GD based spectrogram is much more distorted.

6.4 Conclusion

In this chapter, we have developed a new spectral representation derived from the short-time phase spectrum – namely the group delay deviation representation. We have also shown that several speech features are readily derived from the group delay spectrum. In particular, the harmonic peaks are easily identified. Unlike previous studies on speech-based group delay, our representation does not require a minimum phase signal, or supplementary magnitude information. However, the proposed GD based function appears to manifest less speech information than the IF representation proposed earlier – both in its narrow-band and wide-band forms.

While this chapter has focused on studying the characteristics of GD for speech signals, it remains to be seen if the proposed GD information can be used to derive features that are complementary to existing magnitude-based cepstral features.

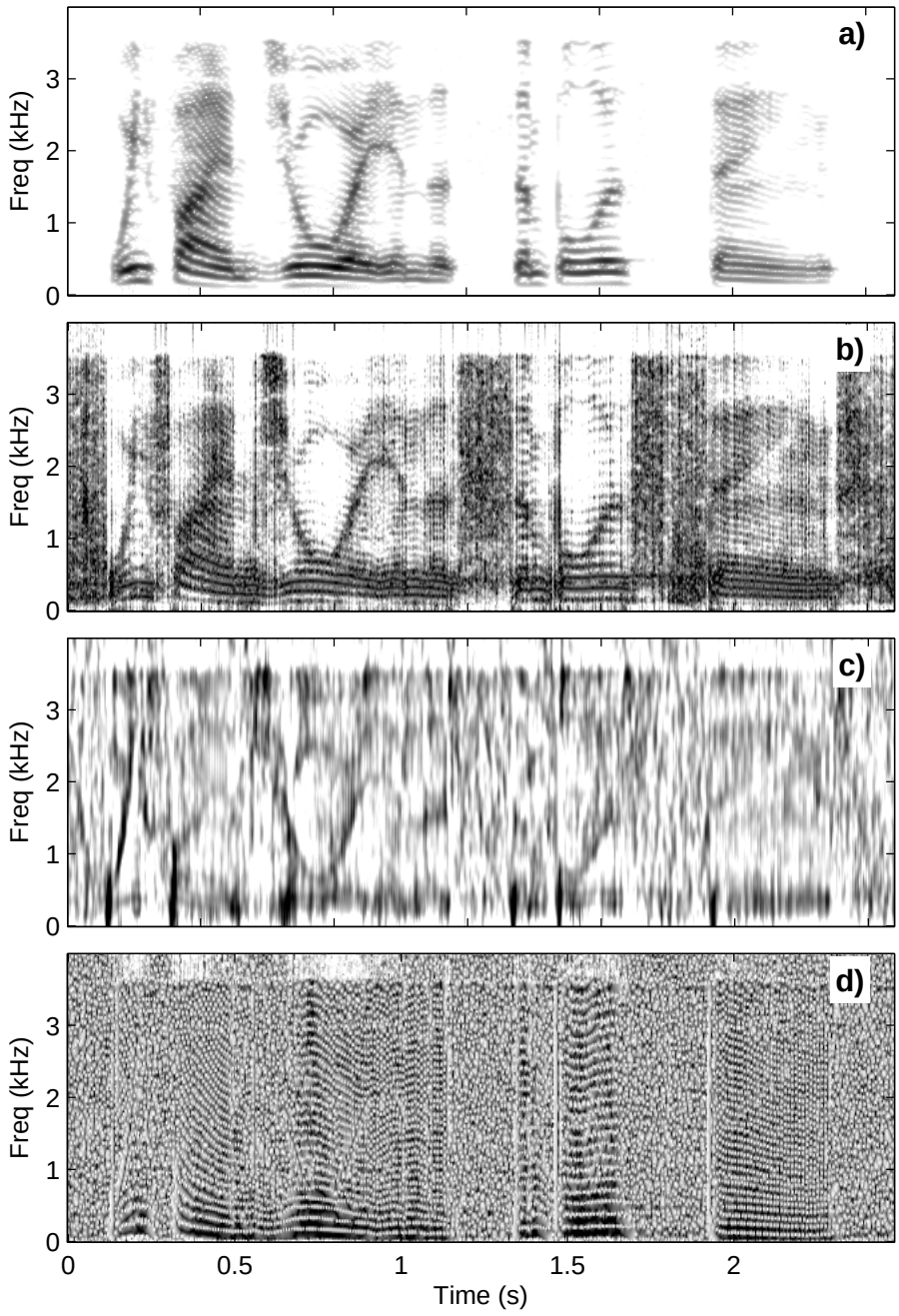

Figure 6.1: Narrow-band (32 ms frames) GD based spectrograms for a speech utterance. Subplots: (a) magnitude spectrogram, (b) instantaneous frequency deviation, (c) chirp group delay spectrogram and (d) inverse group delay deviation spectrogram. Stimulus is an 8kHz sentence: 'we find joy in the simplest things', spoken by a male speaker.

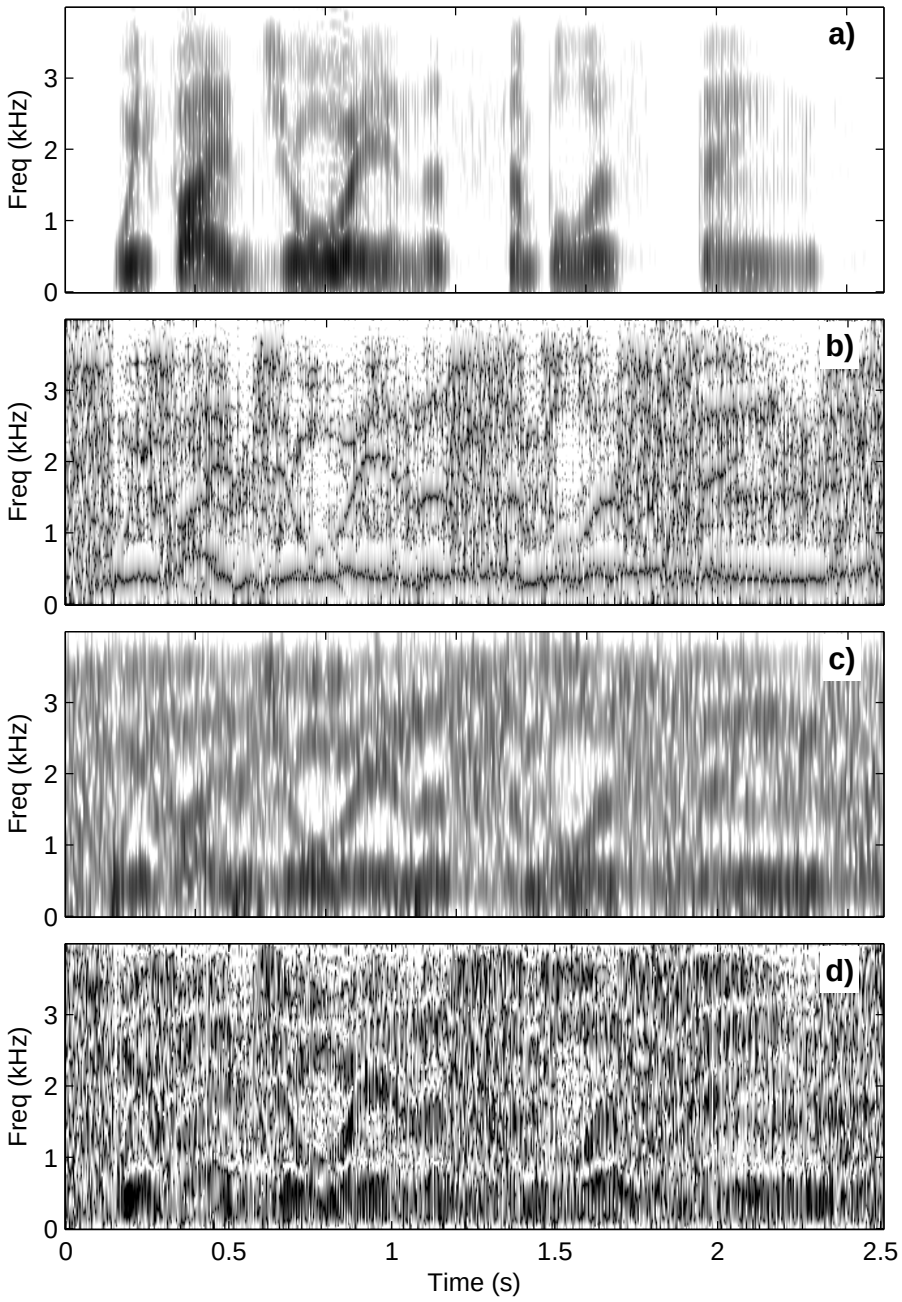

Figure 6.2: Wide-band (4 ms frames) GD based spectrograms for a speech utterance. Subplots: (a) magnitude spectrogram, (b) instantaneous frequency deviation, (c) chirp group delay spectrogram and (d) inverse group delay deviation spectrogram. Stimulus is an 8kHz sentence: 'we find joy in the simplest things', spoken by a male speaker.

Chapter 7

Speech enhancement using the Fourier phase spectrum

7.1 Introduction

The speech enhancement field is concerned with the reduction of noise interference from acoustic speech. The primary goal here is to improve speech quality and intelligibility for human listeners. Several methods have been reported in the literature. This includes spectral subtraction [14], MMSE estimation [39, 40], Wiener filtering [137], Kalman filtering [102] and subspace [43] methods. It can be noted that the above algorithms lack a formal process for enhancing the Fourier phase spectrum directly. In fact, the popular spectral subtraction and MMSE (spectral amplitude, log-spectral amplitude, spectral Wiener) methods operate solely on the short-time amplitude spectrum. In this chapter, we investigate the direct use of phase spectrum manipulation as a means of speech enhancement.

For this work, we consider an additive background noise relationship. This can be represented as

$$y(n) = x(n) + d(n), \tag{7.1}$$

where $y(n)$, $x(n)$ and $d(n)$ are the discrete-time signals of noisy speech, clean speech and noise, respectively. Since speech is typically assumed to be quasi-stationary over short (4-32 ms) intervals, it is analysed frame-wise in the analysis-modification-synthesis (AMS) framework via discrete short-time Fourier analysis. The discrete short-time Fourier transform (DSTFT) of the corrupted speech signal $y(n)$ is given by

$$Y(m,k) = \sum_{n=-\infty}^{\infty} y(n)w(mS-n)\exp(-j2\pi kn/K), \tag{7.2}$$

where k denotes the k'th discrete-frequency of K uniformly spaced frequencies, $w(n)$ is an analysis window function, m is the analysis frame index and S is the frame shift in samples. In spectral domain, the noise relationship is given as

$$Y(m,k) = X(m,k) + D(m,k), \tag{7.3}$$

where $Y(m,k)$, $X(m,k)$, and $D(m,k)$ are the DSTFTs of noisy speech, clean speech, and noise, respectively. Each of the above short-time spectral segments can be expressed in terms of the amplitude spectrum and the phase spectrum. For instance, the DSTFT of the noisy speech signal can be written in polar form as

$$Y(m,k) = |Y(m,k)|e^{j\angle Y(m,k)}, \tag{7.4}$$

where $|Y(m,k)|$ denotes the noisy speech amplitude spectrum and $\angle Y(m,k)$ denotes the noisy speech phase spectrum. Given the noisy speech amplitude and phase spectra, our goal becomes the estimation of the clean speech DSTFT spectrum $X(m,k)$.

Traditional AMS-based speech enhancement methods enhance only the amplitude spectrum $|Y(m,k)|$ while leaving the noisy phase spectrum $\angle Y(m,k)$ intact [8]. In the present work we take the opposite approach – modifying the noisy phase spectrum and leaving the noisy amplitude spectrum unchanged. Noise suppression is achieved by altering the DSTFT phase spectrum in such a way as

to induce large synthesis cancellation among noise components during the inverse DSTFT operation. A preliminary study of this noise suppression technique was reported in [138]. In this chapter, we extend this basic noise suppression mechanism, enabling it to work within an online speech enhancement framework. Here, we formulate a procedure that uses a noise-driven heuristic to control the degree of phase spectrum modification. Using an objective speech quality measure and spectrogram analysis, we demonstrate that the proposed method compares favourably to other popular speech enhancement techniques.

The remainder of this chapter is organised as follows. Section 7.2 provides an overview of the phase-based noise suppression mechanism reported [138]. We then extend this framework to allow online noise suppression. In Section 7.3 we describe the experimental setup and present results. Finally, concluding statements are given in Section 7.4.

7.2 Proposed method

The phase-based noise suppression reported in [138] utilizes the AMS framework commonly employed in speech processing. The AMS framework consists of follow stages:

1. An analysis stage, where speech undergoes DSTFT analysis.

2. A modification stage, where the noisy speech undergoes some form of modification.

3. A synthesis stage, where the inverse discrete short-time Fourier transform (IDSTFT) operation performed, followed by overlap-add synthesis.

A block diagram of the phase-based speech enhancement method is shown in Fig. 7.1. In the above approach, attenuation is achieved by altering the relationship between conjugate DSTFT pairs. These pairs arise naturally as a result of taking the DSTFT of a real-valued signal, i.e. $Y(m,k) = Y^*(m,K-k)$. In the current

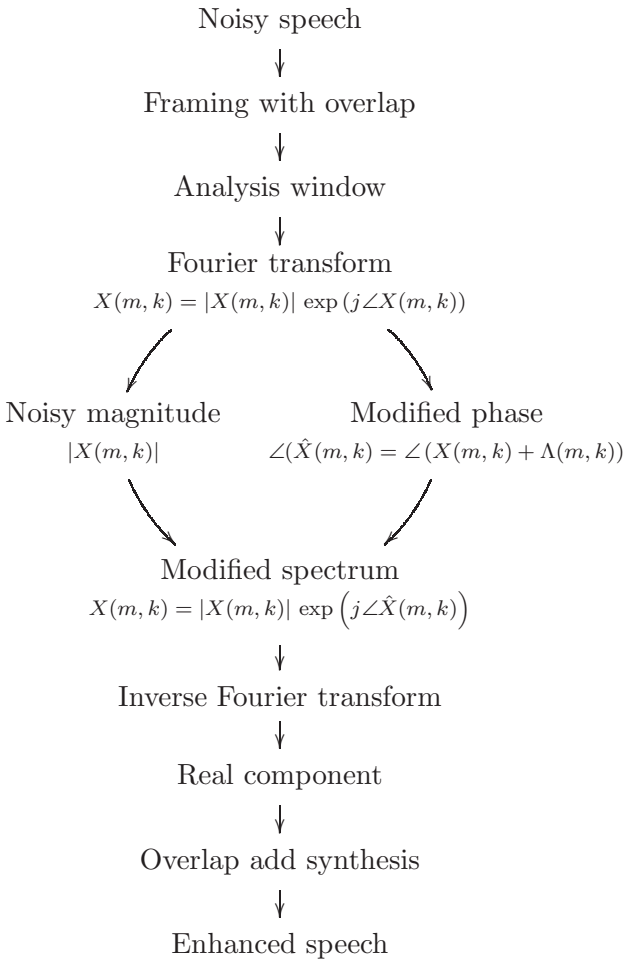

Figure 7.1: Diagram of the phase-based speech enhancement method.

work, we provide a mechanism for linking the phase-modification function with noise estimates. Unlike the method reported in [138], the method detailed here facilitates the handling of time and/or frequency varying noise conditions.

Our modified short-time phase spectrum is computed as follows. First, we obtain the phase spectrum modification function given by

$$\Lambda(m,k) = \lambda\Psi(k)|\widehat{D}(m,k)|, \tag{7.5}$$

where λ is a real-valued empirically determined constant, $\Psi(k)$ is an antisymmetric function and $|\widehat{D}(m,k)|$ is an estimate of the noise amplitude spectrum.[1] The time-invariant antisymmetry function is given by

$$\Psi(k) = \begin{cases} 1, & \text{if } 0 < k/N < 0.5 \\ -1, & \text{if } 0.5 < k/N < 1 \\ 0, & \text{otherwise,} \end{cases} \tag{7.6}$$

where zero weighting is given to the values corresponding to non-conjugate vectors of the DSTFT (i.e. the k=0 value and possible singleton at k=$K/2$ for K=even). Since noise amplitude estimate $|\hat{D}(m,k)|$ is symmetric, multiplication by $\Psi(k)$ produces an antisymmetric $\Lambda(m,k)$ function. It is this antisymmetry that forms the primary basis for noise cancellation during synthesis. The next step in the computation of the modified phase spectrum is to offset the complex spectrum of the noisy speech by the additive real-valued frequency-dependent $\Lambda(m,k)$ modification function

$$\hat{X}_\Lambda(m,k) = Y(m,k) + \Lambda(m,k). \tag{7.7}$$

The modified phase spectrum is then obtained through

$$\angle\hat{X}(m,k) = ARG\left[\hat{X}_\Lambda(m,k)\right], \tag{7.8}$$

where ARG is the complex angle function. The modified phase spectrum is recombined with the noisy amplitude spectrum to produce a modified complex spectrum

$$\hat{X}(m,k) = |Y(m,k)|e^{j\angle\hat{X}(m,k)}. \tag{7.9}$$

In the synthesis stage, the IDSTFT is used to convert the spectral-domain frames, $\hat{X}(m,k)$, to the time-domain. Due to the additive offset introduced in (7.7), the resulting time-domain frames may be complex. This necessitates the explicit

[1] Note that setting the noise estimate $|\widehat{D}(m,k)|$ in (7.5) to unity for all m,k reduces the proposed algorithm to the approach studied in [138].

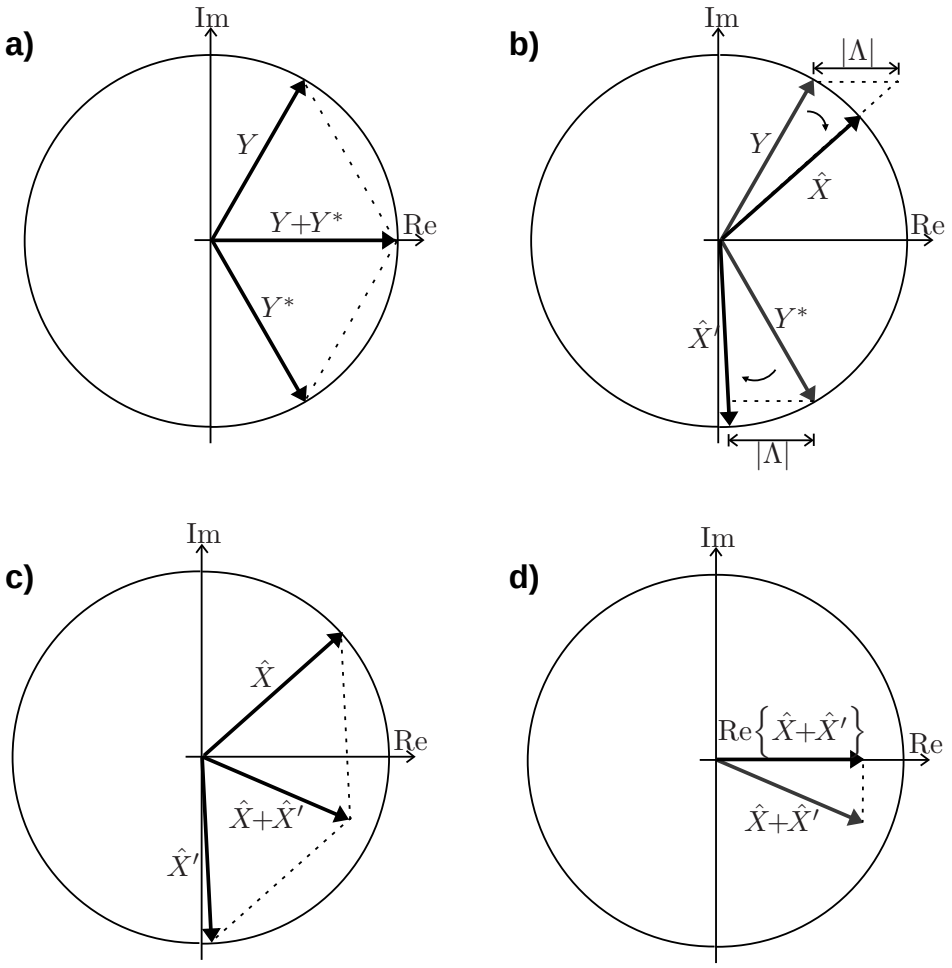

Figure 7.2: Vector diagrams of the phase spectrum modification method. Subplots: a) conjugate vectors, Y and Y^*, as well as their addition vector, $Y{+}Y^*$. b) The real parts of the conjugate vectors are offset by $|\Lambda|$ and $-|\Lambda|$. Thus, the angles of vectors Y and Y^* are altered, while their amplitudes are kept unchanged to produce vectors $\hat{X}$ and $\hat{X}'$, respectively (see (7.9)). c) The resulting vectors are added to produce the $\hat{X}{+}\hat{X}'$ vector. d) The imaginary part of the $\hat{X}{+}\hat{X}'$ is discarded. Note the attenuation of the summed vector: i.e., $\mathrm{Re}[\hat{X} + \hat{X}'] \leq \mathrm{Re}[Y + Y^*]$. For clarity both time and frequency indexes have been omitted in this figure.

removal of any imaginary time-domain component. The enhanced time-domain signal $\hat{x}(n)$, is then produced by employing the overlap-add procedure. We refer to the proposed speech enhancement method as noise driven short-time phase spectrum compensation procedure, or PSC.

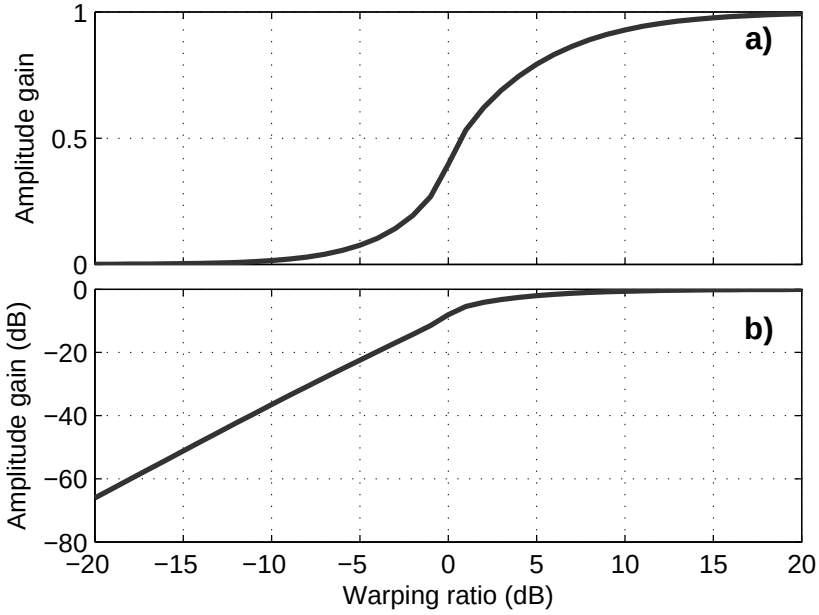

Figure 7.3: The effect of phase warping on spectral amplitude attenuation. Amplitude gain is given as $|\hat{X}(m,k) + X'(m,k)|/|Y(m,k) + Y^*(m,k)|$. Warping ratio (dB) is given as $20\log_{10}(|Y(m,k)/\Lambda(m,k)|)$. Subplots: a) amplitude gain, b) amplitude gain in dB.

Figure 7.2 demonstrates the PSC procedure using vector diagrams for a single conjugate pair. Since $\Lambda(m,k)$ is antisymmetric, the angles of the conjugate pair being considered are pushed in opposite directions, one toward 0 radians and the other toward π radians. The further they are pushed apart, the more out of phase they become. The strength of the modification is dependent on the amplitudes of both the DSTFT vectors and the $\Lambda(m,k)$ function.

Fig. 7.3 highlights the link between $Y(m,k)$, $\Lambda(m,k)$ and the amount of attenuation. Results presented here are averaged over all input phases; i.e., $-\pi \leq \angle Y(m,k) \leq \pi$. It can be seen that attenuation is negatively correlated with the ratio $|Y(m,k)/\Lambda(m,k)|$. Since $\Lambda(m,k)$ is proportional to the noise estimate, when the SNR is high, there is very little warping/cancellation but when the SNR drops, attenuation increases.

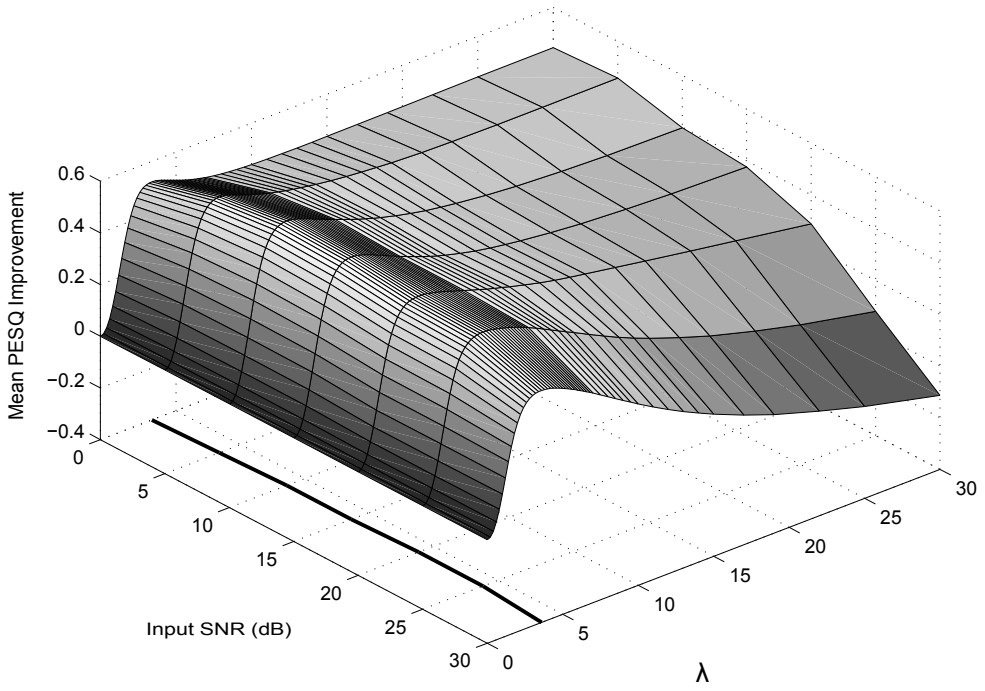

Figure 7.4: Mean PESQ improvement scores as a function of λ scaling factor and input SNR (dB) for white noise. The bold line on the base of the plot indicates λ values that produce maximum mean PESQ improvement scores as a function of SNR. Figure is taken from original publication found in [129].

7.3 Experimental evaluation

7.3.1 Empirical search for optimal λ

The term λ is a tuneable parameter that governs the aggressiveness of noise suppression. As such, it is important to find a value of λ that suppresses noise while having minimal impact on speech. To do this, we used the PSC method to enhance white noise degraded speech. Specifically, our objective was to empirically determine the values of λ that would maximise objective speech quality (in terms of the PESQ measure [117]), for several SNR settings. We used the core test set of the TIMIT speech corpus [55] (192 speech files from 24 speakers) for this purpose.

From the results shown in Fig. 7.4, it can be seen that $\lambda = 3.74$ produces maximum objective speech quality. The resulting relationship between κ and SNR

was remarkably constant across the entire SNR range tested. This is encouraging, as it demonstrates not only an elegant relationship between the level of noise and phase modification, but also a degree of robustness in choosing λ. Figure 7.5 shows an example of $\Lambda(m,k)$ function generation based on a 120 ms sample of F16 noise. The antisymmetric function scaled by λ=3.74 is then applied to the noise estimate to produce the phase spectrum modification function (7.5).

7.3.2 Enhancement experiments

The enhancement experiments were carried out on the core test set of the TIMIT corpus [55]. We artificially add white noise, F16 noise and babble noise at several SNRs to the testing stimulus. Noise signals are from the NOISEX-92 noise database [134] and down-sampled to 16 kHz.

Noise amplitude estimates $|\hat{D}(m,k)|$, were computed from the initial 120 ms of each utterance. The corrupted files were then enhanced using the proposed (PSC) method with $\lambda = 3.74$ in all cases. For all experiments, a Hamming window is used for DSTFT analysis. Three other speech enhancement techniques were also used, namely the spectral subtraction method [14], the MMSE spectral amplitude (SA) method [39] and the MMSE log-spectral amplitude (LSA) method [40]. Mean PESQ scores are given for each noise type and SNR.

7.3.3 Results and discussion

The results of the enhancement experiments, in terms of mean PESQ scores as well as mean PESQ improvement scores, are shown in Table 7.1 and Figs. 7.6a), 7.6b) and 7.6c) respectively. Compared to other methods, the proposed method performed well, providing consistent improvements across all SNRs. As a comparison, the LSA method performed quite well at low SNRs, but caused speech quality degradation at higher SNRs. Results for F16 noise are very similar to the white noise case, showing that the empirically determined λ also works well for coloured noises. For the babble noise case, the proposed method performed slightly better than the other three

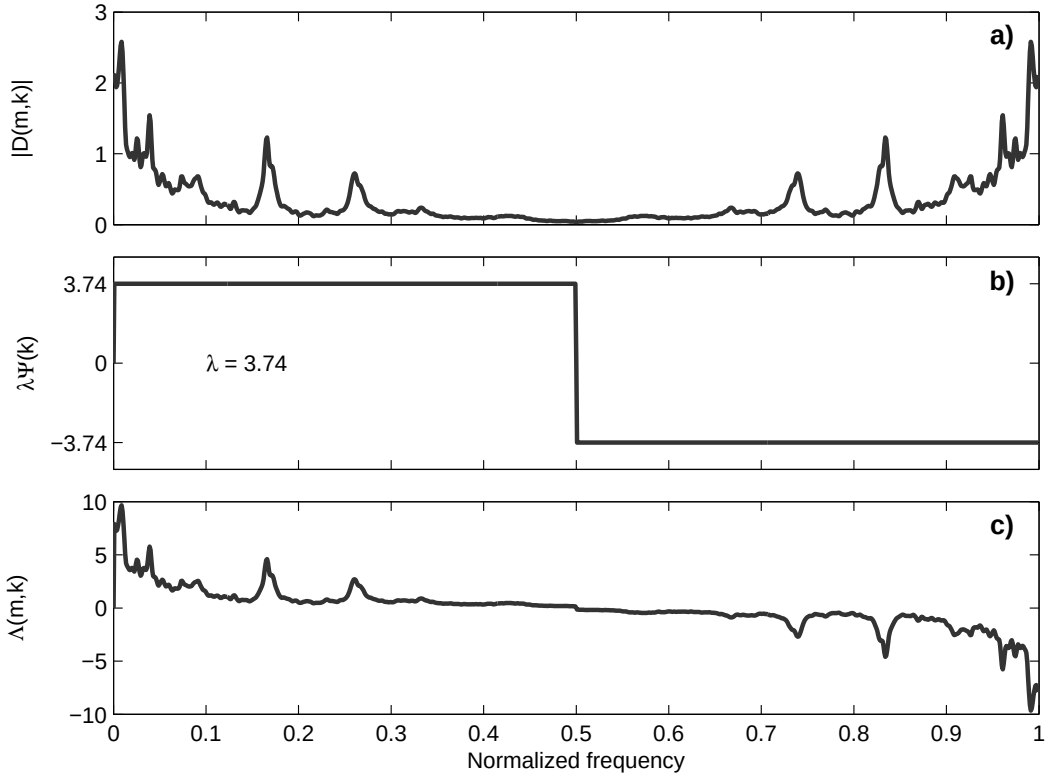

Figure 7.5: Generation of the phase spectrum modification function: (a) noise amplitude spectrum estimate, $|\widehat{D}(m,k)|$, for F16 noise; (b) antisymmetric $\lambda\Psi(k)$ weighting function; and (c) $\Lambda(m,k)$ phase spectrum modification function.

methods in the comparison. An important finding was the relative lack of musical noise in the PSC-enhanced signals. This is likely due to the smooth warping of the spectral-domain speech induced by the PSC method. As a result, there are fewer sharp spectral peaks in the noise residual.

Spectrogram analysis of the PSC method is given in Figs. 7.7 and 7.8, for 10 dB white and 10 dB babble noises respectively. The sentence is: 'the sky that morning was clear and bright blue', spoken by a male speaker and digitized at 8 kHz. The enhanced versions show little loss of speech information despite the background noise suppression. For the white noise case, it should be noted that although some background noise remains, it is fairly white and lacks most of the musical noise artefacts introduced by some of the other speech enhancement methods.

		SNR (dB)						
	Treatment	**0**	**5**	**10**	**15**	**20**	**25**	**30**
White noise	**None**	1.55	1.90	2.26	2.62	2.97	3.32	3.65
	PSC	2.06	2.47	2.87	3.23	3.56	3.88	4.14
	SS	1.78	2.28	2.73	3.18	3.58	3.89	4.12
	SA	2.03	2.40	2.74	3.05	3.33	3.61	3.87
	LSA	2.12	2.54	2.88	3.18	3.45	3.71	3.96
F16 noise	**None**	1.67	2.03	2.38	2.73	3.08	3.42	3.72
	PSC	2.16	2.55	2.92	3.26	3.59	3.88	4.10
	SS	1.88	2.34	2.76	3.19	3.59	3.89	4.09
	SA	2.10	2.46	2.78	3.09	3.37	3.64	3.89
	LSA	2.20	2.57	2.90	3.19	3.46	3.72	3.96
Babble noise	**None**	1.75	2.08	2.43	2.77	3.10	3.43	3.74
	PSC	1.97	2.34	2.71	3.08	3.42	3.73	3.98
	SS	1.68	2.13	2.56	2.97	3.36	3.69	3.94
	SA	1.94	2.28	2.64	2.97	3.28	3.58	3.84
	LSA	1.94	2.31	2.67	3.02	3.33	3.62	3.88

Table 7.1: Mean PESQ scores for the proposed (PSC), spectral subtraction (SS), MMSE spectral amplitude (SA) and MMSE log-spectral amplitude (LSA) methods. Higher PESQ scores correlate with better subjective speech quality. Results are based on those presented in the publication [129].

7.4 Conclusion

In previous work, it has been shown that signal attenuation may be achieved via manipulation of the short-time phase spectrum. In this work, we have presented a noise-driven heuristic to control this attenuation mechanism – allowing it to be used as an online speech enhancement algorithm. The presented method was objectively evaluated using the PESQ measure and it was shown to perform well under a variety of noisy conditions. Though the PSC method fundamentally affects an attenuation

upon the magnitude spectrum, noise suppresion was achieved by smoothly warping the short-time Fourier phase spectrum. We believe this is the primary reason why the PSC method exhibits relativly mild musical noise artefacts in comparison to the direct (i.e. SA and LSA) methods of attenuation.

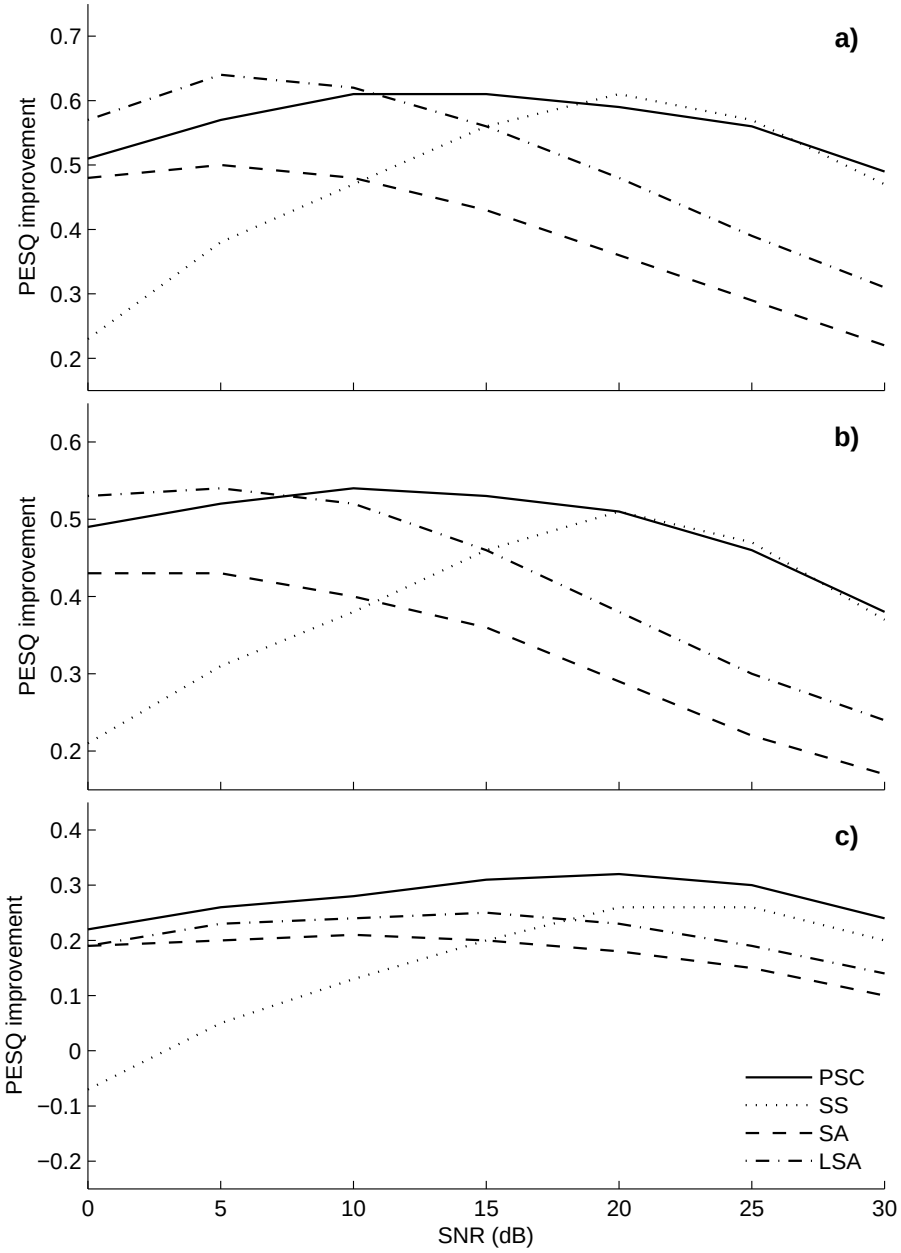

Figure 7.6: Mean PESQ improvement scores for the proposed (PSC) method, spectral subtraction (SS) method, MMSE spectral amplitude (SA) method and MMSE log-spectral amplitude (LSA) method, under a) white noise b) F16 and c) babble noise.

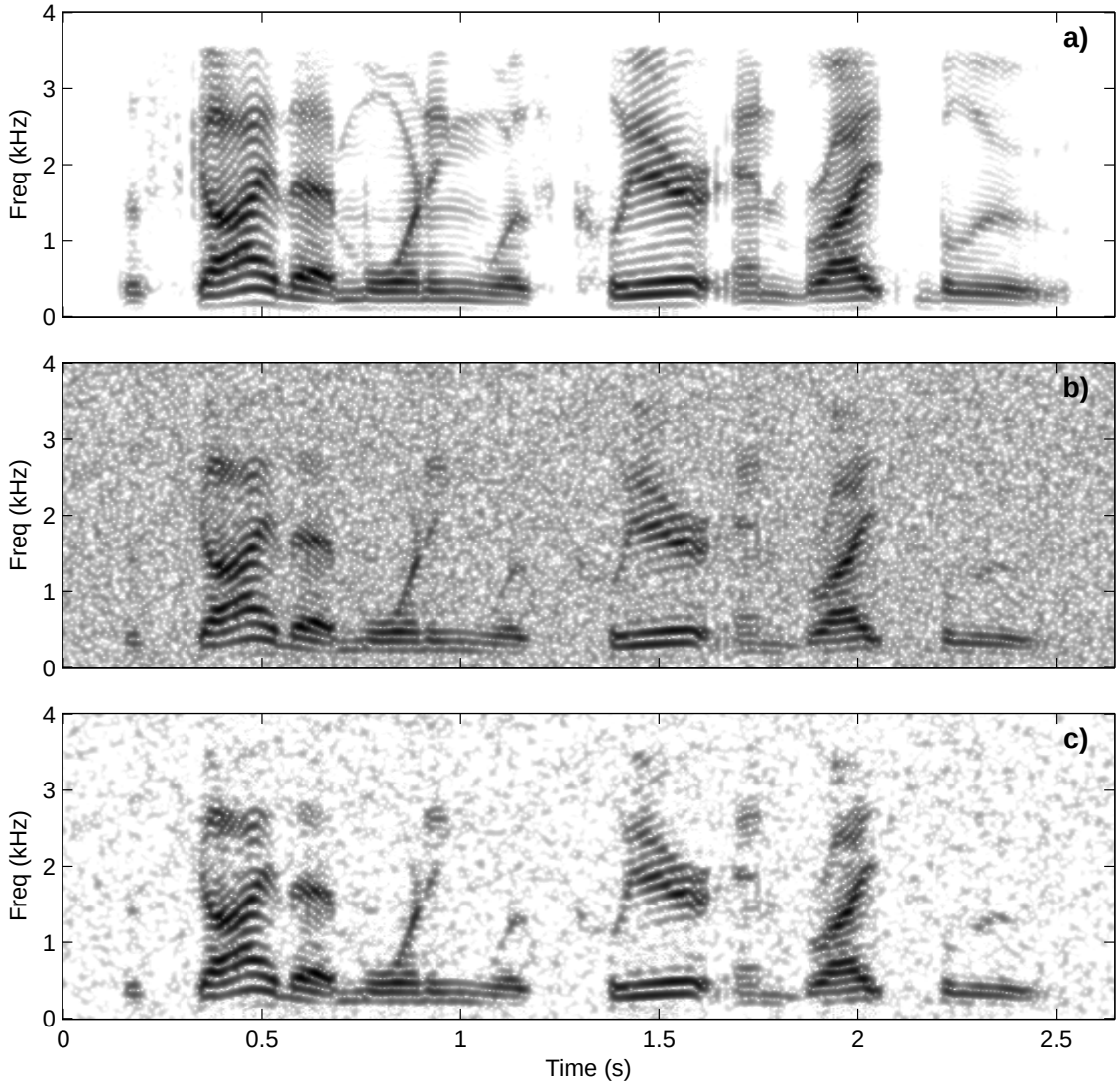

Figure 7.7: Spectrograms for white noise degraded speech enhanced with the PSC method. Subplots: a) clean speech, b) speech degraded with 10 dB white noise and c) degraded speech enhanced with PSC.

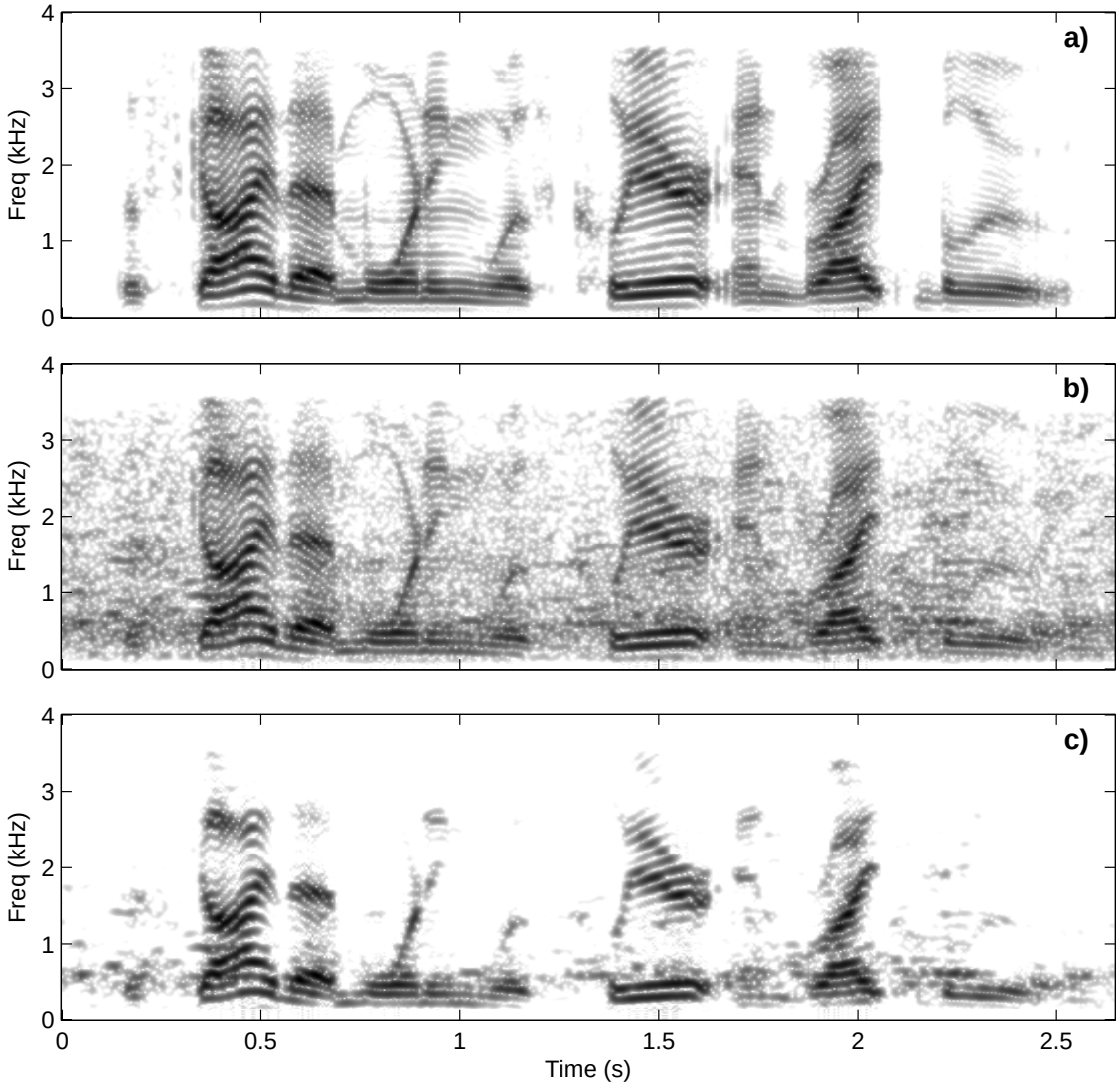

Figure 7.8: Spectrograms for babble noise degraded speech enhanced with the PSC method. Subplots: a) clean speech, b) speech degraded with 10 dB babble noise and c) degraded speech enhanced with PSC.

Part II

Environment-robust estimation of Mel-frequency cepstral coefficients

Chapter 8

Heuristic modification of the MMSE spectral energy estimator

8.1 Introduction

The development of robust automatic speech recognition (ASR) is an important goal. While the performance of ASR is generally sufficient in laboratory conditions, its recognition accuracy tends to degrade rapidly in the presence of additive background noise. Since it is often impossible to eliminate all noise from the operating environment, the problem of ASR robustness has been receiving considerable attention. Several approaches to robustness have been proposed in the literature, most of which fall under two categories: front-end speech / feature enhancement and back-end model adaptation. Back-end adaptation seeks to modify the acoustic models of the recognizer to better match the noisy operating environment. Font-end enhancement on the other hand, seeks to remove the effects of noise prior to recognition, either from the speech signal or from the parameterized features directly.

In this chapter, we are interested in methods that perform enhancement on the speech signal. Several methods falling into this category have been reported

in the literature [80]. This includes spectral subtraction [14], MMSE estimation [40], Wiener filtering (linear MMSE) [137], Kalman filtering [102] and subspace [43] methods. These algorithms are specifically designed to improve the subjective quality of an acoustic signal for human listeners. For example, the MMSE log-spectral amplitude (LSA) estimator is often favoured because of its psychoacoustic considerations. While many of the aforementioned algorithms have been used in ASR [50, 57, 67, 77], there are clear differences between the objectives of robust machine recognition and speech enhancement.

For subjective human listening, it is often held that noise suppression is most effective when applied to the log-spectral-domain. The LSA estimator was derived under such an assumption. However, the typical ASR system does not operate directly on the log-spectral-domain. Instead, higher level features such as Mel-frequency cepstral coefficients (MFCCs) are used. As a result, the complicated suppression rule of the LSA estimator may not be fully justified for use in ASR based speech enhancement.

In this chapter, we examine a similar estimator for use in robust ASR; namely the spectral energy (SE) estimator. Specifically, we investigate its suitability for estimating clean speech MFCCs, from speech corrupted with additive noise. We show that the suppression rule of the SE estimator is closely related to the minimum mean square error (MMSE) MFCC estimator. That is, an estimator that produces a cepstral estimate $\hat{\boldsymbol{c}}_x$ that minimizes the square error from the true, clean MFCC vector $\boldsymbol{c}_x$. Despite this, the SE estimator has several shortcomings that must be addressed before it can be used for robust ASR. Foremost among these problems is its tendency to under-suppress noise at low signal to noise ratios (SNRs). We identify two causes of this under-suppression: 1) an inherent positive bias when using the SE estimator to derive log-filterbank energies and 2) the tendency of the SE estimator to over-estimate the *a priori* SNR within a decision-directed framework [39]. Later, we show that both of these issues may be corrected with the use of heuristic based adaptations. For the first adaptation, we use the speech presence uncertainty

framework. For the second proposed adaptation, we employ a direct modification to the SE estimators spectral suppression profile. The heuristic estimators offer a number of advantages over the LSA estimator. First, the suppression rules are more efficiently implemented and second, they can offer better recognition performance across a wide range of noise types and SNRs.

The rest of this chapter is organized as follows. In Section 8.2, we cover the statistical framework used to derive the common short-time spectral amplitude estimators. In Section 8.3, we investigate the use of the SE estimator for deriving MFCC features. Firstly, we examine the optimality of the SE estimator in the context of MFCC estimation. Secondly, we highlight the considerations that must be taken into account for practical implementation of the SE estimator. In Section 8.4, we first describe the use SPU within the spectral estimation framework. We then show how SPU may be used to overcome the limitations of the SE estimator. In Section 8.5, we examine the family of spectral amplitude estimators and propose a heuristic modification to the SE estimator spectral gain profile. In Section 8.6, we present experimental ASR results for the RM [110], OLLO2 [136] and Aurora2 [104] ASR tasks. Lastly in Section 8.7, we present concluding remarks.

8.2 Statistical framework for short-time spectral amplitude estimation

The discrete short-time Fourier transform (DSTFT) of corrupted speech signal $y(n)$ is given by

$$Y(m,k) = \sum_{n=-\infty}^{\infty} y(n)w(mS-n)\,\exp\left(-j2\pi kn/K\right), \tag{8.1}$$

where k denotes the k'th discrete frequency of K uniformly spaced frequencies, $w(n)$ is an analysis window function, m is the short-time frame index and S is the analysis frame shift (in samples). In this chapter, we consider an additive noise model. Here,

the corrupted speech DSTFT may also be represented as[1]

$$Y(k) = X(k) + D(k), \tag{8.2}$$

where $X(k)$ and $D(k)$ are the DSTFT expansion coefficients for the k'th discrete frequency bin of the clean speech signal and noise signals respectively.

DSTFT expansion coefficients $X(k)$ and $D(k)$ are assumed to be independent complex zero-mean Gaussian variables, with expected power $\lambda_x(k) = E\left[|X(k)|^2\right]$ and $\lambda_d(k) = E\left[|D(k)|^2\right]$, where $E[.]$ is the expectation operator. A detailed justification of this statistical assumption may be found in [39].

The general goal of speech enhancement is to derive an estimate of the clean speech given the observed noisy speech and a noise estimate. To accomplish this, it can be useful to split DSTFT coefficients $Y(k)$ and $X(k)$, into spectral amplitude and phase:

$$Y(k) = R(k)\, \exp\left(j\vartheta(k)\right), \tag{8.3}$$

$$X(k) = A(k)\, \exp\left(j\theta(k)\right), \tag{8.4}$$

where $R(k)$ and $\vartheta(k)$ are the amplitude and phase spectra of the noisy speech respectively, while $A(k)$ and $\theta(k)$ are the amplitude and phase spectra[2] of the clean speech respectively. For typical Fourier analysis-modification-synthesis (AMS) based speech enhancement [17, 39, 40], an estimate $\hat{A}(k)$ is obtained from the noisy signal and combined with the noisy spectral phase $\vartheta(k)$ to produce the estimated clean speech spectrum $\hat{X}(k)$

$$\begin{aligned} \hat{X}(k) &= \hat{A}(k)\, \exp\left(j\vartheta(k)\right) \\ &= Y(k).G(k), \end{aligned} \tag{8.5}$$

[1] For notational convenience, we have dropped the frame index m and dependence on this subscript is implicitly assumed unless stated otherwise.

[2] DSTFT modifiers are implicitly assumed when referring to amplitude and phase spectra.

where

$$G(k) = \frac{\hat{A}(k)}{R(k)}, \tag{8.6}$$

is the spectral amplitude gain of the speech enhancement system and $\hat{A}(k)$ is the estimated clean DSTFT coefficient amplitude. Using the DSTFT estimate $\hat{X}(k)$, enhanced time-domain speech may then be synthesized with an inverse discrete Fourier transform (IDFT) and overlap-add-synthesis [29]. A block diagram of the typical AMS-based speech enhancement framework is given in Fig. 8.1.

Several spectral amplitude estimators have been suggested in the literature. The spectral amplitude gain functions for the spectral Wiener (SW) [80], MMSE spectral amplitude (SA) [39], MMSE log-spectral amplitude (LSA) [40] and MMSE spectral energy (SE) filters are given below:

$$G_{SW}(k) = \frac{\xi(k)}{1+\xi(k)} = \frac{\lambda_x(k)}{\lambda_x(k)+\lambda_d(k)}, \tag{8.7}$$

$$G_{SA}(k) = \frac{\sqrt{\pi\nu(k)}}{2\gamma(k)} \exp\left(\frac{-\nu(k)}{2}\right).\left[(1+\nu(k))I_0\left(\frac{\nu(k)}{2}\right)+\nu(k)I_1\left(\frac{\nu(k)}{2}\right)\right], \tag{8.8}$$

$$G_{LSA}(k) = \frac{\xi(k)}{1+\xi(k)} \exp\left(\frac{1}{2}\int_{\nu(k)}^{\infty}\frac{\exp(-t)}{t}dt\right), \tag{8.9}$$

$$G_{SE}(k) = \frac{\xi(k)}{1+\xi(k)}\sqrt{1+\frac{1}{\nu(k)}}, \tag{8.10}$$

where,

$$\nu(k) = \frac{\xi(k)}{1+\xi(k)}\gamma(k), \tag{8.11}$$

$$\xi(k) = \frac{\lambda_x(k)}{\lambda_d(k)}, \tag{8.12}$$

$$\gamma(k) = \frac{[R(k)]^2}{\lambda_d(k)}. \tag{8.13}$$

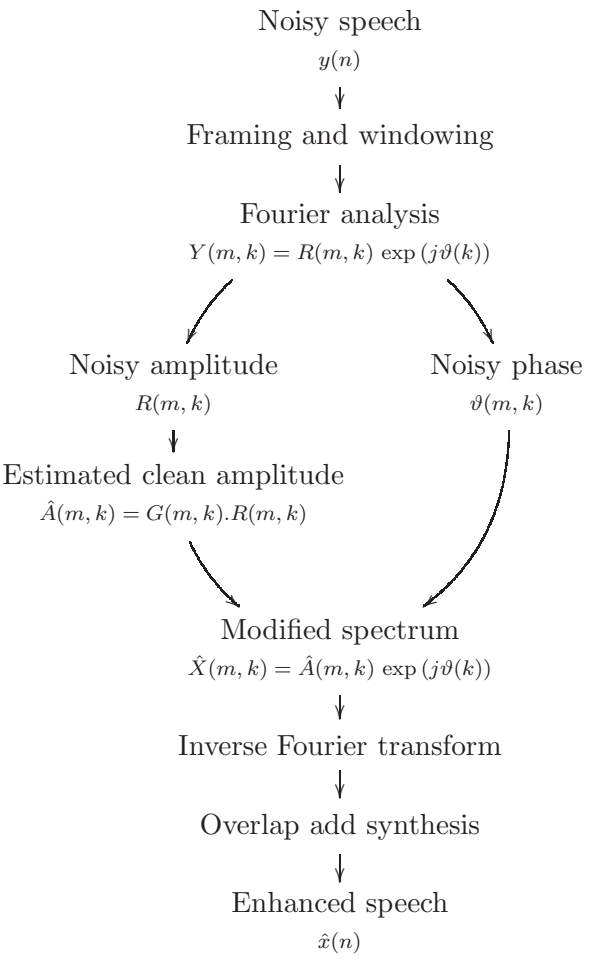

Figure 8.1: Diagram of AMS based speech enhancement using a spectral amplitude estimator.

The parameters ξ and γ are interpreted as the *a priori* signal to noise ratio (SNR) and *a posteriori* SNR respectively. $I_0(.)$ and $I_1(.)$ are given as the zeroth and first order modified Bessel functions respectively. Fig. 8.2 shows the spectral amplitude gain functions for each estimator over several SNR values. Interestingly, all gains become equivalent to the spectral Wiener gain at high SNRs; i.e., when $\nu(k) \gg 1$.

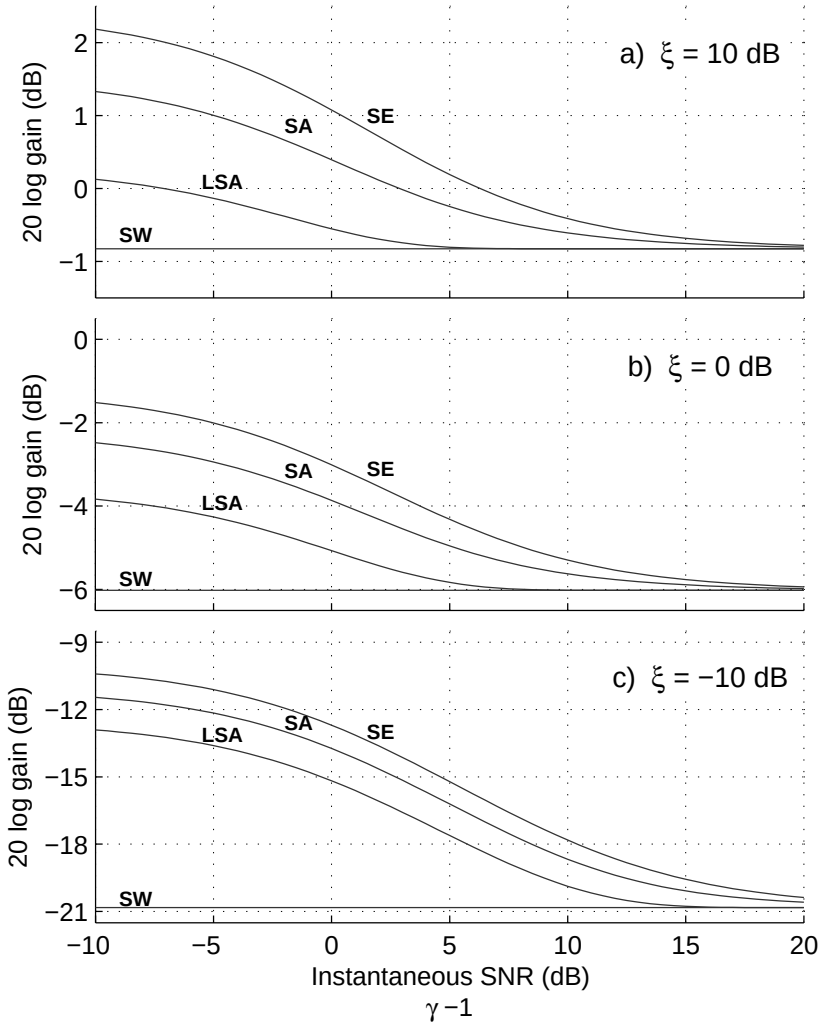

Figure 8.2: Common spectral amplitude gain functions. Gain functions shown are: spectral energy (SE), spectral amplitude (SA), log-spectral amplitude (LSA) and spectral Wiener (SW) estimators. Subplots: a) *A priori* SNR = 10 dB, b) *a priori* SNR = 0 dB, c) *a priori* SNR = -10 dB.

8.3 Use of the SE estimator for ASR

As stated earlier, our goal in this chapter is to investigate the SE estimator for use in ASR-based speech enhancement. One immediate justification for this is the simple gain rule (8.10), which requires less computation than both the SA and LSA

estimators. A second reason for investigating the spectral energy estimator, is that it is closely related to the log-filterbank energy estimator – the intermediate stage of the popular MFCC feature set [69].

Despite these reasons, use of the SE estimator is not common in the ASR field. This is largely due to its poor noise suppression in low SNR environments. Consequently, this problem must be understood and compensated if the SE estimator is to be used in ASR. Two causes of noise under-suppression are identified in the remainder of this section.

8.3.1 Sub-optimality of the SE estimator for generating MFCCs

Since MFCCs are currently the dominant speech parameterization, a suitable goal for ASR-centric speech enhancement is optimal estimation of the MFCC vector. In this sub-section we determine the relationship between the SE estimator and the optimal MMSE MFCC estimator. The optimal MMSE MFCC estimator is given as

$$\hat{\boldsymbol{c}}_x = E\left[\boldsymbol{c}_x|\boldsymbol{Y}\right], \tag{8.14}$$

where $\hat{\boldsymbol{c}}_x \in \mathbb{R}^{Q\times 1}$ is the MFCC estimate that minimizes the square error from the true, clean MFCC vector $\boldsymbol{c}_x$ and $\boldsymbol{Y} = [Y(0), Y(1), ..., Y(K-1)]^T$ is a spectral frame of noisy speech. The MFCC vector is related to the log-filterbank energy vector via the discrete cosine transform (DCT). Since the DCT is a unitary operator, the total squared error in both the MFCC and log-filterbank domains is equivalent. This allows us to recast our problem into MMSE estimation of log-filterbank energies

$$\hat{\boldsymbol{c}}_x = \boldsymbol{C}.E\left[\boldsymbol{L}_x|\boldsymbol{Y}\right], \tag{8.15}$$

where $\boldsymbol{L}_x$ is the clean speech log-filterbank energy and $\boldsymbol{C} \in \mathbb{R}^{Q\times Q}$ is the DCT matrix. Under normal circumstances, higher order cepstral coefficients are truncated from the DCT matrix [69]. Strictly speaking, this means the total mean square error is not same between the log-filterbank and cepstral domains. However it remains

a good approximation since truncated coefficients themselves tend to have very small energies. Continuing, we may now directly determine the suitability of the SE estimator for ASR. The SE criterion for estimating clean spectral amplitudes is given by

$$\hat{A}(k) = \sqrt{E\left[[A(k)]^2|\boldsymbol{Y}\right]}. \tag{8.16}$$

Assuming filterbank energies are accumulated off spectral energies (and not amplitudes), log-filterbank energies will be given by

$$\begin{aligned} \hat{L}_x^{SE}(q) &= \log\left(\sum_k h(q,k) E\left[[A(k)]^2 \middle| \boldsymbol{Y}\right]\right) \\ &= \log E\left[\left(\sum_k h(q,k)[A(k)]^2\right)\middle|\boldsymbol{Y}\right], \end{aligned} \tag{8.17}$$

where $\hat{L}_x^{SE}(q)$ is the SE estimate of the q'th log-filterbank energy and $h(q,k)$ is the filterbank gain of the q'th filterbank and k'th frequency bin. By Jensen's inequality we can show that these estimates will be positively biased, i.e.

$$E\left[\log\left(B(q)\middle|\boldsymbol{Y}\right)\right] \leq \log E\left[B(q)\middle|\boldsymbol{Y}\right], \tag{8.18}$$

where filterbank energy $B(q) = \sum_k h(q,k)[A(k)]^2$. The first term of (8.18) is the ideal (MMSE) log-filterbank energy estimate, and the second term is the estimate produced by the SE estimator. Both terms are quite similar, differing by only the position of the logarithm. The positive bias arises from the fact that this logarithm is a concave function. If a convex operator was used instead, the inequality would be reversed. If a linear operator was used, then the inequality would become an equality. This is important, because over small dynamic ranges, the logarithm is approximately linear. Fig. 8.3 shows this. Here, both filterbank variables B_1 and B_2 have the same mean, but different variances. The variable B_1 exists on a fairly small dynamic range. Over this range the logarithm is approximately linear. The variable B_2 exists on a much larger dynamic range. Over this range the logarithm

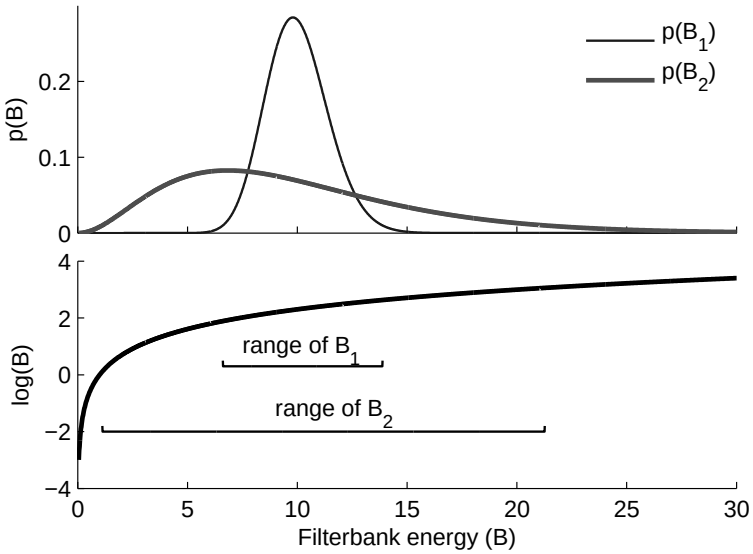

Figure 8.3: Effect of the logarithm on a filterbank variable. Top figure shows probability density functions for two filterbank variables B_1 and B_2. Variable B_1 exists on a small dynamic range, so the logarithm is a predominantly linear effect. Variable B_2 has much larger dynamic range, over which the logarithm is highly concave.

is highly concave.

This suggests that the SE estimator would perform quite well at higher SNRs – conditions where there is little uncertainty / variance in the estimation of filterbanks $B(q)$. In such a case, the concavity of the logarithm would play a minor role, meaning the SE estimator would (effectively) produce unbiased log-filterbank energies. Conversely, at lower SNRs we would expect the positive bias to become worse. Large amounts of noise energy will introduce large variance into the estimation of $B(q)$ – making the logarithm a highly non-linear, concave operation.

8.3.2 Considerations for estimation of *a priori* SNR

A more practical consideration of the SE estimator involves estimation of the *a priori* SNR parameter ξ. While the estimation of *a posteriori* SNR γ is relatively straightforward, several considerations must be taken into account when estimating

ξ. The typical method for estimating ξ is the recursive, decision-directed approach presented in [39]. This approach assumes the *a priori* SNR to be a slowly evolving parameter. Here the *a priori* SNR for the m'th analysis frame and k'th frequency bin $\xi(m,k)$, is estimated as the weighted sum of two terms

$$\xi(m,k) = \alpha \frac{[G(m-1,k)R(m-1,k)]^2}{\lambda_d(m-1,k)} + (1-\alpha)G\left[\gamma(m,k)-1\right], \tag{8.19}$$

where mixing constant $\alpha \approx 0.98$ and

$$G[x] = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{8.20}$$

The term $\Psi\left[\gamma(m,k)-1\right]$ in (8.19) can be interpreted as the instantaneous SNR, and is derived as the maximum likelihood estimate of ξ. The proceeding term in (8.19) is an estimate of ξ derived from the previous, enhanced frame. The inclusion of spectral gain $G(m-1,k)$ makes this term highly dependent on the enhancement regime. This is of particular concern for the SE estimator, as it has a relatively mild spectral gain (w.r.t the SA and LSA estimators, see Fig. 8.2). Because of this, residual noise often makes its way into the *a priori* SNR estimate. This leads to over-estimation of ξ, which itself leads to less suppression and more residual noise in subsequent frames. It should be noted that while we have used a relatively simple method for estimating the a priori SNR, more advanced methods [**?**] are easily substituted into the estimation framework.

8.4 Use of speech presence uncertainty to improve the spectral energy estimator

In the previous section, we have highlighted two causes of noise under-suppression in the SE estimator:

1. The inherent positive bias of the SE estimator to derive log-filterbank energies.

2. The tendency to over-estimate the *a priori* SNR ξ within the decision-directed framework.

Combined, these problems degrade ASR performance substantially in low SNR environments. To address both of these problems, we investigate the use of speech presence uncertainty (SPU) [85].

8.4.1 Overview of speech presence uncertainty within the spectral estimation framework

SPU does not assume speech to be present at all times and at all frequencies. Instead, speech presence is represented as a probabilistic variable. A two-state speech (absent/present) hypothesis can be incorporated into the conditional probability density function (PDF) $p(A(k)|Y(k))$ as follows

$$\begin{aligned} p(A(k)|Y(k)) = p(H_0(k)|Y(k)).p(A(k)|Y(k), H_0(k)) \\ + p(H_1(k)|Y(k)).p(A(k)|Y(k), H_1(k)), \end{aligned} \tag{8.21}$$

where $H_0(k)$ and $H_1(k)$ represent the hypotheses of speech absence and presence respectively for the k'th frequency bin. Given an *a posteriori* probability of speech presence $\varphi(k) \triangleq p(H_1(k)|Y(k))$ for the k'th frequency bin, a SPU modified conditional PDF $p(A(k)|Y(k))$ can be given as follows

$$p(A(k)|Y(k)) = \varphi(k).p(A(k)|Y(k), H_1(k)) + (1 - \varphi(k)).\delta(A(k)), \tag{8.22}$$

where $\delta(.)$ is the Dirac delta function. Here we have assumed that under signal absence hypothesis $H_0(k)$, the clean spectral amplitude $A(k)$ is most surely zero. The form and derivation of $p(A(k)|Y(k), H_1(k))$ may be found in [39]. The SE SPU estimate for the clean spectral amplitude $\hat{A}(k)$ is now given by

$$\hat{A}(k) = \sqrt{\varphi(k) E\left[[A(k)]^2 | Y(k), H_1(k)\right]}, \tag{8.23}$$

or, as a spectral amplitude gain

$$G_{SE-SPU}(k) = \frac{\sqrt{\varphi(k)\nu(k)[1+\nu(k)]}}{\gamma(k)}. \tag{8.24}$$

The *a posteriori* speech presence probability $\varphi(k)$ is given as [39]

$$\varphi(k) = \frac{\Lambda(k)}{1+\Lambda(k)}, \tag{8.25}$$

where $\Lambda(k)$ is the generalized speech presence ratio.

$$\begin{aligned} \Lambda(k) &= \frac{p(H_1(k))}{p(H_0(k))} . \frac{p(Y(k)|H_0(k))}{p(Y(k)|H_1(k))} \\ &= \frac{1-q(k)}{q(k)} . \frac{\exp\left(\nu(k)\right)}{1+\xi_k}, \end{aligned} \tag{8.26}$$

where $q(k) \triangleq p(H_0(k))$ is given as the *a priori* speech absence probability and is regarded as a tuneable parameter. Fig. 8.4 shows the SE spectral amplitude gain for $q(k) = 0$, $q(k) = 0.3$ and $q(k) = 0.5$. When $q(k) = 0$, the SE SPU gain simplifies to the standard SE estimator, while for $q(k) = 0.5$, the SE estimator is transformed into a very aggressive noise suppressor. It can be seen that this added aggressiveness manifests in regions where *a posteriori* SNR γ is small. However, increasing the sensitivity of spectral gain to the parameter γ can sometimes be undesirable. This is because γ has much higher estimation variance than ξ [21]. For these reasons, it can be useful to limit the maximum aggressiveness of the SPU.

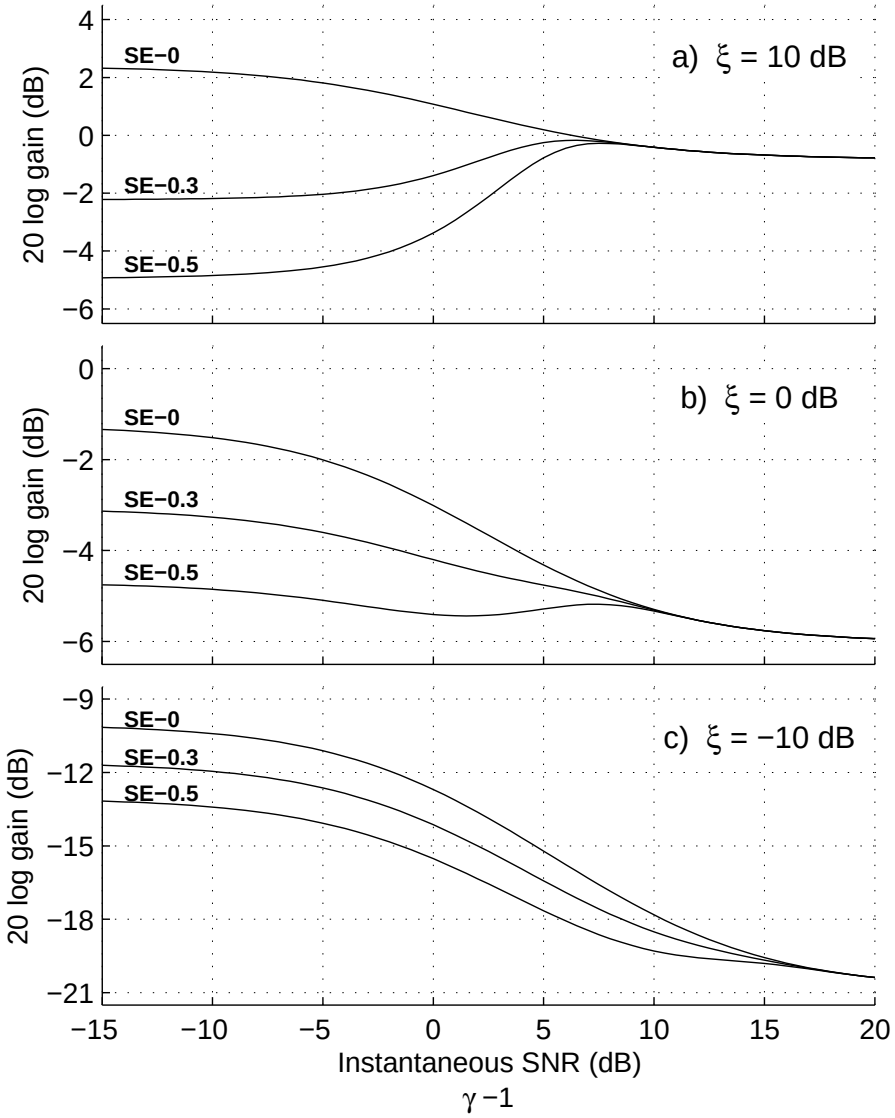

Figure 8.4: Effect of SPU on the spectral energy estimator spectral amplitude gain. Three settings are tested: 0%, 30% and 50% *a priori* likelihood of speech absence. Subplots: a) *a priori* SNR = 10 dB, b) *a priori* SNR = 0 dB, c) *a priori* SNR = -10 dB.

8.4.2 Application of speech presence uncertainty to improve the spectral energy estimator

The net result of SPU is an additional mechanism to suppress noise-like spectral energy. From Fig. 8.4 we can see that increasing the value of $q(k)$ will increase the aggressiveness of the estimator. Doing so would mitigate the bias problem of the SE estimator at lower SNRs. However, this would come at the cost of speech degradation in cleaner conditions where the bias is negligible. In order to make the choice of suppression more flexible, we use a simple noise-driven heuristic to modify the *a priori* speech absence probability $q(k)$

$$q(k) = U\left[\left(1 + \kappa\sqrt{\frac{E\left[[A(k)]^2\right]}{\lambda_d(k)}}\right)^{-1}\right], \tag{8.27}$$

where,

$$U[x] = \begin{cases} x & \text{if } x \leq q_{\max}, \\ q_{\max} & \text{otherwise.} \end{cases} \tag{8.28}$$

The term $E\left[[A(k)]^2\right]$ is an ensemble spectral energy average generated from a clean speech corpus[3] and $\kappa, q_{\max} > 0$ are tuneable parameters. The heuristic presented in (8.27) consists of several components: an ensemble clean speech spectral energy average, a noise power/energy estimate, and scaling parameters κ and $q_{\max}$. The ensemble energy average is introduced to bring frequency dependence to the heuristic. Effectively, its introduction increases the probability of speech presence within *speech-like* frequency regions – typically between 500 and 1000 Hz (see Fig. 8.5). The noise power estimate $\lambda_d(k)$ is the main component of the heuristic and is incorporated to directly address the SE estimator bias problem. As the noise power $\lambda_d(k)$ rises, the aggressiveness of the SPU is increased to compensate. This relationship is controlled further by the variables κ and $q_{\max}$.

Fig. 8.6 shows the effect of κ and $q_{\max}$ for determining the *a priori* speech absence

[3] All speech utterances used for experimentation are scaled/normalized, such that the maximum analysis frame energy (of a given utterance) is one.

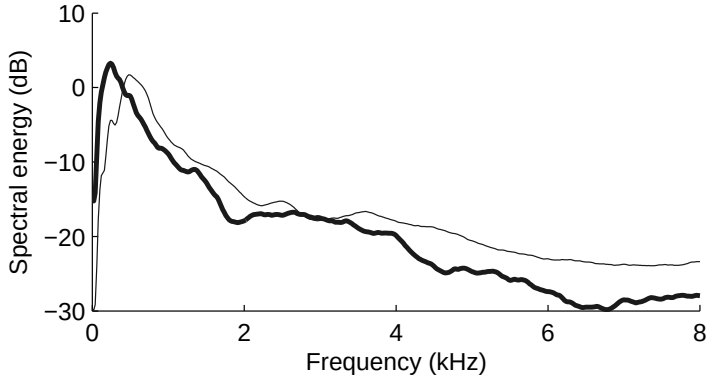

Figure 8.5: Ensemble average clean speech energy (in log-spectral energy domain). $E[[A(k)]^2]$ was generated off the vowel-consonant-vowel OLLO2 training dataset (dark line), and the RM training set (thin line).

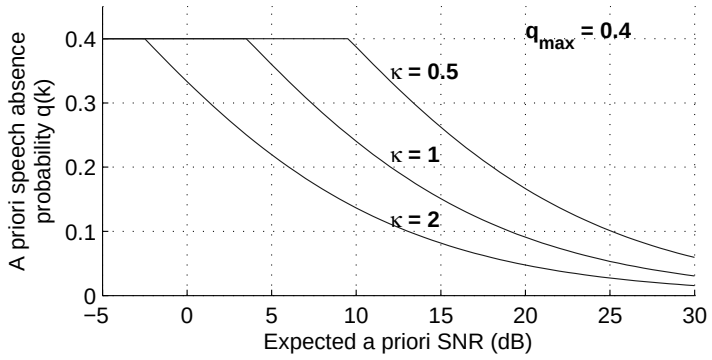

Figure 8.6: Effect of parameters κ and $q_{\max}$ on the determination of *a priori* speech absence probability $q(k)$. The expected *a priori* SNR axis is given by $E[[A(k)]^2]/\lambda_d(k)$, where $E[[A(k)]^2]$ is an ensemble spectral energy average generated from a clean speech corpus (see Fig. 8.5). Lower values of κ increases the aggressiveness of the SPU, assigning a higher probability of speech absence at a given SNR.

probability $q(k)$. In low SNR conditions, the value of $q(k)$ is large. This increases the aggressiveness of the SPU in order to compensate the bias of the SE estimator and prevent overestimation of ξ. At higher SNRs, the value of $q(k)$ decreases, eventually switching off SPU (at $q(k) = 0$) in clean conditions. This reflects the belief that in high SNR conditions there is negligible bias to compensate. The parameter κ

controls the rate at which the SPU is scaled. Small values of κ increase overall SPU aggressiveness and vice-versa. The parameter $q_{\max}$ sets the maximum allowed SPU strength.

In may be noted that in (8.27), we have used a spectral amplitude ratio (between the ensemble speech average and noise) rather than a spectral power/energy ratio. Our choice for this was motivated empirically, rather than mathematically. In our experiments, the amplitude ratio appeared to give a good balance between noise reduction and speech degradation across a wide range of SNRs.

8.5 Direct heuristic modification of the spectral energy estimator

In this section, we consider a direct heuristic modification of the MMSE SE estimator amplitude gain rule. The residual noise problem exhibited by the SE estimator can be corrected by forcing the SE estimator adopt a more aggressive spectral amplitude gain. To do this, we first examine the properties of the spectral amplitude gain family. It can be seen that SE, SA and LSA estimators all possess gain functions with similar properties (see Fig. 8.2). Mathematically, each of the above estimators can be described as the spectral Wiener gain (8.7) multiplied by a modification factor. For the SE estimator, the modification factor is given by

$$\begin{aligned} \zeta_{SE} &= G_{SE}/G_{SW} \\ &= \left(1 + [\nu(k)]^{-1}\right)^{0.5}. \end{aligned} \tag{8.29}$$

At high SNRs the modification factor has negligible effect, i.e. $\zeta_{SE}(\nu) \approx 1$ for $\nu \gg 1$. At lower SNRs, the modification factor is grows larger than 1, increasing the gain of the SE estimator (relative to the SW estimator gain). Similar behaviour is exhibited by both the SA and LSA estimators. However, the modification factor for these estimators differ in two key aspects. Firstly, their modification factors become significant (greater than 1.1) at lower SNRs. Secondly, the overall strength

Estimator	β	**Approximation MSE**
Wiener	0	0.00
LSA	0.37	0.039
SA (1'st power)	0.74	0.0081
SE (2'nd power)	1.0	0.00
0.5'th power	0.57	0.016
1.5'th power	0.88	0.0022
2.5'th power	1.10	0.0023

Table 8.1: Using the generalized heuristic estimator to approximate common spectral estimators.

of the modification (for a given SNR setting) is weaker. It is interesting to note that both aspects are extremely well correlated and can be reasonably described with a single variable. Though it is certainly possible to express this modification in a non-parametric manner, we found it to be well approximated with the analytic equation below:

$$G_{mod}(k) = \frac{\xi(k)}{1+\xi(k)} \times \left(1 + [\nu(k)]^{-[\beta]^{-1}}\right)^{0.5[\beta]^{1.19}}, \tag{8.30}$$

where β is a parameter chosen to influence the modification factor. Thus, to overcome the noise suppression issues of the standard SE estimator, we now need only choose an appropriate suppression modifier β. When $\beta = 1$, the modified gain rule is equivalent to the standard SE estimator. Decreasing the value of β below 1 will force the estimator to adopt a more aggressive stance, while setting $\beta = 0$ will yield the spectral Wiener filter. Surprisingly, the modified gain rule can reasonably approximate several other spectral amplitude estimators as well. Table 8.1 lists the values of β used to approximate several common spectral estimators, along with

approximation error. Approximation mean square error (MSE) is given as

$$\epsilon(\text{dB}) = E\left[\left(20\log_{10}\left(\frac{G_{true}(\xi,\gamma)}{G_{mod}(\xi,\gamma,\beta)}\right)\right)^2\right] \quad \text{for } 10^{-2} \leq \xi,\gamma \leq 10^3. \tag{8.31}$$

When the *a priori* SNR ξ was allowed to take values below 10^{-2}, the approximation did generally get worse. However, it is common practice to clamp ξ at a minimum value of between 10^{-2} and 10^{-3}[80], negating the need to approximate lower (*a priori*) SNR regions.

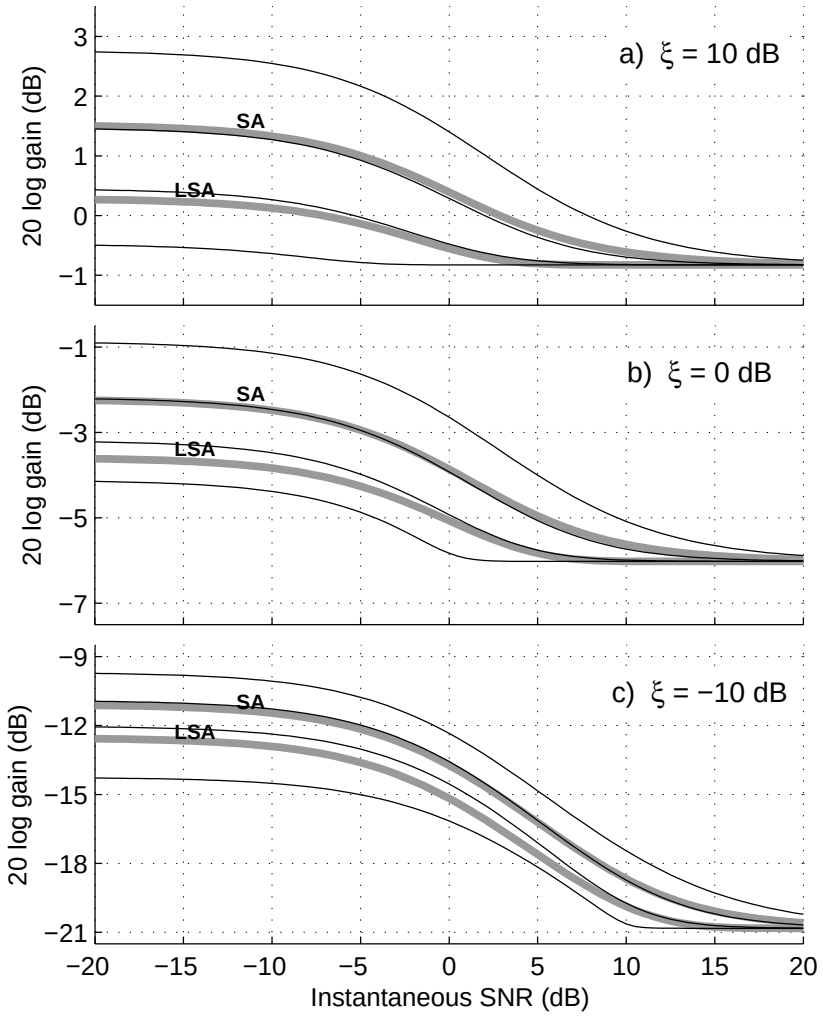

Figure 8.7: Spectral amplitude gains for the generalized heuristic spectral amplitude estimator. SA and LSA spectral amplitude gain functions are shown as thick grey lines. Heuristic-modified SE gains are shown as thin black lines: from lowest gain to highest $\beta = 0.05, 0.43, 0.74, 1.1$. Subplots: a) *a priori* SNR = 10 dB, b) *a priori* SNR = 0 dB, c) *a priori* SNR = -10 dB, d) *a priori* SNR = -20 dB.

8.6 Experimental results

8.6.1 Enhancement system description

For our experiments, we decompose speech utterances into overlapping frames. Each analysis frame is 25 ms in length, and overlaps the previous analysis frame by 15 ms. Each analysis frame has a Hamming window applied before being enhanced with a given regime. Enhanced frames are then synthesized into coherent utterance with the overlap-add method [29]. To derive the noise estimate $\lambda_d(m,k)$, we use a simple voice activity detector (VAD). An initial noise estimate is generated from the first 125 ms of each speech stimulus, and recursively updated. The recursive update is given as follows

$$\lambda_d(m,k) = \eta\lambda_d(m-1,k) + (1-\eta)[R(k)]^2, \tag{8.32}$$

where $\eta = 0.98$ in the case that a noise-only frame has been detected and $\eta = 1$ otherwise. The *a posteriori* SNR can be then be calculated via (8.13). To calculate the *a priori* SNR ξ, we use the decision-directed approach covered in sub-section 8.3.2. For the proposed estimators, the scaling factor κ must be empirically determined. To do this, we degraded several training utterances from each dataset with additive white Gaussian noise at 5 dB. The degraded sentences were enhanced with the proposed SE SPU estimator over several values of κ. The optimal parameter value was selected to minimize the word error rate (WER) using a clean speech HMM recognizer. For the following experiments, we set the SPU heuristic parameter $q_{\max}$ to 0.4.

8.6.2 Automatic speech recognition system description

To test ASR performance, we use a standard MFCC feature vector in conjunction with the HTK recognition framework [142]. Speech utterances are first decomposed into 25 ms long frames, each shifted by 10 ms. Frames then have a Hamming window applied before MFCCs are calculated. We accumulate 26 log-filterbank energies, and retain the first 12 cepstral coefficients (excluding the zeroth). In place

	Speech database		
Parameter	**RM**	**OLLO**	**Aurora2**
Acoustic model	3-state triphone HMMs, 8 Gaussian mixtures per state	3-state monophone HMMs, 32 Gaussian mixtures per state	16-state word HMMs, 3 Gaussian mixtures per state
Sampling freq. (kHz)	16	16	8
Frame length (ms)	25	25	25
Frame shift (ms)	10	10	10
Analysis window	Hamming	Hamming	Hamming
No. Mel filterbanks	26	26	26
Filterbank freq. range (kHz)	0–8	0–8	0–4
Cepstral coefficients	1 through 12	1 through 12	1 through 12
Cepstral lifter factor	22	22	22
Cepstral mean subtraction	Yes	No	Yes
Appended features	Log-frame energy, Δ's, Δ^2's	Log-frame energy, Δ's, Δ^2's	Log-frame energy, Δ's, Δ^2's
Δ window (frames)	± 2	± 2	± 3
Δ^2 window (frames)	± 2	± 2	± 5
Feature dimension	39	39	39

Table 8.2: Overview of the ASR parameters used for experimental analysis.

of the zeroth cepstral coefficient, the total log energy of each frame is used. Once this is done, we append delta and acceleration coefficients to give a 39 dimensional feature vector. For each of the estimators presented, we produce MFCCs from their respective time-domain enhanced signals. An overview of the ASR system used in our experiments is given in table 8.2. Training is provided by clean, unaltered utterances. We give results for the spectral Wiener (SW), MMSE log-spectral amplitude (LSA), MMSE spectral amplitude (SA), MMSE spectral energy (SE) estimator and heuristic SE estimator (HSE). For the SE estimator, we also give results for a static $q(k) = 0.3$ SPU (SE-0.3), a data-driven SPU (SE-DD) proposed in [82] and the proposed heuristic-driven SPU (SE-prop). For the HSE estimator, we use parameter set $\beta = 0.75$ (HSE-0.75) and $\beta = 0.4$ (HSE-0.4) designed to mimic the SA and LSA estimators respectively. We conduct experiments over three speech databases, each of which covers a different ASR topology.

- **Resource Management** [110]. Continuous triphone-based recognition, medium vocabulary with structured language model.
- **OLLO2** [136]. Single token, monophone-based recognition, medium vocabulary with no language model.
- **Aurora2 Digits** [104]. Continuous word-based recognition, small vocabulary with no language model.

8.6.3 Resource management word recognition

A speaker independent section of the DARPA resource management (RM) database is used for medium-vocabulary recognition [110]. The database was recorded in clean conditions (sample rate of 16 kHz) and has a vocabulary of approximately 1000 words. For training, there are 3990 sentences spoken by 109 speakers. For testing, we use the February '89 test set which has 300 sentences spoken by 10 different speakers. White, Volvo and babble noises are artificially added at several SNRs. For recognition, we train triphone-level HMMs, having three states with eight Gaussian mixtures each. Cepstral mean subtraction (CMS) is applied as a standard post-processor. For the proposed SE SPU estimator, a SPU scaling factor of $\kappa = 0.5$ was used. ASR word error rate (WER) scores are given in table 8.3.

8.6.4 OLLO2 logatome recognition

In this section we present results for a subset of the Oldenburg logatome database (OLLO) [136]. OLLO2 is unique from the RM and Aurora2 databases in several areas. Firstly, it is not a continuous recognition task, requiring only a single logatome per speech file to be recognized. Secondly, there is very little context available for recognition. Logatomes are nonsense words, and consist of every possible vowel-consonant pairing, making the acoustic model relatively sensitive to noise.

We use the vowel-consonant-vowel stimulus for recognition – abba, adda, egge etc, for a total of 70 logatomes. Each sound file has a single spoken logatome digitized

at 16 kHz and is recorded under one of several conditions (slow, fast, loud, quiet, questioning and normal). The testing and training datasets are matched in terms of regional dialect and recording conditions and consist of roughly 27,000 utterances each. We degrade the testing stimulus with (stationary) additive white and F16 noise at various SNRs. We train monophone HMMs, with 3 states per phoneme and 32 Gaussian mixtures per state. Because of the short-duration of the utterances, we do not apply cepstral mean subtraction. For the proposed SE SPU estimator, we set $\kappa = 0.5$. Word error rates are determined by treating the entire logatome as a word. ASR WER scores are given in table 8.4.

8.6.5 Aurora2 digit recognition

Aurora2 is a speaker independent database for connected digit recognition [104]. Unlike the RM database, Aurora2 lacks a language model, though its acoustic models are relatively sparse. Spoken digits in the database consist of zero through nine as well as 'oh', giving a vocabulary size of 11. Testing and training utterances were down-sampled to 8 kHz and filtered with G712 characteristics. Finally, utterances have had noise artificially added at several SNRs. Word-level HMMs are built, each with 16 states and 3 Gaussian mixtures per state. CMS was applied as a standard post-processor. For the proposed SE SPU estimator we used a scaling factor of $\kappa = 5$. ASR WER scores are given in tables 8.5 and 8.6 for recognition tasks A and B respectively.

8.6.6 Discussion

It can be seen that the standalone SE estimator performs quite well in clean and light noise environments. However its performance degrades substantially at lower SNR ranges, particularly for the RM and OLLO2 recognition tasks. The introduction of standard SPU did make the SE estimator considerably more robust, but like the SA and LSA estimators this appeared to come at the cost of speech degradation for higher SNRs. This was also true for the direct heuristic SE estimator. In

fact, the HSE estimator was able to closely mimic the performance of both the SA and LSA estimators (when using parameter $\beta = 0.75$ and $\beta = 0.4$ respectively). Unfortunately, this also means the HSE estimator suffered the same drawbacks. It should be noted however, that the HSE estimator is a simpler estimator – being easier to implement and more efficient to run than both the SA and LSA estimators.

A bigger improvement in recognition accuracy can be seen for the heuristic SE SPU. Here the aggressiveness of the SPU is scaled back at higher SNRs, limiting the amount of speech distortion. Overall, this gave the proposed SE SPU estimator superior performance compared with the other estimators. The use of a white noise development set to train the heuristic SPU worked well for most noise environments. Notable exceptions were the restaurant, airport and babble Aurora2 noise tasks, where the heuristic SPU (trained on white noise) was too aggressive. However this may also reflect the limitations of using VAD to track non-stationary noises.

The Aurora2 recognition task seemed especially sensitive to speech distortion. This was immediately apparent when determining the SPU scaling parameter κ. For the Aurora2 task, κ was found to be much higher than both the RM and OLLO2 tasks (5 versus 0.5). As a result, the SPU addition played a minor role for the Aurora2 task. Here, the SE and SE SPU estimators performed similarly, only showing significant divergence at the 0 dB noise level. Another interesting observation concerns the spectral Wiener filter, which performed poorly on both the RM and Aurora2 datasets. With a very aggressive suppression rule, ξ tended to be heavily under-estimated in the Wiener-based decision-directed framework. One exception for this was the OLLO2 0 dB white noise task, where the spectral Wiener estimator performed well within the decision-directed framework.

It is of interest to note the overall divergence of results between the Aurora2, OLLO2 and RM databases. The differing levels of optimal noise suppression / speech distortion trade-off suggest ASR architecture (phoneme vs. word based, language models used, continuous vs. static recognition etc) plays a large role in determining the effectiveness of front-end enhancement algorithms.

8.7 Conclusion

In this chapter we have investigated the use of the spectral energy estimator for use in robust ASR. Traditionally, the spectral energy estimator has suffered from the problem of residual noise. In order to improve the SE estimator for use in robust ASR, we identified the causes of the residual noise. These problems were then addressed with a simple, heuristic based SPU. Experimental results show a significant improvement in robustness, over both the baseline results and the more common log-spectral amplitude estimator. The improvement gained by the heuristic SPU is especially evident at lower SNRs, where the standalone SE estimator typically struggles.

In this chapter, we also proposed a direct heuristic modification to the SE estimator, allowing it to mimic other algorithms in the spectral estimator family. Using a static value for the suppression parameter β, the HSE estimator closely resembled the SA and LSA estimators – offering very similar performance for reduced computational cost.

		SNR (dB)					
Treatment		∞	**30**	**20**	**10**	**0**	**AVG**
White noise	None	4.30	5.48	11.89	47.13	95.89	17.20
	SA	4.26	5.04	7.43	27.02	79.55	10.94
	LSA	4.42	5.36	7.55	23.39	76.85	10.18
	SW	9.86	9.78	22.02	61.17	92.96	25.71
	SE	4.18	5.04	9.11	37.86	91.98	14.05
	SE-0.3	4.50	5.01	7.00	23.03	75.87	9.89
	SE-DD	4.26	4.93	7.35	25.58	82.09	10.53
	SE-prop	4.07	5.04	7.16	22.21	74.07	9.62
	HSE-0.4	4.34	5.40	7.55	22.80	75.75	10.02
	HSE-0.75	4.18	5.04	7.43	27.69	79.98	11.09
Babble noise	None	4.30	4.73	8.21	38.87	94.68	14.03
	SA	4.26	4.89	7.78	28.90	89.87	11.46
	LSA	4.42	5.08	8.56	32.58	90.61	12.66
	SW	9.86	13.96	28.55	73.52	96.44	31.47
	SE	4.18	4.73	8.06	31.17	91.51	12.04
	SE-0.3	4.50	4.93	7.74	30.00	90.30	11.79
	SE-DD	4.26	5.04	7.78	30.47	91.98	11.89
	SE-prop	4.07	4.89	7.67	27.06	90.22	10.92
	HSE-0.4	4.34	4.89	8.56	32.77	91.04	12.64
	HSE-0.75	4.18	4.93	8.10	28.82	90.30	11.51
Volvo noise	None	4.30	4.03	5.01	7.86	23.03	5.30
	SA	4.26	4.42	4.34	4.73	10.09	4.44
	LSA	4.42	4.61	4.46	4.93	8.76	4.61
	SW	9.86	8.21	7.12	8.64	18.15	8.46
	SE	4.18	4.18	4.22	6.22	16.54	4.70
	SE-0.3	4.50	4.69	4.42	4.93	8.96	4.64
	SE-DD	4.26	4.54	4.26	5.04	8.41	4.53
	SE-prop	4.07	4.26	4.07	4.50	6.84	4.23
	HSE-0.4	4.34	4.65	4.54	4.97	8.76	4.62
	HSE-0.75	4.18	4.42	4.38	4.73	10.56	4.43
Average	None	4.30	4.75	8.37	31.29	71.20	12.18
	SA	4.26	4.78	6.52	20.22	59.84	8.95
	LSA	4.42	5.02	6.86	20.30	58.74	9.15
	SW	9.86	10.65	19.23	47.78	69.18	21.88
	SE	4.18	4.65	7.13	25.08	66.68	10.26
	SE-0.3	4.50	4.88	6.39	19.32	58.38	8.77
	SE-DD	4.26	4.84	6.46	20.36	60.83	8.98
	SE-prop	4.07	4.73	6.30	17.92	57.04	8.26
	HSE-0.4	4.34	4.98	6.88	20.18	58.52	9.09
	HSE-0.75	4.18	4.80	6.64	20.41	60.28	9.01

Table 8.3: RM ASR word error rates. WER average is computed from 10 dB to ∞ dB SNR.

		SNR (dB)					
Treatment		∞	**30**	**20**	**10**	**0**	**AVG**
White noise	None	16.95	35.87	79.12	96.78	98.56	57.18
	SA	17.19	19.74	32.84	74.73	94.61	36.13
	LSA	17.84	19.30	26.42	58.57	92.46	30.53
	SW	19.82	23.36	34.00	49.89	71.83	31.77
	SE	17.03	23.56	56.47	89.07	97.86	46.53
	SE-0.3	17.55	19.15	28.75	67.89	93.60	33.34
	SE-DD	17.10	21.30	50.21	85.56	95.75	43.54
	SE-prop	17.01	19.42	24.24	41.39	88.96	25.52
	HSE-0.4	17.74	19.26	26.50	58.92	92.47	30.61
	HSE-0.75	17.13	19.96	34.34	76.44	94.87	36.97
F16 noise	None	16.95	19.96	29.37	65.25	94.37	32.88
	SA	17.19	18.03	21.79	33.86	63.84	22.72
	LSA	17.84	18.12	21.89	31.80	56.84	22.42
	SW	19.82	21.94	30.19	43.69	64.51	28.91
	SE	17.03	18.63	24.35	42.33	84.29	25.59
	SE-0.3	17.55	18.11	21.25	31.69	59.72	22.15
	SE-DD	17.10	18.39	23.10	37.36	74.46	23.99
	SE-prop	17.01	18.16	21.13	30.67	55.43	21.74
	HSE-0.4	17.74	18.16	22.00	32.00	58.17	22.48
	HSE-0.75	17.13	18.02	21.92	34.51	65.74	22.90
Average	None	16.95	27.92	54.25	81.02	96.47	45.03
	SA	17.19	18.89	27.32	54.3	79.23	29.43
	LSA	17.84	18.71	24.16	45.19	74.65	26.48
	SW	19.82	22.65	32.1	46.79	68.17	30.34
	SE	17.03	21.1	40.41	65.7	91.08	36.06
	SE-0.3	17.55	18.63	25	49.79	76.66	27.75
	SE-DD	17.1	19.85	36.66	61.46	85.11	33.77
	SE-prop	17.01	18.79	22.69	36.03	72.2	23.63
	HSE-0.4	17.74	18.71	24.25	45.46	75.32	26.54
	HSE-0.75	17.13	18.99	28.13	55.48	80.31	29.93

Table 8.4: OLLO2 ASR word error rates. WER average is computed from 10 dB to ∞ dB SNR.

	Treatment	SNR (dB) clean	20	15	10	5	0	AVG
Subway noise	None	0.68	2.95	5.59	12.34	30.49	70.56	24.39
	SA	0.86	3.16	4.54	8.84	17.13	39.02	14.54
	LSA	0.95	3.56	5.43	10.07	19.22	40.28	15.71
	SW	17.50	31.56	38.75	47.80	59.66	78.85	51.32
	SE-0.3	0.92	3.19	5.07	9.61	18.11	39.73	15.14
	SE	0.74	3.01	4.94	9.58	19.62	45.93	16.62
	SE-DD	0.80	3.01	4.82	9.03	18.36	41.23	15.29
	SE-prop	0.74	3.07	4.61	8.63	17.50	39.88	14.74
	HSE-0.4	0.89	3.56	5.31	9.89	19.19	40.62	15.71
	HSE-0.75	0.83	3.10	4.51	8.72	16.98	38.90	14.44
Babble noise	None	0.68	2.03	4.63	16.51	53.02	90.84	33.41
	SA	0.86	2.90	6.53	15.15	34.58	66.05	25.04
	LSA	0.95	5.65	10.46	19.68	37.79	66.32	27.98
	SW	17.50	39.60	46.58	58.43	74.61	87.30	61.30
	SE-0.3	0.92	4.72	9.37	19.07	38.66	66.96	27.76
	SE	0.74	1.90	4.08	10.97	30.77	67.17	22.98
	SE-DD	0.80	2.48	6.20	15.18	34.95	66.72	25.11
	SE-prop	0.74	2.06	4.47	13.03	33.86	65.99	23.88
	HSE-0.4	0.89	6.08	11.34	21.40	40.24	67.84	29.38
	HSE-0.75	0.83	2.87	6.23	14.90	34.37	65.84	24.84
Car noise	None	0.68	2.09	4.38	11.39	34.39	81.12	26.67
	SA	0.86	1.55	2.71	5.22	13.39	33.61	11.30
	LSA	0.95	1.55	2.98	5.79	15.27	37.55	12.63
	SW	17.50	25.14	32.09	41.87	60.93	83.54	48.71
	SE-0.3	0.92	1.46	2.80	5.25	13.27	32.99	11.15
	SE	0.74	1.70	3.04	6.59	17.54	46.47	15.07
	SE-DD	0.80	1.64	2.80	5.79	14.58	36.12	12.19
	SE-prop	0.74	1.76	2.83	6.29	14.14	34.83	11.97
	HSE-0.4	0.89	1.58	3.16	6.08	15.21	36.89	12.58
	HSE-0.75	0.83	1.49	2.65	5.25	13.84	34.72	11.59
Exhibition noise	None	0.68	3.73	6.97	15.21	38.88	82.14	29.39
	SA	0.86	3.52	5.86	12.77	27.61	51.31	20.21
	LSA	0.95	5.46	8.58	16.94	31.72	52.98	23.14
	SW	17.50	40.39	48.63	61.28	75.87	86.27	62.49
	SE-0.3	0.92	4.23	6.97	14.84	29.93	52.64	21.72
	SE	0.74	2.99	5.55	11.82	25.55	53.93	19.97
	SE-DD	0.80	3.33	5.99	13.17	27.74	52.98	20.64
	SE-prop	0.74	2.93	5.58	11.63	26.26	52.98	19.88
	HSE-0.4	0.89	5.46	8.45	16.97	32.24	54.06	23.44
	HSE-0.75	0.83	3.30	5.71	12.71	27.00	51.10	19.96
Set A averages	None	0.68	2.70	5.39	13.86	39.20	81.17	28.47
	SA	0.86	2.78	4.91	10.50	23.18	47.50	17.77
	SE-0.3	0.92	3.40	6.05	12.19	24.99	48.08	18.94
	SE	0.74	2.40	4.40	9.74	23.37	53.38	18.66
	SE-DD	0.80	2.62	4.95	10.79	23.91	49.26	18.31
	SE-prop	0.74	2.46	4.37	9.90	22.94	48.42	17.62
	HSE-0.4	0.89	4.17	7.07	13.59	26.72	49.85	20.28
	HSE-0.75	0.83	2.69	4.78	10.40	23.05	47.64	17.71
	LSA	0.95	4.06	6.86	13.12	26.00	49.28	19.87
	SW	17.50	34.17	41.51	52.35	67.77	83.99	55.96

Table 8.5: Aurora2A ASR word error rates. WER average is computed from 0 dB to 20 dB SNR.

		SNR (dB)						
Treatment		**clean**	**20**	**15**	**10**	**5**	**0**	**AVG**
Restaurant noise	None	0.68	2.33	4.70	16.18	45.96	78.97	29.63
	SA	0.86	4.67	9.67	21.03	39.39	65.15	27.98
	LSA	0.95	8.29	14.09	25.61	42.89	67.15	31.61
	SW	17.50	37.80	46.85	57.48	72.89	86.83	60.37
	SE-0.3	0.92	6.60	12.40	25.05	42.00	66.10	30.43
	SE	0.74	2.58	5.74	13.66	34.88	65.00	24.37
	SE-DD	0.80	3.75	8.17	19.90	39.85	65.31	27.40
	SE-prop	0.74	2.73	5.93	15.51	37.40	65.61	25.44
	HSE-0.4	0.89	8.69	14.89	26.93	43.94	68.19	32.53
	HSE-0.75	0.83	4.45	9.30	20.57	39.18	65.15	27.73
Street noise	None	0.68	2.60	5.11	13.45	37.64	75.09	26.78
	SA	0.86	3.20	4.99	9.49	21.67	46.46	17.16
	LSA	0.95	3.54	6.35	11.25	25.27	49.24	19.13
	SW	17.50	32.83	40.08	51.57	67.68	85.07	55.45
	SE-0.3	0.92	3.36	5.50	10.13	23.49	47.25	17.95
	SE	0.74	2.42	4.14	8.77	23.52	53.17	18.40
	SE-DD	0.80	3.30	5.05	9.22	23.19	49.46	18.04
	SE-prop	0.74	2.51	4.47	8.65	22.22	48.00	17.17
	HSE-0.4	0.89	3.84	6.47	11.31	25.79	49.88	19.46
	HSE-0.75	0.83	3.11	4.93	9.34	21.19	46.40	16.99
Airport noise	None	0.68	1.79	4.09	11.72	37.16	73.90	25.73
	SA	0.86	2.42	5.16	12.29	27.35	51.33	19.71
	LSA	0.95	4.15	7.07	14.97	30.21	53.21	21.92
	SW	17.50	33.97	39.43	50.91	68.83	84.40	55.51
	SE-0.3	0.92	3.49	6.38	13.93	30.12	52.64	21.31
	SE	0.74	1.76	3.43	9.25	24.63	54.04	18.62
	SE-DD	0.80	2.51	4.92	12.38	28.30	53.03	20.23
	SE-prop	0.74	1.76	3.55	10.35	26.36	50.88	18.58
	HSE-0.4	0.89	4.18	7.46	15.81	31.85	54.73	22.81
	HSE-0.75	0.83	2.30	5.04	11.99	27.05	50.76	19.43
Train noise	None	0.68	1.54	4.32	11.66	34.74	80.31	26.51
	SA	0.86	2.72	4.75	7.62	20.02	40.45	15.11
	LSA	0.95	3.58	5.52	9.84	21.69	42.18	16.56
	SW	17.50	28.36	34.37	45.97	64.79	84.94	51.69
	SE-0.3	0.92	3.24	5.09	8.45	20.46	40.08	15.46
	SE	0.74	1.82	3.86	7.25	20.80	49.00	16.55
	SE-DD	0.80	2.53	4.35	7.71	20.58	42.39	15.51
	SE-prop	0.74	2.04	4.04	7.22	19.78	41.53	14.92
	HSE-0.4	0.89	3.67	5.68	10.09	22.15	42.55	16.83
	HSE-0.75	0.83	2.62	4.60	7.62	20.24	40.48	15.11
Set B averages	None	0.68	2.07	4.56	13.25	38.88	77.07	27.16
	SA	0.86	3.25	6.14	12.61	27.11	50.85	19.99
	SE-0.3	0.92	4.17	7.34	14.39	29.02	51.52	21.29
	SE	0.74	2.15	4.29	9.73	25.96	55.30	19.49
	SE-DD	0.80	3.02	5.62	12.30	27.98	52.55	20.30
	SE-prop	0.74	2.26	4.50	10.43	26.44	51.51	19.03
	HSE-0.4	0.89	5.10	8.63	16.04	30.93	53.84	22.91
	HSE-0.75	0.83	3.12	5.97	12.38	26.92	50.70	19.82
	LSA	0.95	4.89	8.26	15.42	30.02	52.95	22.31
	SW	17.50	33.24	40.18	51.48	68.55	85.31	55.76

Table 8.6: Aurora2B ASR word error rates. WER average is computed from 0 dB to 20 dB SNR.

Chapter 9

MMSE estimation of Mel-frequency cepstral coefficients

9.1 Introduction

In Chapter 8, we proposed heuristic modifications to the spectral energy estimator (SE) to improve its performance for front-end ASR feature enhancement. One of the shortcomings identified with the SE estimator was the presence of bias when estimating log-filterbank energies. Two heuristic adaptations were then proposed to remedy this bias. In this chapter, we take a mathematically thorough approach for estimating the Mel-frequency cepstral coefficient (MFCC) vector. In particular we derive the optimal estimation of a clean speech, MFCC vector $\boldsymbol{c}_x$, from speech corrupted with additive noise. Under the additive noise assumption, $y(n) = x(n) + d(n)$, where $y(n), x(n)$ and $d(n)$ are the noisy speech, clean speech and noise signals respectively.

In the literature, several approaches for increasing additive noise robustness have been proposed. Most fall under the following general categories: robust feature selection and extraction [30, 66], speech enhancement [42, 50, 57, 67, 77], model

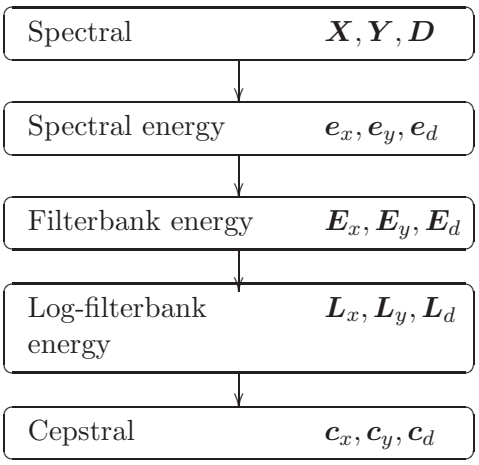

Figure 9.1: The hierarchy of variables required for calculation of the MFCC vector $\boldsymbol{c}_y$ given spectral vector $\boldsymbol{Y}$. Corresponding hierarchies exist for the noise: $\boldsymbol{D}$ through $\boldsymbol{c}_d$ and clean speech $\boldsymbol{X}$ through $\boldsymbol{c}_x$.

adaptation [3, 53], model-based feature enhancement [88, 130], missing feature theory [12, 27, 116] and multistyle training [31].

In this chapter we again investigate a front-end feature enhancement method; specifically, the optimal estimation of the clean speech MFCC vector $\boldsymbol{c}_x$. There are many ways in which to produce such an estimate. One of the most common takes the form of a stochastic, unbiased minimum mean square error (MMSE) estimator. Here, the clean cepstrum estimate $\hat{\boldsymbol{c}}_x$ is given as

$$\begin{aligned} \hat{\boldsymbol{c}}_x &= E\left[\boldsymbol{c}_x|\boldsymbol{Y}\right] \\ &= \int_{-\infty}^{\infty} \boldsymbol{c}_x p(\boldsymbol{c}_x|\boldsymbol{Y}) d\boldsymbol{c}_x, \end{aligned} \tag{9.1}$$

where $E[.]$ is the expectation operator and $\boldsymbol{Y}$ is the corresponding frame of observed (noisy) spectral domain speech. While the MMSE framework gives a concise mathematical formulation of the estimation problem, it is very sensitive to the statistical assumptions that underpin it. For cepstrum estimation, this is especially important since an additive noise relationship in the time domain is highly non-linear in the cepstral domain. Given a noisy speech spectral vector $\boldsymbol{Y}$, several intermediate

variables are calculated prior to the MFCC vector $\boldsymbol{c}_y$ [69]. Fig. 9.1 highlights the hierarchy of intermediate variables required for a cepstral estimate.

From (9.1), the primary goal of the MMSE estimation framework is the development of an accurate and analytically tractable form for posterior probability density function (PDF) $p(\boldsymbol{c}_x|\boldsymbol{Y})$. Given the extensive list of intermediate steps linking the spectral and cepstral domains, the direct derivation of this PDF may not seem particularly appealing. Because of this, a number of cepstral estimators have bypassed the explicit derivation of $p(\boldsymbol{c}_x|\boldsymbol{Y})$ by directly modelling the relationships between higher tier variables. In particular, many seek to directly describe mathematical relationships between clean and corrupted log-filterbank energies and/or cepstral coefficients. In cepstral domain, the posterior PDF can be given as

$$\begin{aligned} p(\boldsymbol{c}_x|\boldsymbol{Y}) &\approx p(\boldsymbol{c}_x|\boldsymbol{c}_y) \\ &\approx \frac{p(\boldsymbol{c}_x)p(\boldsymbol{c}_y|\boldsymbol{c}_x)}{p(\boldsymbol{c}_y)}. \end{aligned} \tag{9.2}$$

Apart from bypassing the lower tier variables, estimators of this type offer another appealing property – the inclusion of a cepstral level prior PDF $p(\boldsymbol{c}_x)$. This allows a clean speech model to be described in terms of the more meaningful cepstrum domain. Despite this, estimators of this form have several drawbacks:

- Conditioning upon $\boldsymbol{c}_y$ is not optimal. Since $\boldsymbol{c}_y$ is produced from a (non-invertible) filterbank transformation, it is only a subset of observational evidence $\boldsymbol{Y}$.
- The additive noise relationship (in time/spectral domain) is highly non-linear in the log-filterbank and cepstral domains.
- If a closed-form solution of the estimate is required, the form of both $p(\boldsymbol{c}_y|\boldsymbol{c}_x)$ and $p(\boldsymbol{c}_x)$ must be chosen to facilitate simple integration in (9.1) .

Several approaches have been proposed in past literature to address these

problems. One approach involves direct linearization of a log-filterbank/cepstral domain noise model with vector Taylor series (VTS) expansion [87]. VTS linearizes a noise model that assumes filterbank energies are additive; i.e., $\boldsymbol{E}_y = \boldsymbol{E}_x + \boldsymbol{E}_d$, where $\boldsymbol{E}_y$,$\boldsymbol{E}_x$ and $\boldsymbol{E}_d$ are the noisy speech, clean speech and noise filterbank energies, respectively. VTS uses this linearized model in conjunction with an *a priori* speech model to enhance speech features. For the *a priori* speech model, a Gaussian mixture model (GMM) trained on a clean speech corpus is generally used. With a sufficiently large number of mixtures, VTS has been shown to give good recognition performance for several speech tasks [87, 130]. To get the best performance with VTS, full covariances must be used to model the log-filterbank noise distortion model. This can occur explicitly if a full covariance log-filterbank energy GMM is used directly; or implicitly, if a diagonal covariance cepstral GMM is used. Because of these issues, VTS often has a high computational cost. Another concern with VTS is that the linearization itself is lossy. This problem can be exacerbated by operating the linearization over large ranges and / or choosing poor Taylor expansion points.

In [44], a more direct log-filterbank energy estimator was proposed. To do this, several assumptions were made. Firstly, spectral values belonging to a given filterbank were assumed to have identical energies. This assumption comes with the added requirements of rectangular filterbanks. Secondly, filterbank energies were transformed to give noise-normalized filterbank energies. This allowed modelling of the proposed (normalized filterbank) feature with a non-central chi-square PDF. Despite the above assumptions, a closed form solution to the estimator was not given. Instead, the final estimation was given as an interpolation of pre-computed, numerically integrated solutions. Because of these limitations, estimators of this form are unsuited for estimation of the MFCC feature set.

In [143], a different set of statistical assumptions were made. Here, filterbank energies were assumed to follow a chi distribution with two degrees of freedom; or alternatively, a Rayleigh distribution. This allows the use of the log-spectral amplitude estimator framework [40] to derive log-filterbank energies. However,

there does not appear to be any physical evidence to justify modelling filterbank energies with the Rayleigh distribution. In the literature, it has been common to model individual Fourier expansion coefficients with two degrees of freedom – typically with complex zero-mean Gaussian distributions [39][80]. Since filterbanks contain many such coefficients, it might be expected for them to exhibit more than two degrees of freedom. Another by-product of using the log-spectral amplitude estimator framework is the lack of correlation modelling between filterbanks. Since typical Mel-filterbanks are heavily overlapped and share a considerable amount of information, this is a clear deficiency.

In [70], filterbanks were described with a more general gamma distribution. Here, clean speech and noise filterbank energies were allowed to have independent gamma PDF shape parameters, but restricted to have identical gamma PDF scale parameters. Again, this may be a suboptimal assumption. Since clean speech and noise filterbanks ($\boldsymbol{E}_x$ and $\boldsymbol{E}_n$ respectively) have the same number of spectral energies summed into them, we might expect them to have similar shape parameters. However, depending on the relative energy of the speech and noise, we might expect them to have very different scale parameters. Again correlations between filterbank energies are implicitly ignored by modelling with diagonal filterbank covariances. In addition, the relationship between the clean speech and noisy filterbank energies was assumed to be governed by a Wiener-like filter response – an assumption that conflicts with the earlier filterbank gamma PDF assumption.

Each of the aforementioned methods is suboptimal in some manner, typically offering a trade-off between noise model simplification and computational tractability. In this work, we attempt to overcome this limitation. Specifically, we have two objectives: 1) the development of a mathematically robust cepstral noise distortion model and 2) a computationally tractable method for calculating the cepstral estimates. In this chapter, we avoid the problems of making statistical assumptions by investigating the direct estimation $E[\boldsymbol{c}_x|\boldsymbol{Y}]$. One advantage of this approach is that we may start with a very well known and statistically

sound framework – namely the framework used to derive the spectral Wiener and MMSE spectral amplitude estimators. This allows for thorough estimation of log-filterbank energies and the correlations that exist between them, as well as the subsequent MFCC feature vector. Since the low level framework models noise distortion between spectral domain variables, it must be extended before it may be used as a cepstral estimator. In this chapter, we detail the mathematical transformations required for converting the spectral models into filterbank, log-filterbank and cepstral models. This allows us to define the cepstral PDF $p(\boldsymbol{c}_x|\boldsymbol{Y})$. With this PDF, a closed-form estimate of the clean speech MFCC vector $\hat{\boldsymbol{c}}_x$ is then derived. We evaluate the proposed estimator on the RM and Aurora2 speech databases, and show improvements over the log-spectral amplitude estimator, log-filterbank energy estimator [143] and VTS estimator [87].

The rest of this chapter is organized as follows. In Section 9.2 we provide a brief review of the short-time spectral amplitude estimation framework. This framework can be used derive the posterior PDF $p(\boldsymbol{e}_x|\boldsymbol{Y})$ for spectral energies. In Section 9.3, we extend the spectral framework to develop models for filterbank $p(\boldsymbol{E}_x|\boldsymbol{Y})$ and log-filterbank $p(\boldsymbol{L}_x|\boldsymbol{Y})$ energies. These models are then extended again into cepstral domain models $p(\boldsymbol{c}_x|\boldsymbol{Y})$ in Section 9.4. In Section 9.5, we evaluate the proposed cepstral estimator on the Aurora2 and RM recognition tasks. Lastly, in Section 9.6 we conclude the chapter.

9.2 Statistical framework for MMSE short-time spectral estimation

The discrete short-time Fourier transform (DSTFT) of corrupted speech signal $y(n)$ is given by

$$Y(m,k) = \sum_{n=-\infty}^{\infty} y(n)w(mS-n)\exp(-j2\pi kn/K), \tag{9.3}$$

where k denotes the k'th discrete frequency of K uniformly spaced frequencies, $w(n)$ is an analysis window function, m is the short-time frame index and S is the

analysis frame shift (in samples). Under the additive noise assumption made earlier, the corrupted speech DSTFT spectrum may also be represented as

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{D}, \tag{9.4}$$

or for individual DSTFT coefficients[1]

$$Y(k) = X(k) + D(k), \tag{9.5}$$

where $X(k)$ and $D(k)$ are the DSTFT expansion coefficients for the k'th discrete frequency bin of the clean speech signal and noise signals, respectively.

Under the statistical framework developed by Ephraim and Malah [39, 40], individual DSTFT expansion coefficients $X(k)$ and $D(k)$ are assumed to be independent complex zero-mean Gaussian random variables (RVs), with expected energy $\lambda_x(k) = E\left[|X(k)|^2\right]$ and $\lambda_d(k) = E\left[|D(k)|^2\right]$. Detailed justification of this statistical assumption may be found in [39, 80]. Using these statistical assumptions, several spectral estimators have been developed. This includes the spectral Wiener (SW) [80], MMSE spectral amplitude (SA) [39] and MMSE log-spectral amplitude (LSA) [40] filters. While each of the aforementioned spectral estimators is optimal in some sense, the goal of ASR based speech enhancement is the accurate estimation of clean speech MFCC vector $\boldsymbol{c}_x$. However, in order to develop a MMSE MFCC estimator, we must first derive the posterior PDF $p(\boldsymbol{c}_x|\boldsymbol{Y})$ required by (9.1). In this section, derivations are given for the posterior spectral energy PDF $p(\boldsymbol{e}_x|\boldsymbol{Y})$. Later, in sections 9.3 and 9.4 we extend this framework by deriving the posterior filterbank PDF $p(\boldsymbol{E}_x|\boldsymbol{Y})$ and posterior cepstral PDF $p(\boldsymbol{c}_x|\boldsymbol{Y})$ respectively.

9.2.1 Spectral energy estimation

To develop the posterior spectral energy PDF $p(\boldsymbol{e}_x|\boldsymbol{Y})$, we may use the statistical framework described previously. Under this framework, the posterior PDF of

[1] For notational convenience, we have dropped the frame index m and dependence on this subscript is implicitly assumed unless stated otherwise.

individual spectral amplitudes, $p(A(k)|\boldsymbol{Y})$ can be given by the following Rice distribution [39]

$$\begin{aligned} p(A(k)|\boldsymbol{Y}) &= p(A(k)|Y(k)) \\ &= \frac{A(k)\exp\left(\frac{-[A(k)]^2}{\lambda(k)}\right) I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}A(k)\right)}{\int_0^\infty \tau\exp\left(\frac{-\tau^2}{\lambda(k)}\right) I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}\tau\right) d\tau}, \end{aligned} \tag{9.6}$$

where $A(k) = |X(k)|$ is the clean speech spectral amplitude, $I_0(.)$ is the zeroth order modified Bessel function, and

$$\lambda(k) = \frac{\lambda_x(k)\lambda_d(k)}{\lambda_x(k) + \lambda_d(k)}, \tag{9.7}$$

$$\nu(k) = \frac{\xi(k)}{1+\xi(k)}\gamma(k), \tag{9.8}$$

$$\xi(k) = \frac{\lambda_x(k)}{\lambda_d(k)}, \tag{9.9}$$

$$\gamma(k) = \frac{e_y(k)}{\lambda_d(k)}, \tag{9.10}$$

where ξ and γ are interpreted as the *a priori* signal to noise ratio (SNR) and *a posteriori* SNR respectively and $e_y(k) = |Y(k)|^2$ is the observed noisy speech spectral energy. Using (9.6), the posterior spectral energy PDF can be given as[2]

$$p(e_x(k)|\boldsymbol{Y}) = \frac{\exp\left(\frac{-e_x(k)}{\lambda(k)}\right)\ I_0\left(2\sqrt{\frac{\nu(k)e_x(k)}{\lambda(k)}}\right)}{\lambda(k)\exp\big(\nu(k)\big)}, \tag{9.11}$$

where clean speech spectral energy $e_x(k) = |X(k)|^2$. In order to develop future filterbank models, it is useful to derive the conditioned spectral energy mean and variance. Using (9.11), the conditioned expectation of individual spectral energies

[2]Further detail may be found in appendix A.

can be given as[3] [61]

$$\begin{aligned} E\left[e_x(k)|\boldsymbol{Y}\right] &= \hat{e}_x(k) \\ &= \lambda(k)[1+\nu(k)] \\ &= \left(\frac{\xi(k)}{1+\xi(k)}\right)^2 \left(1+\frac{\xi(k)}{\xi(k)\gamma(k)}\right) e_y(k). \end{aligned} \tag{9.12}$$

We may also solve for the conditioned spectral energy variance. Diagonal covariance terms are given by

$$\begin{aligned} E\left[[e_x(k)-\hat{e}_x(k)]^2|\boldsymbol{Y}\right] &= \Sigma_{e_x}(k,k) \\ &= [\lambda(k)]^2\,[1+2\nu(k)] \\ &= [\hat{e}_x(k)]^2 - \left(\frac{\xi(k)}{1+\xi(k)}\right)^4 [e_y(k)]^2. \end{aligned} \tag{9.13}$$

Since individual Fourier expansion coefficients are assumed to be independent, off-diagonal spectral energy covariances will be zero, i.e.,

$$\Sigma_{e_x}(k,k') = 0, \quad \text{for } k \neq k'. \tag{9.14}$$

9.2.2 Estimation of *a priori* SNR ξ

As explained in the previous chapter (Section 8.3.2), special care needs to be taken when estimating the SNR parameter ξ. In order to prevent its overestimation, we may utilize the speech presence uncertainty framework [85]. Under SPU, an *a posteriori* probability of speech presence $\varphi(k)$ can be given by [39]

$$\varphi(k) = \frac{\Lambda_k}{1+\Lambda_k}, \tag{9.15}$$

where Λ_k is the generalized speech presence ratio

$$\Lambda_k = \frac{1-q(k)}{q(k)}.\frac{\exp\big(\nu(k)\big)}{1+\xi(k)}. \tag{9.16}$$

[3] Further detail may be found in appendix B.

The term $q(k)$ is the *a priori* probability of speech absence and is regarded as a tuneable parameter. A number of methods exist for determining $q(k)$. For the SA estimator, it is common to use a static value of 0.3 [80]. Subsequent proposals geared mostly toward the LSA estimator are recursive, data driven procedures [26, 82, 124]. Given the *a posteriori* probability of speech presence $\varphi(k)$, SPU updated estimates for spectral energies can be given as

$$\begin{aligned}\hat{e}'_x(k) &= \varphi(k)\hat{e}_x(k) \\ &= \varphi(k)\left(\frac{\xi(k)}{1+\xi(k)}\right)^2\left(1+\frac{1}{\nu(k)}\right)e_y(k).\end{aligned} \tag{9.17}$$

When $\varphi(k)$ is small, it is a good indication that the *a priori* SNR has been overestimated. We use the SPU modified estimate $\hat{e}'_x(k)$ to derive a more appropriate value for the *a priori* SNR. Using $\hat{e}'_x(k)$ as the desired estimate, the spectral energy estimator (9.12) can be rearranged to give

$$\xi'(k) = -\left(\frac{2e_y(k)}{\lambda_d(k) - \sqrt{[\lambda_d(k)]^2 + 4e_y(k)\hat{e}'_x(k)}} + 1\right)^{-1}. \tag{9.18}$$

When speech is surely present ($\varphi(k) = 1$), it can be shown that $\hat{e}'_x(k) = \hat{e}_x(k)$, and thus $\xi'(k) = \xi(k)$. As the value of $\varphi(k)$ decreases, the value of $\xi'(k)$ is reduced to compensate. Using the updated $\xi'(k)$, the SPU updated spectral energy variance can be given as

$$\Sigma_{e'_x}(k,k) = [\hat{e}'_x(k)]^2 - \left(\frac{\xi'(k)}{1+\xi'(k)}\right)^4 [e_y(k)]^2. \tag{9.19}$$

For further detail on the spectral estimation framework and SPU, the reader is referred to [80].

9.3 Models for filterbank and log-filterbank variables

In this section we develop models for filterbank and log-filterbank variables. Since filterbank energies are linearly related to spectral energies, we may estimate the conditioned filterbank mean vector $\hat{\boldsymbol{E}}_x \in \mathbb{R}^{Q\times 1}$, and covariance matrix $\boldsymbol{\Sigma}_{E_x} \in \mathbb{R}^{Q\times Q}$, as follows

$$\begin{aligned} E[\boldsymbol{E}_x|\boldsymbol{Y}] &= \hat{\boldsymbol{E}}_x \\ &= \boldsymbol{H}\hat{\boldsymbol{e}}_x, \end{aligned} \tag{9.20}$$

$$\begin{aligned} COV[\boldsymbol{E}_x|\boldsymbol{Y}] &= \boldsymbol{\Sigma}_{E_x} \\ &= \boldsymbol{H}\boldsymbol{\Sigma}_{e_x}\boldsymbol{H}^T, \end{aligned} \tag{9.21}$$

where Q is the total number of filterbanks, $COV[.]$ is the covariance operator, and $\boldsymbol{H} \in \mathbb{R}^{Q\times K}$ is a filterbank gain matrix, such that $H(q,k)$ is the gain for the k'th frequency bin and q'th filterbank. For a typical (overlapping) Mel-filterbank, it is likely there will be positive covariances among adjacent filterbanks. This makes the covariance $\boldsymbol{\Sigma}_{E_x}$ a sparse, but non-diagonal matrix.

To determine the actual structure of the PDF $p(E_x(q)|\boldsymbol{Y})$, we would ideally convolve individual, filterbank scaled spectral energy PDFs (9.11) together. Unfortunately such a method leads to a complicated closed form solution. However, the resulting PDF does appear to be well approximated by the gamma distribution. Thus, given estimates for the filterbank mean and variance, the filterbank variable $E_x(q)$ can be described by the following gamma distribution

$$p(E_x(q)|\boldsymbol{Y}) = \frac{[E_x(q)]^{\alpha_q - 1}\, \exp\left(-\frac{E_x(q)}{\beta_q}\right)}{\beta_q^{\alpha_q}\Gamma(\alpha_q)}, \tag{9.22}$$

where $\Gamma(.)$ is the gamma function. The shape parameter α_q, and scale parameter β_q can be found using the method of moments:

$$\alpha_q = \frac{[\hat{E}_x(q)]^2}{\Sigma_{E_x}(q,q)}, \tag{9.23}$$

$$\beta_q = \frac{\Sigma_{E_x}(q,q)}{\hat{E}_x(q)}. \tag{9.24}$$

Further detail of the gamma PDF approximation is given in sub-section 9.3.1. It should be noted that as a shape parameter α_q is invariant Using (9.22), we may also define the posterior PDF for the log-filterbank variable[2] $L_x(q) = \log E_x(q)$

$$p(L_x(q)|\boldsymbol{Y}) = \frac{\exp\big(\alpha_q \left[L_x(q) - \log \beta_q\right] - \exp\left[L_x(q) - \log \beta_q\right]\big)}{\Gamma(\alpha_q)}. \tag{9.25}$$

The MMSE log-filterbank energy $\hat{L_x}(q)$, and log-filterbank variance $\Sigma_{L_x}(q,q)$, are given as[3] [61]

$$\begin{aligned} E\left[\log E_x(q)|\boldsymbol{Y}\right] &= \hat{L_x}(q) \\ &= \log \hat{E}_x(q) - \log(\alpha_q) + \Psi_0(\alpha_q), \end{aligned} \tag{9.26}$$

$$\begin{aligned} E\left[[\log E_x(q) - \hat{L}(q)]^2|\boldsymbol{Y}\right] &= \Sigma_{L_x}(q,q) \\ &= \Psi_1(\alpha_q), \end{aligned} \tag{9.27}$$

where $\Psi_0(.)$ is the digamma function and $\Psi_1(.)$ is the trigamma function. We may use an efficient series expansion of the digamma and trigamma functions [125] for calculating the log-filterbank mean and variance. The MMSE log-filterbank energy can be estimated as

$$\hat{L}_x(q) \approx \log \hat{E}_x(q) - \frac{0.500}{(\alpha_q + 0.045)} - \frac{0.108}{(\alpha_q + 0.045)^2} \quad \alpha_q \geq 1. \tag{9.28}$$

Similarly, variance terms are well approximated as

$$\Sigma_{L_x}(q,q) \approx \frac{1}{(\alpha_q - 0.1157)} + \frac{0.4040}{(\alpha_q - 0.1157)^2} \quad \alpha_q \geq 1. \tag{9.29}$$

[2]Further detail may be found in appendix A.
[3]Further detail may be found in appendix B.

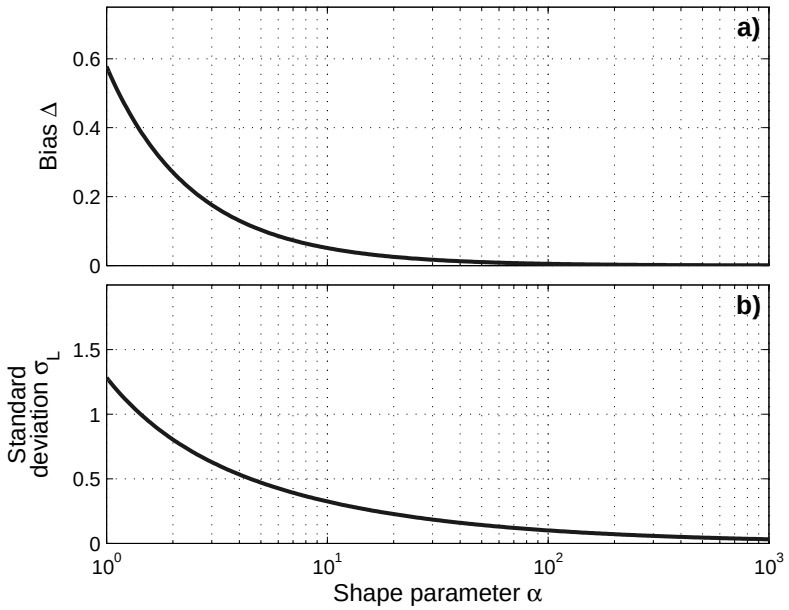

Figure 9.2: Effect of shape parameter α on the log-filterbank energy estimates. Subplots: a) difference term (9.31) b) the standard deviation of the log-filterbank variable.

It can be shown that the maximum *a posteriori* (MAP) estimate for log-filterbank energies has a much simpler solution[3]

$$\begin{aligned}\hat{L}_{x-MAP}(q) &= \underset{L_x(q)}{\operatorname{argmax}} \left[p(L_x(q)|\boldsymbol{Y})\right] \\ &= \log\left(\alpha_q \beta_q\right) \\ &= \log \hat{E}_x(q).\end{aligned} \tag{9.30}$$

From (9.26) and (9.30), we can see that the MMSE and MAP estimates are closely related. Here, the MMSE and MAP log-filterbank energies differ by a term $\Delta(q)$, given by

$$\begin{aligned}\Delta(q) &= \hat{L}_{x-MAP}(q) - \hat{L_x}(q) \\ &= \log \alpha_q - \Psi_0(\alpha_q).\end{aligned} \tag{9.31}$$

[3]Further detail may be found in appendix B.

The MAP estimate is of interest because it is equivalent to the filterbank energy estimator; that is, both are MMSE optimal in the filterbank energy domains. This means the MMSE spectral energy estimator also happens to the MAP log-filterbank energy estimator (though not necessarily the MAP cepstral estimator). This is the estimator that was investigated in greater detail in Chapter 8. Here, we have a closed-form analytic solution for the SE estimator bias identified earlier.

The effect of α on the difference term $\Delta(q)$ and log-filterbank variance is shown in Fig. 9.2. Here we can see that the MAP estimate is always larger than the MMSE estimate (i.e. $\Delta(q) \geq 0$). Such a result is consistent with Jensen's inequality. Since the logarithm is a concave operator, we are forced to have the following relationship between the MMSE filterbank and MMSE log-filterbank energy estimators

$$\log E[\log E_q|\boldsymbol{Y}] \geq E[\log E_q|\boldsymbol{Y}]. \tag{9.32}$$

When $\alpha \gg 1$, the difference term (9.31) tends toward zero. This suggests the MAP and MMSE estimators should be equivalent at higher values of α. We may further note that α cannot take values below 1. This can be shown by substituting (9.12),(9.13) into (9.20),(9.21) then substituting into (9.23)[4]. As a result, substituting $\alpha_q = 1$ into (9.31) we find that the maximum difference is given as $\Delta_{\max} \approx 0.577$ (the Euler-Mascheroni constant). Further investigation into the behaviour of α is given in the following sub-section.

To finish describing the log-filterbank variable, covariance cross terms must be calculated. However, direct covariance estimation with (9.22) or (9.25) is quite difficult. Instead, a first order estimation of the log-filterbank variable can be made by approximating the logarithm with a Taylor expansion

$$\log\big(E_x(q)\big) \approx \log\big(\hat{E}_x(q)\big) - 1 + \frac{E_x(q)}{\hat{E}_x(q)}. \tag{9.33}$$

[4]See appendix C.

Which leads to log-filterbank covariance estimates of

$$\Sigma_{L_x}(q,q') \approx \frac{\Sigma_{E_x}(q,q')}{\hat{E}_x(q)\hat{E}_x(q')}. \tag{9.34}$$

It should be noted that the covariance $\mathbf{\Sigma}_{L_x}$ exhibits the same sparcity as $\mathbf{\Sigma}_{E_x}$. Thus only overlapped filterbank covariances need to be calculated. For a typical triangular Mel-filterbank, this involves covariance entries for immediately adjacent filterbanks, i.e. for $|q - q'| = 1$. Since the Taylor expansion is only a first order estimate, the more accurate (9.29) should still be used for calculating the dominant diagonal covariances.

9.3.1 Empirical analysis of the filterbank approximations

In this sub-section we examine the use of the gamma PDF for modelling the filterbank variable. For this experiment, we simulate a single filterbank, consisting of a few spectral bins with known $\lambda_x(k)$ and $\lambda_d(k)$. Using the $\lambda_x(k)$ and $\lambda_d(k)$ values, we may generate a parameter set of $X(k)$ and $D(k)$, (where $X(k)$ and $D(k)$ are realizations of complex zero-mean Gaussian RVs). Using (9.11), we can then determine the posterior PDF $p(e_x(k)|Y(k))$ for each bin. Without loss of generality, we assume the filterbank gain for each bin is unity. This means the true filterbank PDF can be estimated as the discrete convolution of individual (discretized) spectral energy PDFs. To determine how close this PDF is to the gamma PDF approximation, we must calculate gamma PDF parameters α and β. Filterbank energy mean and variance can be given as the summation of spectral energy means (9.12) and variances (9.13) respectively. Using these estimates, gamma PDF values for α and β, may be calculated from (9.23) and (9.24) respectively. For completeness, we also tried fitting Gaussian, log-Gaussian/normal and chi-square distributions to the filterbank PDF. These PDFs were fitted with an equivalent method of moments.

	Simulation parameters							PDF χ^2 fit			
Set	$\boldsymbol{\lambda}_x$	$\boldsymbol{\lambda}_d$	$\mathbf{X}$	$\mathbf{D}$	E_y	α	β	Gamma	Gauss.	Log-Norm.	Chi-square
A1	[9; 25; 16]	[1; 3; 1]	[-3.033, -3.668j; -0.529, -1.476j; -1.428, -1.739j]	[0.510, 0.201j; 0.416, -0.178j; 0.624, -0.063j]	25.032	11.644	2.152	0.011	1.255	0.289	0.067
A2	[9; 25; 16]	[1; 3; 1]	[0.935, 1.769j; 1.999, -7.910j; -1.962, 3.105j]	[-0.001, 0.147j; -1.977, 0.271j; -0.869, -0.711j]	76.648	15.558	4.300	0.017	0.597	0.273	3.763
A3	[4; 13; 33; 45; 15; 10]	[1; 2; 4; 2; 2; 1]	[-0.103, 2.088j; -2.535, -2.075j; -3.036, 2.620j; -0.146, -6.213j; 2.707, -2.376j; -1.339, -1.060j]	[0.157, -0.068j; 1.871, 0.446j; 0.156, -0.418j; -0.411, -0.168j; 0.511, 0.180j; -0.848, 0.298j]	81.900	20.982	3.810	0.012	0.438	0.193	2.784
B1	[9; 25; 16]	[10; 30; 10]	[-3.033, -3.668j; -0.529, -1.476j; -1.428, -1.739j]	[1.612, 0.636j; 1.314, -0.564j; 1.974, -0.200j]	20.053	2.767	10.685	0.042	4.518	0.757	14.440
B2	[9; 25; 16]	[10; 30; 10]	[0.935, 1.769j; 1.999, -7.910j; -1.962, 3.105j]	[-0.004, 0.464j; -6.253, 0.856j; -2.748, -2.250j]	96.628	3.150	15.408	0.002	3.445	0.834	21.285
B3	[4; 13; 33; 45; 15; 10]	[10; 20; 40; 20; 20; 10]	[-0.103, 2.088j; -2.535, -2.075j; -3.036, 2.620j; -0.146, -6.213j; 2.707, -2.376j; -1.339, -1.060j]	[0.497, -0.215j; 5.918, 1.410j; 0.492, -1.323j; -1.301, -0.531j; 1.617, 0.568j; -2.681, 0.942j]	109.427	5.441	16.712	0.003	2.097	0.440	22.689

Table 9.1: Chi-square analysis of the filterbank energy probability density function shape.

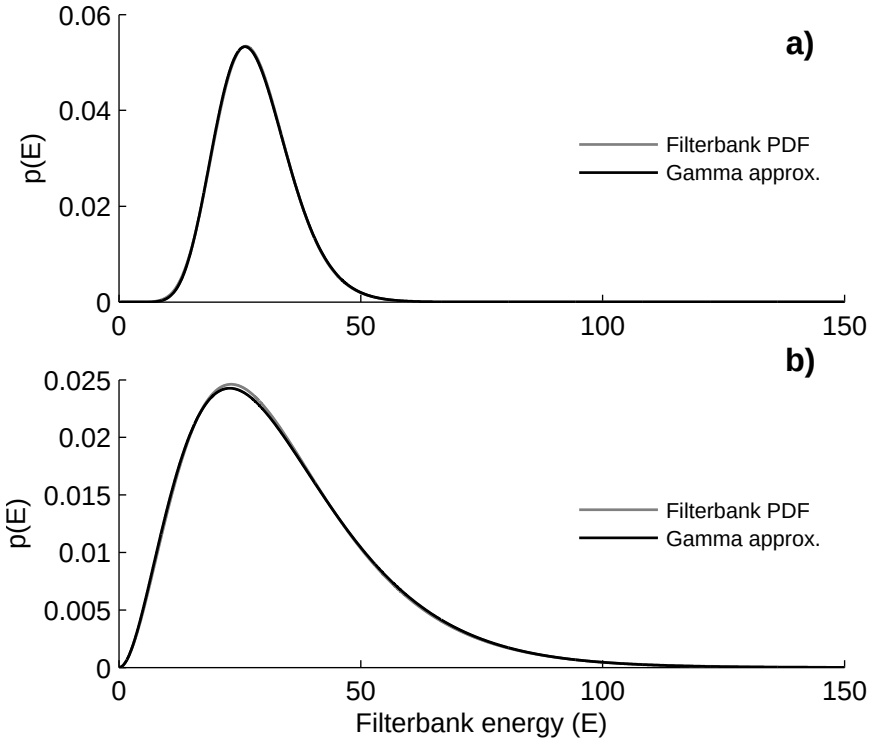

Figure 9.3: Using a gamma PDF to approximate the conditioned filterbank variables. The filterbank is the summation of three conditioned spectral energies whose PDFs are given by (9.11). Subplots: a) high SNR parameter set gamma PDF fitting (see table 9.1-A1) b) low SNR parameter set gamma PDF fitting (see table 9.1-B1). For the B1 simulation, the gamma PDF fitting begins to deviate from the true filterbank PDF.

To determine the quality of fit, we use a chi-square statistic,

$$\chi^2 = \sum_j \frac{\left(p_E[i] - p_T[i]\right)^2}{p_T[x]}, \tag{9.35}$$

where $p_E[i]$ is the discretized form of the approximating PDF and $p_T[i]$ is the discretized form of the true PDF. To calculate the χ^2 statistic we used discretized bins where $p_T[i] > 0.0001$.

Table 9.1, lists the PDF fitting results for six parameter sets. Here, six specific realizations of $\boldsymbol{X}$, $\boldsymbol{Y}$ are shown, generated from the listed values of $\boldsymbol{\lambda_x}$ and $\boldsymbol{\lambda_d}$. For simulations A1 and A2, we simulate a 3-bin 10 dB SNR filterbank. For simulations

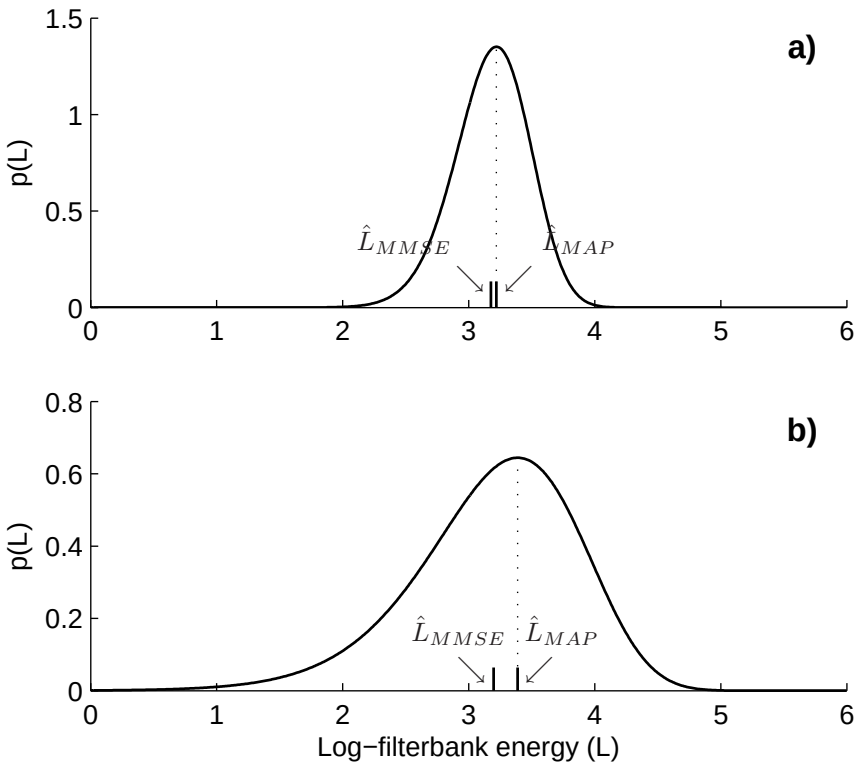

Figure 9.4: Plot of the conditioned log-filterbank PDF. Subplots: a) high SNR parameter set $\alpha = 11.64$, $\beta = 2.15$, $\hat{L} = 3.18$, $\hat{L}_{MAP} = 3.22$; b) low SNR parameter set $\alpha = 2.77$, $\beta = 10.69$, $\hat{L} = 3.20$, $\hat{L}_{MAP} = 3.39$. When α is large, the difference between the MAP and MMSE estimates is very small. The log-filterbank PDF given by (9.25) is valid under the assumption of gamma distributed conditioned filterbank energies.

B1 and B2, we simulate a 3-bin 0 dB SNR filterbank. Finally for simulations A3 and B3, we simulate 6-bin, 10 dB and 0 dB filterbanks respectively. For all of the simulations, the gamma PDF gave a much better fit than the other PDFs. In general, we noticed that the quality of the gamma PDF fit was consistently very good for larger values of α ($\alpha > 10$). Large α filterbank values typically occurred under two circumstances: 1) high SNR parameters were used, or 2) a large number of spectral bins were summed into the filterbank. In the opposite circumstances (low SNR, low spectral bin count), the gamma PDF fit was less consistent, sometimes having a suboptimal fit. One such suboptimal filterbank realization is shown in the B1 simulation. To show this visually, we plot the gamma PDF fitting of simulations

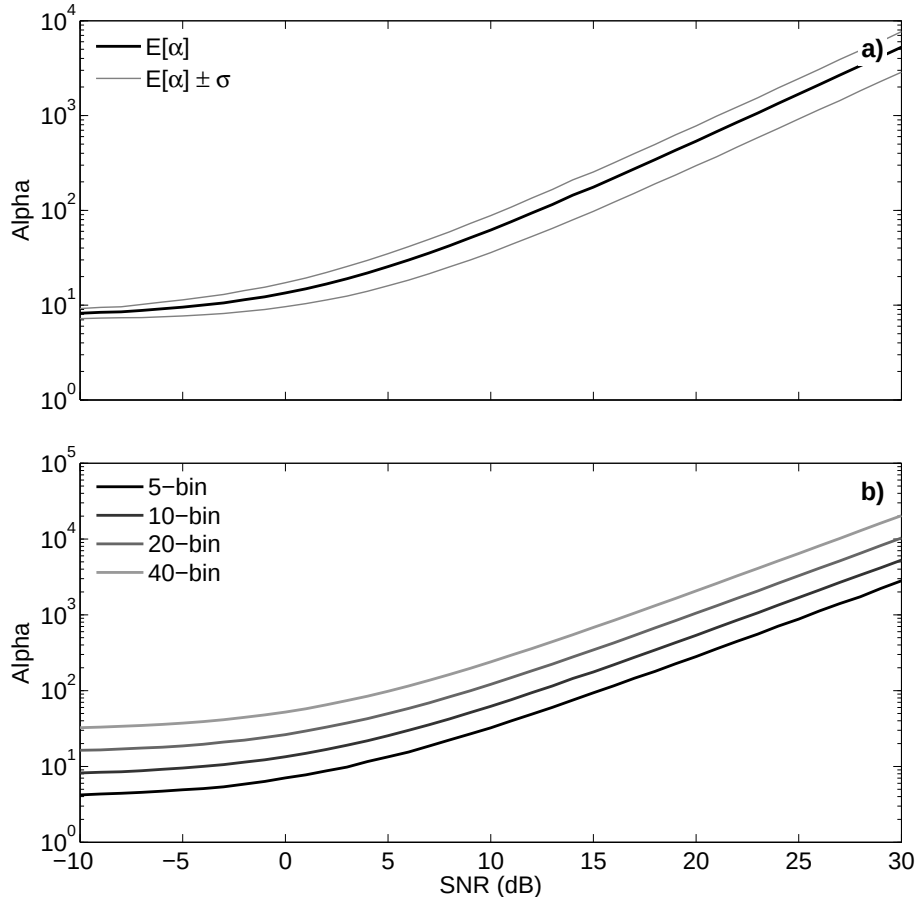

Figure 9.5: The effect of SNR and bin count on the filterbank shape parameter α. Subplots: a) Average α value for a 10-bin filterbank, with $\pm$ one standard deviation, and b) Average α value for 5-bin, 10-bin, 20-bin and 40-bin filterbanks.

A1 and B1 in Fig. 9.3a) and b) respectively. The simulation B1 is identical to A1 except that the energy of the noise $D(k)$ is increased ten-fold.

We did not investigate the shape of the log-filterbank PDF in as much detail, though we did find it was approximately Gaussian. Fig. 9.4 shows the log-filterbank PDF for simulations A1 and B1. When $\alpha \gg 1$, we found the log-filterbank PDF had a very good Gaussian fit. Under such conditions, the log-filterbank MAP and MMSE estimates converge.

The parameter α and its relationship to SNR and filterbank bin count is shown in Fig. 9.5. For this experiment, we simulate multiple filterbanks to find the mean

and variance of α. To do this, we first generate a set of $\boldsymbol{\lambda_x}$, $\boldsymbol{\lambda_d} \in \mathbb{R}^{B\times 1}$, where B is the desired filterbank bin count. Each element of $\boldsymbol{\lambda_x}$ and $\boldsymbol{\lambda_d}$ is then assigned a (uniform distributed) random value between the limits $[0, 10]$. The vector $\boldsymbol{\lambda_d}$ can then be scaled to give the desired SNR, where filterbank SNR is given as

$$\text{FBE SNR (dB)} = 10\log_{10}\left(\frac{\sum_k \lambda_x(k)}{\sum_k \lambda_d(k)}\right). \tag{9.36}$$

We then generate realizations of $X(k)$ and $D(k)$ in a similar manner to the previous experiment, using both to find α values. Values for α used here are averaged over 1000 realizations of $\boldsymbol{\lambda_x}$ and $\boldsymbol{\lambda_d}$, each of which is used to generate 1000 realizations of $X(k)$ and $D(k)$; i.e., one million filterbank simulations per SNR / bin count setting. In Fig. 9.5a), we show the range of α values for a 10-bin filterbank. For Fig. 9.5b) we show the average value of α for a 5-bin, 10-bin, 20-bin and 40-bin filterbank. Here, we can see a positive correlation between the SNR and α. The relationship between α and the number of filterbanks is even clearer. It is evident there is a linear relationship between filterbank count and α. That is, if filterbank bin count doubled, α will be doubled as well.

In the earlier filterbank chi-square fitting experiment, we performed an analysis of the gamma PDF approximation using several specific realizations of a filterbank variable. However, our primary goal is not to determine the quality of fit but to determine whether the gamma PDF assumption leads to suboptimal log-filterbank energy estimation. The following *toy* experiment operates under two main assumptions: 1) the statistical assumptions used by the spectral Wiener, SA and LSA estimators are correct, and 2) we have exact estimates of the *a priori* SNR $\xi(k)$ and *a posteriori* SNR $\gamma(k)$. A single experimental simulation consists of the following steps:

1. Generate realizations of $X(k)$ and $D(k)$ for $k = 0, 1, ...K-1$. $X(k)$ and $D(k)$ are complex zero-mean Gaussian RV realizations, generated from a known set of $\lambda_x(k)$ and $\lambda_d(k)$.

2. Calculate the oracle estimate for clean filterbank energy $L_{\text{oracle}} = \log\left(\sum_k |X(k)|^2\right)$.

3. Find clean filterbank energy estimates $\hat{L}_{\text{est}}$ using a particular estimator.

4. Determine the bias $[\hat{L}_{\text{est}} - L_{\text{oracle}}]$ and square error $[\hat{L}_{\text{est}} - L_{\text{oracle}}]^2$ of the estimate.

Bias and root mean-square-error (RMSE) is then calculated for each estimator over 500,000 such simulations. We use the spectral subtraction (SS), spectral Wiener (SW), log-spectral amplitude (LSA), spectral amplitude (SA), log-filterbank energy (LFBE)[143], proposed MAP (9.30) and proposed MMSE (9.28) estimators to generate log-filterbank energy estimates. For the LFBE estimator, we derive SNR parameters using filterbank versions of $\boldsymbol{\lambda_x}$ and $\boldsymbol{\lambda_d}$ as given in [143].

In table 9.2, we simulate a 5-bin, 10-bin and 20-bin log-filterbank variable at -10 dB, 0 dB and 10 dB SNRs. For the 5-bin simulations $\boldsymbol{\lambda_x} = [3, 250, 10, 100, 150]$ and $\boldsymbol{\lambda_d} \propto [3, 20, 20, 5, 30]$. For the 10-bin filterbank simulation, $\boldsymbol{\lambda_d} = [3, 3, 100, 250, 250, 100, 150, 50, 10, 4]$ and $\boldsymbol{\lambda_d} \propto [3, 10, 5, 5, 20, 50, 30, 10, 20, 20]$. Lastly, for the 20-bin simulation individual values for $\boldsymbol{\lambda_x}$ and $\boldsymbol{\lambda_d}$ are doubled up from the previous experiment, i.e. $\boldsymbol{\lambda_x} = [3, 3, 3, 3, 100, 100, 250, 250, 250, 250....]$. For all of the simulations, $\boldsymbol{\lambda_d}$ is scaled to give the required filterbank energy SNR (9.36).

From table 9.2, we can make a few observations. Firstly, a large positive bias is incurred when no enhancement is undertaken. Secondly, the MAP estimator was also consistently positively biased. The amount of MAP bias varied between the experiments, with low SNRs and low bin counts increasing the bias. This means that for the 10 dB SNR, 20-bin simulation, the MAP and MMSE estimators were virtually identical.

Ideally, in all conditions we would want the MMSE estimator to be unbiased. This was essentially true for all but the -10 dB 5-bin and -10 dB 10-bin simulations. This corresponds to the suboptimal conditions covered in previous experiments (i.e. low SNR, small number of filterbank bins). However, even under suboptimal

		-10 dB SNR		0 dB SNR		10 dB SNR	
		RMSE	**Bias**	**RMSE**	**Bias**	**RMSE**	**Bias**
5-bin filterbank	None	2.565	2.438	0.276	0.105	0.276	0.105
	SW	2.143	-1.962	0.270	-0.088	0.270	-0.088
	SS	3.732	0.174	0.300	-0.024	0.300	-0.024
	LSA	0.733	-0.388	0.260	-0.072	0.260	-0.072
	SA	0.625	-0.059	0.246	-0.018	0.246	-0.018
	LFBE[143]	0.630	0.047	0.266	0.040	0.266	0.040
	MAP	0.647	0.177	0.247	0.029	0.247	0.029
	MMSE	0.622	-0.009	0.245	-0.000	0.245	-0.000
10-bin filterbank	None	2.489	2.424	0.822	0.721	0.190	0.103
	SW	1.651	-1.520	0.578	-0.443	0.175	-0.082
	SS	1.636	1.242	0.541	0.164	0.169	0.000
	LSA	0.622	-0.443	0.417	-0.259	0.167	-0.068
	SA	0.454	-0.133	0.330	-0.085	0.152	-0.026
	LFBE[143]	0.467	-0.096	0.386	0.083	0.176	0.075
	MAP	0.444	0.091	0.322	0.0494	0.150	0.011
	MMSE	0.434	-0.002	0.318	-0.000	0.149	-0.000
20-bin filterbank	None	2.444	2.411	0.759	0.707	0.146	0.099
	SW	1.552	-1.484	0.500	-0.430	0.130	-0.078
	SS	1.533	1.391	0.394	0.202	0.112	0.007
	LSA	0.573	-0.485	0.352	-0.269	0.122	-0.068
	SA	0.351	-0.177	0.243	-0.105	0.105	-0.029
	LFBE[143]	0.372	-0.191	0.266	0.047	0.134	0.080
	MAP	0.307	0.046	0.220	0.024	0.100	0.005
	MMSE	0.303	-0.000	0.218	-0.000	0.100	-0.000

Table 9.2: Analysis of log-filterbank energy estimation using a simulated filterbanks.

conditions such as these, the vast majority of bias was successfully removed. Furthermore, this small deficiency tends to be eclipsed by the practical estimation of $\gamma(k)$ and $\xi(k)$ – a difficult task at very low SNRs.

In general the LFBE estimator [143] performed comparatively poorly on these tests, especially for the higher SNR / larger filterbank simulations. These cases

correspond to filterbanks with high degrees of freedom (which is roughly proportional to α). Here it appears the Rayleigh distribution (with two degrees of freedom) is too restrictive to adequately model filterbanks of varying size and/or SNR. We should point out that the framework we used for testing is not the framework used by the original authors to derive the MMSE LFBE estimator [143]. Here, filterbank energies were described as being the amplitudes of a hidden, complex zero-mean Gaussian RVs. This is the same model Ephraim and Malah used to model spectral amplitudes [39, 40]. The motivation for this was the reduction in computational complexity offered by applying the LSA estimator directly on the filterbank level; i.e., tracking 20-30 filterbank SNR parameters instead of a few hundred spectral SNR parameters. However, there does not appear to be any other justification for this and such a modelling assumption violates the statistical framework used by the original LSA estimator [40].

9.4 Models for Mel-frequency cepstral coefficients

In this section we extend the log-filterbank energy variables $\boldsymbol{L}_x$ into cepstral variables $\boldsymbol{c}_x$. Since the DCT is a linear transform, the conditioned expectation of clean speech cepstral vector $\boldsymbol{c}_x$ is given as

$$\begin{aligned} E[\boldsymbol{c}_x|\boldsymbol{Y}] &= \boldsymbol{\mu}_c \\ &= \mathbf{C}\hat{\boldsymbol{L}}_x, \end{aligned} \tag{9.37}$$

where $\boldsymbol{\mu}_c \in \mathbb{R}^{Q\times 1}$, is the mean MFCC vector and $\mathbf{C} \in \mathbb{R}^{Q\times Q}$ is the discrete cosine transform matrix. Given log-filterbank covariance matrix $\boldsymbol{\Sigma}_{L_x}$, the cepstral covariance matrix $\boldsymbol{\Sigma}_c$ can be given as

$$\begin{aligned} COV[\boldsymbol{c}_x|\boldsymbol{Y}] &= \boldsymbol{\Sigma}_c \\ &= \mathbf{C}\boldsymbol{\Sigma}_{L_x}\mathbf{C}^T. \end{aligned} \tag{9.38}$$

Given typical speech, the non-diagonal entries of $\boldsymbol{\Sigma}_c$ tend to be quite small. Because of this, later calculations can be greatly simplified if all non-diagonal covariance entries are assumed to be zero.

Given conditioned estimates for the cepstral mean and covariance, we can describe the cepstral vector with a multivariate Gaussian distribution. Use of the Gaussian PDF is motivated by the central limit theorem, as each cepstral coefficient is a summation of several log-filterbank variables. This choice is further reinforced by the fact that log-filterbank energies are themselves approximately Gaussian distributed (see Fig. 9.4). Thus the final model of the clean cepstral vector $\boldsymbol{c}_x$ is given by

$$p(\boldsymbol{c}_x|\boldsymbol{Y}) = \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c). \tag{9.39}$$

If no additional information is available, a final MMSE clean speech cepstral estimate $\hat{\boldsymbol{c}}_x$ is given as

$$\hat{\boldsymbol{c}}_x = \boldsymbol{\mu}_c. \tag{9.40}$$

9.4.1 Use of an *a priori* speech model

The low-level statistical framework described in earlier sections does not leverage any speech specific information. Instead, *a priori* knowledge was incorporated at the individual spectral coefficient level via the parameter $\lambda_x(k)$. A downside of this approach is that we have not incorporated *a priori* knowledge at the cepstral level. As a result, the cepstral vector estimate $\boldsymbol{\mu}_c$ is not actually guaranteed to resemble speech. To address this, we may recast our estimation problem into a Bayesian estimator for cepstral-level variables. This allows the direct use of a more informative cepstral domain prior. The cepstral domain Bayesian estimator can be given as

$$\begin{aligned} \hat{\boldsymbol{c}}_x &= E[\boldsymbol{c}_x|\boldsymbol{c}_y] \\ &= \frac{\int_{-\infty}^{\infty} \boldsymbol{c}_x p(\boldsymbol{c}_x) p(\boldsymbol{c}_y|\boldsymbol{c}_x) d\boldsymbol{c}_x}{\int_{-\infty}^{\infty} p(\boldsymbol{c}_x) p(\boldsymbol{c}_y|\boldsymbol{c}_x) d\boldsymbol{c}_x}. \end{aligned} \tag{9.41}$$

We can use a Gaussian mixture model (GMM) to model the prior PDF

$$p(\boldsymbol{c}_x) = \sum_i \pi_i \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{9.42}$$

where π_i, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ are the mixture weight, mean and diagonal covariance of the i'th Gaussian mixture respectively. The GMM parameters are learned from a clean speech training corpus using an expectation maximization algorithm. To determine the conditional PDF $p(\boldsymbol{c}_y|\boldsymbol{c}_x)$, we may use the approach given in [70]. Here, a cepstral distortion variable $\boldsymbol{c}_d$ is defined. Under our cepstral framework, $\boldsymbol{c}_d$ can be modelled with mean

$$\hat{\boldsymbol{c}}_d = \boldsymbol{c}_y - \boldsymbol{\mu}_c, \tag{9.43}$$

and diagonal variance $\boldsymbol{\Sigma}_c$. Like $\boldsymbol{c}_x$, $\boldsymbol{c}_d$ is Gaussian distributed, and as a result so is the conditioned PDF $p(\boldsymbol{c}_y|\boldsymbol{c}_x)$,

$$p(\boldsymbol{c}_y|\boldsymbol{c}_x) = \mathcal{N}(\boldsymbol{c}_y; \boldsymbol{c}_x + \hat{\boldsymbol{c}}_d, \boldsymbol{\Sigma}_c). \tag{9.44}$$

Substituting, (9.42), (9.43) and (9.44) into (9.41), gives

$$\begin{aligned}\hat{\boldsymbol{c}}_x &= \frac{\int_{-\infty}^{\infty} \boldsymbol{c}_x \left[\sum_i \pi_i \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right] \mathcal{N}(\boldsymbol{c}_y; \boldsymbol{c}_x + \hat{\boldsymbol{c}}_d, \boldsymbol{\Sigma}_c) d\boldsymbol{c}_x}{\int_{-\infty}^{\infty} \left[\sum_i \pi_i \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right] \mathcal{N}(\boldsymbol{c}_y; \boldsymbol{c}_x + \hat{\boldsymbol{c}}_d, \boldsymbol{\Sigma}_c) d\boldsymbol{c}_x} \\ &= \frac{\int_{-\infty}^{\infty} \boldsymbol{c}_x \left[\sum_i \pi_i \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right] \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) d\boldsymbol{c}_x}{\int_{-\infty}^{\infty} \left[\sum_i \pi_i \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right] \mathcal{N}(\boldsymbol{c}_x; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) d\boldsymbol{c}_x}.\end{aligned} \tag{9.45}$$

This can be further simplified to give [32, 70]

$$\hat{\boldsymbol{c}}_x = \sum_i \zeta_i \boldsymbol{b}_i, \tag{9.46}$$

where

$$\zeta_i = \frac{\pi_i \, \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{c}}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_c + \boldsymbol{\Sigma}_i\right)}{\sum_{i'} \pi_{i'} \, \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{c}}; \boldsymbol{\mu}_{i'}, \boldsymbol{\Sigma}_c + \boldsymbol{\Sigma}_{i'}\right)}, \tag{9.47}$$

$$\boldsymbol{b}_i = \boldsymbol{B}_i \left[\boldsymbol{\Sigma}_c^{-1} \boldsymbol{\mu}_c + \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i\right], \tag{9.48}$$

$$\boldsymbol{B}_i = \boldsymbol{\Sigma}_i \boldsymbol{\Sigma}_c \left[\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_c\right]^{-1}. \tag{9.49}$$

Though calculation of the estimation variance is not required for the ASR experiments given in Section 9.5, we provide it here for the interested reader,

$$\begin{aligned} E\left[\{c_x(q) - \hat{c}_x(q)\}^2 | \boldsymbol{c}_y\right] &= \Sigma_{c_x}(q,q) \\ &= \left[\sum_i \zeta_i \left(B_i(q,q) + [b_i(q)]^2\right)\right] - [\hat{c}_x(q)]^2. \end{aligned} \tag{9.50}$$

It should be noted that the estimate $\hat{\boldsymbol{c}}_x$ depends implicitly on the observed data (and noise estimates) via $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ computed in (9.37) and (9.38) respectively. We may make a few additional observations about the estimate $\hat{\boldsymbol{c}}_x$ computed in (9.45),(9.46). Firstly, if we use an uninformative (constant) prior PDF, we would simply get the estimation $\hat{\boldsymbol{c}}_x = \boldsymbol{\mu}_c$. Likewise, the estimation variance $\boldsymbol{\Sigma}_{c_x}$ will be unchanged from $\boldsymbol{\Sigma}_c$. When an informative prior PDF is used, the estimate $\hat{\boldsymbol{c}}_x$ is given as a linear combination of the evidence estimate $\boldsymbol{\mu}_c$ and the *a priori* GMM means $\boldsymbol{\mu}_i$. The amount of mixing is dependent on the reliability (variance) of both the conditioned evidence PDF and the prior PDF. When the evidence PDF $p(\boldsymbol{c}_y|\boldsymbol{c}_x)$ has small variance relative to the GMM, the evidence estimate $\boldsymbol{\mu}_c$ will dominate the final estimate. When the evidence PDF has large variance relative to the GMM, more of the prior model is leveraged. In this way, the speech model allows us to reconstruct the unreliable portions of estimate $\boldsymbol{\mu}_c$. The use of a prior model also tends to significantly reduce estimation variance. Of course, both of these benefits are dependent on an informative and relevant prior speech model being used. In addition, the use of a prior speech model comes with the cost of substantial computational overhead.

Similar methods that regenerate poor speech features have been proposed earlier in the literature, notably missing feature imputation [116]. However in the above approach, the amount of prior model mixed into the final estimate is controlled by the relative variances of $p(\boldsymbol{c}_y|\boldsymbol{c}_x)$ and $p(\boldsymbol{c}_x)$, rather than a binary decision rule.

9.5 Experimental results

9.5.1 Enhancement system description

For our experiments, we decompose speech utterances into overlapping frames. Each analysis frame is 25 ms in length, and overlaps the previous analysis frame by 15 ms. Each analysis frame has a Hamming window applied before being enhanced with a given regime. To derive the noise estimate $\lambda_d(m,k)$, we use a simple voice activity detector (VAD). An initial noise estimate is generated from the first 125 ms of each speech stimulus, and recursively updated. The recursive update is given as follows

$$\lambda_d(m,k) = \eta\lambda_d(m-1,k) + (1-\eta)e_y(k), \tag{9.51}$$

where $\eta = 0.98$ in the case that a noise-only frame has been detected and $\eta = 1$ otherwise. The *a posteriori* SNR can be then be calculated via (9.10). To calculate the *a priori* SNR ξ, we use the decision-directed approach covered in section 9.2.2. For the LFBE estimator, filterbank-level λ_x and λ_d as estimated as per [143], though we use a VAD for the noise estimation.

For the proposed MFCC estimator, the estimation of a single MFCC frame can be summarized as follows:

1. For each spectral bin, estimate spectral energies (8.23) and variance (9.19).
2. Determine each filterbank energy (9.20) and variance (9.21).
3. For each filterbank, estimate filterbank shape parameter α (9.23).
4. Accumulate log-filterbank MAP energies (9.30).
5. Correct MAP log-filterbank difference (9.28), calculate log-filterbank variance (9.29) and covariance (9.34).
6. Calculate cepstral mean (9.37) and variance (9.38).
7. If a speech prior PDF is given, calculate adjusted cepstral estimates (9.46).

A diagram detailing the relationships between estimation variables, is given in Fig. 9.6.

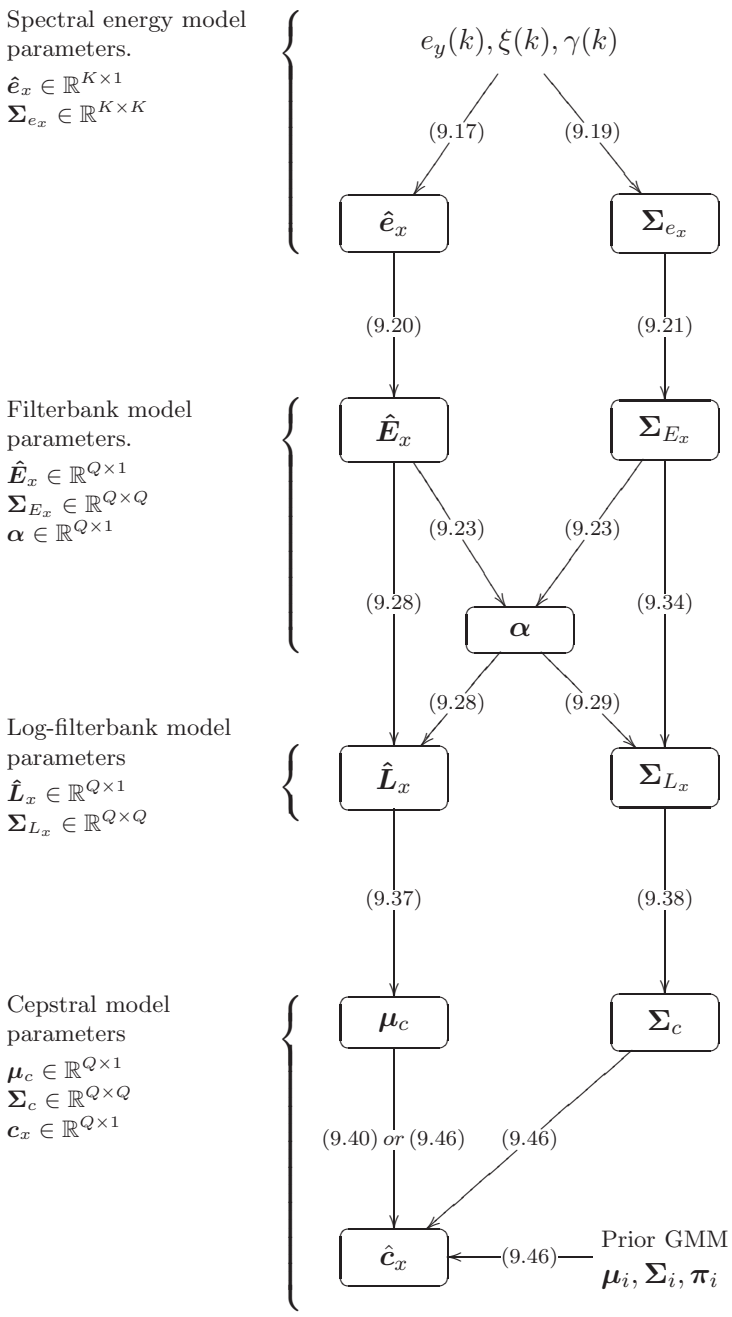

Figure 9.6: Mathematical relationship between spectral energy model parameters, filterbank energy model parameters, log-filterbank energy model parameters and cepstral model parameters.

9.5.2 Automatic speech recognition system description

To test ASR performance, we use a standard MFCC feature set in conjunction with the HTK recognition framework [142]. We accumulate 26 log-filterbank energies, and retain the first 12 cepstral coefficients (excluding the zeroth). In place of the zeroth cepstral coefficient, the total log energy of each frame is used. Once this is done, we append delta and acceleration coefficients to give a 39 dimensional feature vector. Training is provided by clean, unaltered utterances. We give results for the MMSE spectral amplitude (SA), MMSE log-spectral amplitude (LSA), VTS [87] and log-filterbank energy (LFBE) [143] estimators. For the VTS estimator, we use a 16 mixture diagonal covariance log-filterbank GMM (built from the clean speech training corpus) for the speech prior.

We also give results for the proposed MAP, MMSE and MMSE with prior (MMSE-p) estimators. For the MAP estimator, we build cepstral vectors from the log-filterbank MAP estimates (9.30). For the MMSE prior estimator, we use a 16 mixture diagonal covariance cepstral GMM for the speech prior. A tabular overview of the ASR system used can be found in Chapter 8, table 8.2. We conduct experiments over the Aurora2 digit and RM speech recognition tasks. Both speech databases are summarized below.

9.5.3 Resource management word recognition

A speaker independent section of the DARPA resource management (RM) database is used for medium-vocabulary recognition [110]. The database was recorded in clean conditions (sample rate of 16 kHz) and has a vocabulary of approximately 1000 words. For training, there are 3990 sentences spoken by 109 speakers. For testing, we use the February '89 test set which has 300 sentences spoken by 10 different speakers. White, Volvo and babble noises are artificially added at several SNRs. For recognition, we train triphone-level HMMs, having three states with eight Gaussian mixtures each. Cepstral mean subtraction (CMS) is applied as a standard post-processor. For the proposed estimators, we use an SPU of $q(k) = 0.3$.

For the SA and LSA estimators, SPU did not improve recognition accuracy and was thus omitted. ASR word error rate (WER) scores are given in table 9.5.

9.5.4 Aurora2 digit recognition

Aurora2 is a speaker independent database for connected digit recognition [104]. Unlike the RM database, Aurora2 lacks a language model, though its acoustic models are relatively sparse. Spoken digits in the database consist of zero through nine as well as 'oh', giving a vocabulary size of 11. Testing and training utterances were down-sampled to 8 kHz and filtered with G712 characteristics. Finally, utterances have had noise artificially added at several SNRs. Word-level HMMs are built, each with 16 states and 3 Gaussian mixtures per state. CMS was applied as a standard post-processor. For the proposed estimators, we use an SPU of $q(k) = 0.05$. For the SA and LSA estimators, SPU severely degraded recognition accuracy and was thus omitted. ASR WER scores are given in tables 9.3 and 9.4 for recognition tasks A and B respectively.

9.5.5 Discussion

Across both the RM and Aurora2 tasks, the proposed MAP, MMSE and MMSE-prior MFCC estimators can be seen to give good recognition performance in comparison to the other estimators. For the Aurora2 recognition task, there was a fairly consistent reduction in WER when moving from the MAP estimator to the MMSE estimator and finally to the MMSE-prior estimator. Not surprisingly, the MAP and MMSE estimators only diverged at lower SNRs. Use of the speech prior on the other hand, improved results across the entire SNR range. While uniform across SNR, this improvement was more pronounced for the non-stationary noises – such as babble, exhibition and airport. Other estimators (that do not use a prior model) typically struggled with these noise tasks, which likely reflects the use of VAD to track non-stationary noise. This suggests the use of an *a priori* speech model can decreases sensitivity to suboptimal noise estimation.

For the RM recognition task, differences between the MAP, MMSE and MMSE-prior MFCC estimators were much smaller. Here, the benefits of removing bias (MMSE and MMSE-prior estimators) seemed to be offset by increased speech degradation at higher SNRs. Use of the prior model likewise gave quite small gains in improvement. It appears that the RM task is less suited to the use of a prior speech model. This is best reflected in the poor results for the VTS estimator (pure model-based enhancement) in comparison to the standard noise suppressors (SA and LSA estimators). One reason for this may be that the RM *a priori* speech model is simply less informative. For the RM task, triphone models were used – many of which are already acoustically similar. Here, using a global speech model to make cepstral estimates more 'speech-like' appears to give marginal benefits. This is in comparison to Aurora2 task, where individual word models are acoustically well separated. We should point out that while the VTS estimator (pure model based enhancement) performed poorly for this task, the proposed estimators did not.

9.6 Conclusion

In this chapter we have investigated the derivation of a statistical framework for MFCC estimation. In order to avoid as many statistical assumptions as possible, we based our framework off a well established spectral domain framework. To make this framework suitable for MFCC estimation, several mathematical transformation were studied to covert spectral domain models into cepstral domain models. With the framework derived, two MMSE estimators were proposed. Experimental results show a significant improvement in ASR robustness, over both the baseline results and several common front-end enhancement methods.

	Treatment	SNR (dB) ∞	20	15	10	5	0	AVG
Subway noise	None	0.68	2.95	5.59	12.34	30.49	70.56	24.39
	SA	0.89	3.07	4.54	9.12	17.68	39.70	14.82
	LSA	1.04	3.90	5.68	11.30	20.94	42.95	16.95
	VTS	0.68	2.67	5.37	11.79	25.76	53.15	19.75
	LFBE	0.74	3.81	5.83	13.72	35.40	70.62	25.88
	MAP	0.74	3.04	4.76	9.30	19.10	43.41	15.92
	MMSE	0.77	2.92	4.64	9.15	18.21	42.03	15.39
	MMSE-p	0.74	2.82	4.79	9.18	17.68	40.65	15.02
Babble noise	None	0.76	2.03	4.63	16.51	53.02	90.84	33.41
	SA	0.82	3.42	7.56	17.23	38.33	68.80	27.07
	LSA	0.88	8.86	15.57	26.57	45.77	70.95	33.54
	VTS	0.68	2.24	4.20	12.06	34.67	75.18	25.67
	LFBE	0.74	4.20	10.94	26.57	54.93	85.25	36.38
	MAP	0.74	2.00	4.11	11.79	31.71	68.20	23.56
	MMSE	0.77	2.09	4.53	12.70	32.29	66.20	23.56
	MMSE-p	0.74	2.06	4.38	11.34	30.14	65.24	22.63
Car noise	None	0.78	2.09	4.38	11.39	34.39	81.12	26.67
	SA	0.81	1.58	2.80	5.31	14.08	34.30	11.61
	LSA	0.92	1.70	3.40	6.53	16.10	38.98	13.34
	VTS	0.68	2.27	4.00	6.89	14.94	52.16	16.05
	LFBE	0.74	1.97	3.58	9.78	34.09	74.98	24.88
	MAP	0.74	1.79	2.86	6.32	16.10	43.93	14.20
	MMSE	0.77	1.73	2.80	5.91	14.88	40.41	13.15
	MMSE-p	0.74	1.73	2.77	5.82	14.38	39.82	12.90
Exhibition noise	None	0.74	3.73	6.97	15.21	38.88	82.14	29.39
	SA	0.80	3.70	6.17	13.45	28.51	51.96	20.76
	LSA	1.02	6.76	9.97	18.82	35.51	57.76	25.76
	VTS	0.68	3.12	4.44	9.04	19.84	44.62	16.21
	LFBE	0.74	3.86	8.36	19.25	48.19	81.61	32.25
	MAP	0.74	3.33	5.62	12.19	27.24	54.86	20.65
	MMSE	0.77	3.21	5.65	11.54	25.58	51.71	19.54
	MMSE-p	0.74	3.12	5.43	10.77	23.45	49.12	18.38
Set A averages	None	0.74	2.70	5.39	13.86	39.20	81.17	28.47
	SA	0.83	2.94	5.27	11.28	24.65	48.69	18.57
	LSA	0.97	5.31	8.66	15.81	29.58	52.66	22.40
	VTS	0.68	2.58	4.50	9.95	23.80	56.28	19.42
	LFBE	0.74	3.46	7.18	17.33	43.15	78.11	29.85
	MAP	0.74	2.54	4.34	9.90	23.58	52.60	18.58
	MMSE	0.77	2.49	4.41	9.83	22.74	50.09	17.91
	MMSE-p	0.74	2.43	4.34	9.28	21.41	48.71	17.23

Table 9.3: Aurora2A ASR word error rates. WER average is computed from 0 dB to 20 dB SNR.

		SNR (dB)						
Treatment		∞	**20**	**15**	**10**	**5**	**0**	**AVG**
Restaurant noise	None	0.68	2.33	4.70	16.18	45.96	78.97	29.63
	SA	0.89	5.13	11.30	23.64	42.03	67.58	29.94
	LSA	1.04	11.45	18.02	31.01	47.80	71.45	35.95
	VTS	0.68	2.00	4.82	12.93	33.74	67.18	24.13
	LFBE	0.74	5.22	12.62	28.22	52.07	79.71	35.57
	MAP	0.74	3.04	6.29	15.20	36.26	66.17	25.39
	MMSE	0.77	2.98	6.45	15.84	36.66	66.26	25.64
	MMSE-p	0.74	2.92	6.36	15.14	35.34	65.12	24.98
Street noise	None	0.76	2.60	5.11	13.45	37.64	75.09	26.78
	SA	0.82	3.33	5.20	9.95	22.61	47.85	17.79
	LSA	0.88	4.41	7.65	13.48	28.96	54.38	21.78
	VTS	0.68	3.23	4.87	10.82	24.21	55.41	19.71
	LFBE	0.74	3.20	6.41	16.93	42.62	75.97	29.03
	MAP	0.74	2.60	4.59	8.62	23.61	52.66	18.42
	MMSE	0.77	2.48	4.41	8.43	22.10	49.64	17.41
	MMSE-p	0.74	2.51	4.35	8.28	21.10	48.49	16.95
Airport noise	None	0.78	1.79	4.09	11.72	37.16	73.90	25.73
	SA	0.81	2.54	5.91	13.09	29.76	53.80	21.02
	LSA	0.92	5.13	9.54	18.70	35.67	58.75	25.56
	VTS	0.68	1.37	3.34	7.84	22.25	53.71	17.70
	LFBE	0.74	2.71	6.50	18.52	46.47	75.63	29.97
	MAP	0.74	1.82	3.55	9.48	25.77	54.88	19.10
	MMSE	0.77	1.88	3.64	10.02	25.77	53.06	18.87
	MMSE-p	0.74	1.85	3.58	9.42	24.49	51.15	18.10
Train noise	None	0.74	1.54	4.32	11.66	34.74	80.31	26.51
	SA	0.80	2.93	4.75	8.05	20.98	40.76	15.49
	LSA	1.02	4.17	6.39	11.08	24.07	44.99	18.14
	VTS	0.68	1.60	4.35	9.26	21.14	54.24	18.12
	LFBE	0.74	2.90	5.12	12.13	40.82	74.85	27.16
	MAP	0.74	1.94	3.86	7.22	20.12	48.35	16.30
	MMSE	0.77	2.16	3.76	7.03	19.44	44.80	15.44
	MMSE-p	0.74	2.13	3.86	6.54	18.73	43.72	15.00
Set B averages	None	0.74	2.07	4.56	13.25	38.88	77.07	27.16
	SA	0.83	3.48	6.79	13.68	28.85	52.50	21.06
	LSA	0.97	6.29	10.40	18.57	34.13	57.39	25.36
	VTS	0.68	2.05	4.35	10.21	25.34	57.64	19.92
	LFBE	0.74	3.51	7.66	18.95	45.50	76.54	30.43
	MAP	0.74	2.35	4.57	10.13	26.44	55.52	19.80
	MMSE	0.77	2.38	4.57	10.33	25.99	53.44	19.34
	MMSE-p	0.74	2.35	4.58	9.85	24.92	52.12	18.76

Table 9.4: Aurora2B ASR word error rates. WER average is computed from 0 dB to 20 dB SNR.

		SNR (dB)					
Treatment		∞	**30**	**20**	**10**	**0**	**AVG**
White noise	None	4.30	5.48	11.89	47.13	95.89	17.20
	SA	4.26	5.04	7.43	27.02	79.55	10.94
	LSA	4.42	5.36	7.55	23.39	76.85	10.18
	VTS	4.18	4.77	8.96	47.01	96.36	16.23
	LFBE	4.18	5.28	8.25	29.06	90.61	11.69
	MAP	4.50	5.01	7.00	23.03	75.87	9.89
	MMSE	4.54	4.97	6.92	23.19	76.46	9.91
	MMSE-p	4.42	4.93	6.96	23.19	75.87	9.88
Babble noise	None	4.30	4.73	8.21	38.87	94.68	14.03
	SA	4.26	4.89	7.78	28.90	89.87	11.46
	LSA	4.42	5.08	8.56	32.58	90.61	12.66
	VTS	4.18	4.81	6.92	31.13	99.26	11.76
	LFBE	4.18	4.77	9.93	43.76	100.31	15.66
	MAP	4.50	4.93	7.74	30.00	90.30	11.79
	MMSE	4.54	5.04	7.90	30.31	89.71	11.95
	MMSE-p	4.42	5.01	7.67	28.78	88.46	11.47
Volvo noise	None	4.30	4.03	5.01	7.86	23.03	5.30
	SA	4.26	4.42	4.34	4.73	10.09	4.44
	LSA	4.42	4.61	4.46	4.93	8.76	4.61
	VTS	4.18	4.42	5.71	9.07	17.21	5.84
	LFBE	4.18	4.18	4.89	8.45	21.20	5.42
	MAP	4.50	4.69	4.42	4.93	8.96	4.64
	MMSE	4.54	4.69	4.38	4.89	8.88	4.63
	MMSE-p	4.42	4.73	4.50	5.04	8.53	4.67
Average	None	4.30	4.75	8.37	31.29	71.20	12.18
	SA	4.26	4.78	6.52	20.22	59.84	8.94
	LSA	4.42	5.02	6.86	20.30	58.74	9.15
	VTS	4.18	4.67	7.20	29.07	70.94	11.28
	LFBE	4.18	4.74	7.69	27.09	70.71	10.93
	MAP	4.50	4.88	6.39	19.32	58.38	8.77
	MMSE	4.54	4.90	6.40	19.46	58.35	8.83
	MMSE-p	4.42	4.89	6.38	19.00	57.62	8.67

Table 9.5: RM ASR word error rates. WER average is computed from 10 dB to ∞ dB SNR.

Part III

Thesis conclusion

Chapter 10

Thesis summary, conclusions and future work

10.1 Speech processing literature review summary

In Chapters 2 and 3, a broad review of speech processing literature was given. Chapter 2 examined the general properties of the speech production and perception systems. In addition to this, a brief synopsis of speech based Fourier phase spectrum processing was also given, detailing its signal processing requirements and current shortcomings.

In Chapter 3, the structure of a generic state-of-the-art ASR system is detailed. The two main components of an ASR system: the front end feature extraction stage and back end classifier stage are explained. The problem of additive noise in ASR is then introduced, including a review of current methods used to address it.

10.2 Use of the short-time Fourier phase spectrum for speech processing

10.2.1 Summary

Chapter 4

In Chapter 4, the direct mathematical relationships that exist between the short-time magnitude and short-time phase spectra were investigated. Specific focus was given to the problem of regenerating spectral magnitude from phase and vice versa. In section 4.3, a linear method for regenerating a magnitude spectrum from phase was shown. In the latter case (reconstruction of spectral phase from magnitude), a modified version of the Gerchberg-Saxton algorithm was proposed. The proposed algorithm was shown to reconstruct phase spectra at roughly twice the rate of the standard Gerchberg-Saxton algorithm.

Chapters 5 and 6

In Chapters 5 and 6, the derivatives of the short-time Fourier phase spectrum were examined. In Chapter 5 we focused attention on the short-time instantaneous frequency (IF) spectrum. Here, it was found that the choice of analysis window was critically important for IF spectrum analysis. In particular, we have shown that novel IF behaviour could be induced given an appropriate windowing function. This is in contrast to the magnitude spectrum, whose core properties are generally insensitive to window choice.

Of particular importance to narrow-band IF analysis is the spectral leakage profile of the analysis window. For tradition IF analysis, clustering behaviour in the IF spectrum is preferred. This necessitates the use of a high sidelobe decay window in order to minimize the interaction of sidelobe leakage. However, further study of the IF spectrum lead to other avenues of extracting useful information. In particular, we focused on the IF deviation spectrum, discussing its relationship to the IF spectrum and characterising its behaviour. This deviation term contained many

useful traits, and was shown to manifest both pitch and formant information for narrow-band analysis. Unlike the high sidelobe decay windows used for IF clustering purposes, it was found that a zero-decay 50 dB Chebyshev window was optimal for exploiting information within the IF deviation quantity. Using the traits of the IF deviation, a new spectral representation was proposed.

In Chapter 6, a similar analysis was conducted with the related group delay spectrum. Like the IF spectrum, the GD spectrum could be mathematically described as a static term summed with an oscillating deviation term. Ultimately, this oscillation term was found to contain only a subset of the information contained in the corresponding the IF deviation term. While the GD deviation was able to characterize the position of spectral components, we did not find any mechanism for conveying further information. Because of this, the narrow-band GD deviation quantity displayed pitch (excitation source), but not formant (vocal tract) information. Nonetheless, using this information a novel group delay based spectral representation was proposed.

Chapter 7

In Chapter 7, a speech enhancement algorithm centred around short-time phase spectrum manipulation was presented. It has previously been shown that noise suppression could be achieved by altering the phase of Fourier expansion coefficients. This is in contrast to the majority of spectral speech enhancement methods which modify the short-time magnitude spectrum in order to induce suppression. In section 7.2, the mechanism used to achieve phase spectrum based noise suppression was detailed from previous literature [138]. This involved shifting the phase of Fourier expansion conjugate pairs. To exploit this effect for speech enhancement purposes, a simple noise-driven heuristic was derived in the present work. This allowed the creation of a simple AMS based speech enhancement algorithm. Testing revealed that the enhancement algorithm was able to suppress noise with minimal appearance of musical noise. This was confirmed through informal listening tests, as well as the

PESQ metric. One possible explanation for the reduction in musical noise is that our algorithm smoothly warps the original phase spectrum. This means the resulting enhanced signal lacked the sharp spectral peaks and valleys that tend to characterise musical noise.

10.2.2 Future work

In Chapters 5 and 6, we have shown that the IF deviation and GD deviation spectra contain many useful traits. In the process of this investigation, it was found that the analysis window leakage profile was of critical importance – especially in the derivation of instantaneous frequencies. It is interesting to note that the Hann and Blackman windows favoured by previous authors (for deriving IF density based representations), do not possess the highest sidelobe decay rate possible. This implies that many previously introduced IF quantities may be improved with the choice of a more appropriate window. For example, one may arbitrarily increase sidelobe decay rate (at the cost of mainlobe width), by using the general cosine window family

$$w(n) = \left[0.5\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right)\right]^k, \quad 0 \leq n \leq N, \tag{10.1}$$

where k is a tuning parameter. Setting $k = 1$ yields the Hann window while raising k further also increases the degree of sidelobe decay rate [64]. Thus many IF density based representations could be improved by directly choosing a sidelobe decay rate suitable to the particular application.

Another area of phase spectrum analysis that has potential for improvement is the reduction of volatility. By using the phase spectrum derivatives (IF and GD spectra), we have shown that the phase is well behaved at dominant sinusoid locations. However, in areas of high leakage interaction, the phase is chaotic. The natural way to remedy this is to smooth the phase derived quantities with respect to time, frequency or both. However, another potential avenue of averaging is to smooth the phase characteristics over multiple analysis windows. Similar analysis

window functions will all yield similar phase characteristics at dominant sinusoid locations. However, phase characteristics at other (leakage) frequency locations would vary substantially due to the changing interactions of sidelobe leakage. This would allow for more accurate identification of sinusoidal components, as spurious phase values would be easier to identify. Such behaviour suggests we would be able to use multiple windows to improve many phase based spectral representations, including the ones presented in this dissertation.

As a final note, while we have shown the phase spectrum to possess many useful speech features, we have not yet shown it to contain useful information complementary to the magnitude spectrum. Such a goal remains an interesting and challenging problem for the future.

10.3 Environment-robust estimation of Mel-frequency cepstral coefficients

10.3.1 Summary

Chapter 8

In Chapter 8 we have investigated the use of the spectral energy estimator for use in robust ASR. Traditionally, the spectral energy estimator has suffered from the problem of residual noise. In order to improve the SE estimator for use in robust ASR, the causes of noise under-suppression were investigated. These problems were then addressed with two heuristic adaptations. In the first method, a heuristic based speech presence uncertainty (SPU) modification was proposed. On its own, SPU can overcome the problems of the SE estimator at lower SNRs. However, this tends to introduce speech distortion at higher SNRs. To make the application of SPU more flexible, we introduced a simple heuristic noise-driven SPU.

In the second approach, a direct modification to the spectral amplitude gain function (of the spectral energy estimator) was proposed. This modification allowed the heuristic estimator to emulate several common spectral estimators, including

the spectral amplitude, log spectral amplitude and spectral Wiener estimators.

Experimental results for both heuristic methods indicate a significant improvement in robustness. The improvement gained by the modified spectral energy estimators is especially evident at the lower SNRs – regions where the standalone SE estimator typically struggled. For the proposed SE-SPU estimator, recognition accuracy was superior to the other common noise suppressors. This was true across all SNRs and all speech datasets. For the direct heuristic estimator, recognition accuracy closely matched the corresponding spectral amplitude and log-spectral amplitude estimators. While this included areas of deficiency (speech distortion in clean conditions), the proposed heuristic estimator has the advantage of being both simpler to implement and more efficient to run.

Chapter 9

In this Chapter, the stochastic estimation of the Mel-frequency cepstral coefficient feature set was investigated. The analysis started with a well known spectral domain noise distortion model. To make a suitable cepstral estimator, several mathematical transformations were detailed to transform statistical models between the spectral, filterbank, log-filterbank and cepstral domains. These transformations included the assumption of gamma distributed conditioned filterbank energies. The validity of this assumption was evaluated with several Monte-Carlo analyses and was shown to be very accurate under the most operating conditions.

With a cepstral noise distortion model defined, two estimators were proposed: 1) an estimator that used the cepstral model in conjunction with an *a priori* speech model, and 2) a stand-alone estimator. Experimental results show that both estimators greatly improved robustness compared with the baseline, vector Taylor series and spectral amplitude estimators. In general, the use of an *a priori* speech model did improve recognition accuracy, though this varied between recognition tasks. For the Aurora2 digit recognition task, the use of the prior speech model gave quite large recognition improvements. However, for the RM task, improvements

were more modest.

10.3.2 Future work

For our speech recognition tasks, we noticed that front-end enhancement algorithms behaved very differently depending on the particular recognition task. For example, the Aurora2 digit recognition task appeared to be very intolerant to speech distortion. This meant heavy noise suppressors (like the LSA estimator) performed badly. Conversely, the OLLO2 task tolerated much higher speech distortion, meaning heavier noise suppressors performed better. This variability also existed when using *a priori* speech models. While the peculiarities of each dataset become apparent after testing, it would be interesting to derive a more elegant method for measuring the impact of ASR topology (word versus phoneme based, dictionary size, continuous versus static recognition e.t.c) on front-end feature enhancement performance prior to actual deployment.

Another potential avenue for improvement over the current work is the extension of the estimation to dynamic (delta and acceleration) coefficients. This would involve extending the statistical framework again to include inter-frame correlations on top of the intra-frame correlations already studied.

Appendix A

Derivation of probability density functions

This appendix provides a step-by-step derivation of the spectral energy and log-filterbank energy PDFs used in Chapter 9. Under the assumed statistical framework given in section 9.2.1, the PDF $p(A(k))$ is given by a Rayleigh distribution

$$p(A(k)) = \frac{2A(k)}{\lambda_x(k)} \exp\left(-\frac{[A(k)]^2}{\lambda_x(k)}\right). \tag{A.1}$$

The Rayleigh distribution describes a variable $z = \sqrt{x_a^2 + x_b^2}$, where x_a and x_b are independent and identically distributed zero-mean Gaussian variables. For our purposes, it is used to describe the amplitudes of spectral variables (which were previously assumed Gaussian distributed along both the real and imaginary axis). The conditional PDF $p(Y(k)|A(k), \theta(k))$ is given as

$$\begin{aligned} p(Y(k)|A(k), \theta(k)) &= \frac{1}{\pi\lambda_d(k)} \exp\left(-\frac{|D(k)|^2}{\lambda_d(k)}\right) \\ &= \frac{1}{\pi\lambda_d(k)} \exp\left(-\frac{|Y(k) - A(k)\, \exp\left(j\theta(k)\right)|^2}{\lambda_d(k)}\right), \end{aligned} \tag{A.2}$$

where $\theta(k)$ is the spectral phase of clean spectral value $X(k)$. Here we have assumed that spectral value $Y(k)$ is only related to $A(k)$ and not any other spectral bins. If

we additionally assume $\theta(k)$ to be uniformly distributed over the $[-\pi, \pi]$ interval, it may be integrated out of (A.2) to give [61]:{3.339}

$$\begin{aligned} p(Y(k)|A(k)) &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1}{\pi\lambda_d(k)}\exp\left(-\frac{|Y(k)-A(k)\exp\left[j\theta(k)\right]|^2}{\lambda_d(k)}\right)d\theta(k) \\ &= \frac{1}{2\pi^2\lambda_d(k)}\exp\left(\frac{-|Y(k)|^2-[A(k)]^2}{\lambda_d(k)}\right). \\ &\qquad \int_{-\pi}^{\pi}\exp\left(\frac{2|Y(k)|A(k)\,\cos\,\theta(k)}{\lambda_d(k)}\right)d\theta(k) \\ &= \frac{1}{\pi\lambda_d(k)}\exp\left(\frac{-|Y(k)|^2-[A(k)]^2}{\lambda_d(k)}\right)I_0\left(\frac{2|Y(k)|A(k)}{\lambda_d(k)}\right) \\ &= \frac{1}{\pi\lambda_d(k)}\exp\left(\frac{-|Y(k)|^2-[A(k)]^2}{\lambda_d(k)}\right)I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}A(k)\right), \end{aligned} \tag{A.3}$$

where $I_0(.)$ is the zeroth order modified Bessel function. Using Bayes rule, (A.1) and (A.3) can be combined to give the conditioned spectral amplitude PDF $p(A(k)|Y(k))$

$$\begin{aligned} p(A(k)|\boldsymbol{Y}) &= \frac{p(A(k))p(Y(k)|A(k))}{\int_0^\infty p(A(k))p(Y(k)|A(k))dA(k)} \\ &= \frac{A(k)\exp\left(\frac{-[A(k)]^2}{\lambda_x(k)+\lambda_d(k)}\right)I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}A(k)\right)}{\int_0^\infty \tau\exp\left(\frac{-\tau^2}{\lambda_x(k)+\lambda_d(k)}\right)I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}\tau\right)d\tau} \\ &= \frac{A(k)\exp\left(\frac{-[A(k)]^2}{\lambda(k)}\right)I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}A(k)\right)}{\int_0^\infty \tau\exp\left(\frac{-\tau^2}{\lambda(k)}\right)I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}\tau\right)d\tau}. \end{aligned} \tag{A.4}$$

Original derivation of the spectral amplitude estimator and its corresponding PDF can be found in [39]. We may derive the spectral energy PDF with a few additional algebraic manipulations. Firstly, the integral in the denominator of (A.4) can be

solved[1] [61]:{6.631-7,8.406-1,8.464-1,8.464-2,9.210-1},

$$p(A(k)|\boldsymbol{Y}) = \frac{2A(k)\exp\left(\frac{-[A(k)]^2}{\lambda(k)}\right) I_0\left(2\sqrt{\frac{\nu(k)}{\lambda(k)}}A(k)\right)}{\lambda(k)\ \exp(\nu(k))}. \tag{A.5}$$

There is a one to one mapping between $e_x(k)$ and $A(k)$ over the $[0,\infty]$ interval. If we equate the cumulative density functions (CDFs) for each variable, then differentiate both w.r.t $e_x(k)$, we get

$$\begin{aligned} p(e_x(k)|\boldsymbol{Y}) &= p(A(k)|\boldsymbol{Y}).\left|\frac{dA(k)}{de_x(k)}\right| \\ &= \frac{p(A(k)|\boldsymbol{Y})}{2\sqrt{e_x(k)}}. \end{aligned} \tag{A.6}$$

Substituting (A.5) into (A.6) and using the substitution $A(k) = \sqrt{e_x(k)}$ yields the conditioned spectral energy PDF

$$p(e_x(k)|Y(k)) = \frac{\exp\left(\frac{-e_x(k)}{\lambda(k)}\right)\ I_0\left(2\sqrt{\frac{\nu(k)e_x(k)}{\lambda(k)}}\right)}{\lambda(k)\exp\big(\nu(k)\big)}. \tag{A.7}$$

A similar approach may be used for converting the filterbank energy PDF (9.22) to a log-filterbank energy PDF (9.25). The main difference is that the logarithm (in comparison to the squaring operator) is a one to one mapping from the $[0,\infty]$ to $[-\infty,\infty]$ intervals. Assuming a gamma PDF for the conditioned filterbank energy variable, the PDF for the conditioned log-filterbank energies can be given as

$$\begin{aligned} p(L_x(q)|\boldsymbol{Y}) &= p(E_x(q)|\boldsymbol{Y}).\left|\frac{dE_x(q)}{dL_x(q)}\right| \\ &= \frac{[E_x(q)]^{\alpha_q-1}\ \exp\left(-\frac{E_x(q)}{\beta_q}\right)}{\beta_q^{\alpha_q}\Gamma(\alpha_q)}.\exp(L_x(q)) \\ &= \frac{\exp\big(\alpha_q\left[L_x(q)-\log\beta_q\right]-\exp\left[L_x(q)-\log\beta_q\right]\big)}{\Gamma(\alpha_q)}. \end{aligned} \tag{A.8}$$

[1] Detail of a similar integration is given in appendix B.

Appendix B

Derivation of spectral and log-filterbank energy estimates

This appendix provides a step by step derivation of the spectral energy and log-filterbank energy estimates (both MMSE and MAP) used in Chapter 9. Given the conditioned spectral energy PDF $p(e_x(k)|\boldsymbol{Y})$ (9.11), the first raw moment (mean) of spectral energy is given by

$$\begin{aligned} E[e_x(k)|\boldsymbol{Y}] &= \hat{e}_x(k) \\ &= \int_0^\infty e_x(k).p(e_x(k)|\boldsymbol{Y})de_x(k) \\ &= \frac{\int_0^\infty e_x(k)\exp\left(\frac{-e_x(k)}{\lambda(k)}\right) I_0\left(2\sqrt{\frac{\nu(k)e_x(k)}{\lambda(k)}}\right) de_x(k)}{\lambda(k)\exp\big(\nu(k)\big)}. \end{aligned} \tag{B.1}$$

The above equation may be solved and simplified with [61]:{6.643-1,9.220-2,9.212-1,9.210-1},

$$\begin{aligned}
\hat{e}_x(k) &= \frac{[\nu(k)]^{-0.5}.[\lambda(k)]^2 \, \exp\left(\frac{\nu(k)}{2}\right) M_{-1.5,0}\left(\nu(k)\right)}{\lambda(k) \, \exp\left(\nu(k)\right)} \\
&= [\nu(k)]^{-0.5}.\lambda(k) \, \exp\left(-\frac{\nu(k)}{2}\right).M_{-1.5,0}\left(\nu(k)\right) \\
&= [\nu(k)]^{-0.5}.\lambda(k) \, \exp\left(-\frac{\nu(k)}{2}\right).[\nu(k)]^{0.5} \, \exp\left(-\frac{\nu(k)}{2}\right) \Phi_{2,1}\left(\nu(k)\right) \\
&= [\nu(k)]^{-0.5}.\lambda(k) \, \exp\left(-\frac{\nu(k)}{2}\right).[\nu(k)]^{0.5} \, \exp\left(\frac{\nu(k)}{2}\right) \Phi_{-1,1}\left(-\nu(k)\right) \\
&= [\nu(k)]^{-0.5}.\lambda(k) \, \exp\left(-\frac{\nu(k)}{2}\right).[\nu(k)]^{0.5} \, \exp\left(\frac{\nu(k)}{2}\right) \left(1+\nu(k)\right) \\
&= \lambda(k)\left(1+\nu(k)\right),
\end{aligned} \tag{B.2}$$

where, $M(.)$ is the Whittaker function and $\Phi(.)$ is the confluent hypergeometric function. The second central moment (variance) of the spectral energy is given as

$$\begin{aligned}
E\left[\left(e_x(k)-\hat{e}_x(k)\right)^2 |\boldsymbol{Y}\right] &= \Sigma_{e_x}(k,k) \\
&= \frac{\int_0^\infty [e_x(k)]^2 \exp\left(\frac{-e_x(k)}{\lambda(k)}\right) \, I_0\left(2\sqrt{\frac{\nu(k)e_x(k)}{\lambda(k)}}\right) de_x(k)}{\lambda(k)\exp\big(\nu(k)\big)} - [\hat{e}_x(k)]^2.
\end{aligned} \tag{B.3}$$

The above equation can be solved in a similar manner to (B.1) using [61]:{6.643-1,9.220-2,9.212-1,2.9210-1},

$$\begin{aligned}
\Sigma_{e_x}(k,k) &= \frac{2[\nu(k)]^{-0.5}.[\lambda(k)]^3 \, \exp\left(\frac{\nu(k)}{2}\right) M_{-2.5,0}\left(\nu(k)\right)}{\lambda(k) \, \exp\left(\nu(k)\right)} - \left[\lambda(k)\left(1+\nu(k)\right)\right]^2 \\
&= [2\nu(k)]^{-0.5}.[\lambda(k)]^2 \, \exp\left(-\frac{\nu(k)}{2}\right).[\nu(k)]^{0.5} \, \exp\left(\frac{-\nu(k)}{2}\right) \Phi_{3,1}\left(\nu(k)\right) - \left[\lambda(k)\left(1+\nu(k)\right)\right]^2 \\
&= [2\nu(k)]^{-0.5}.[\lambda(k)]^2 \, \exp\left(-\frac{\nu(k)}{2}\right).[\nu(k)]^{0.5} \, \exp\left(\frac{\nu(k)}{2}\right) \Phi_{-2,1}\left(-\nu(k)\right) - \left[\lambda(k)\left(1+\nu(k)\right)\right]^2 \\
&= 2[\lambda(k)]^2\left(1+2\nu(k)+\frac{[\nu(k)]^2}{2}\right) - \left[\lambda(k)\left(1+\nu(k)\right)\right]^2 \\
&= [\lambda(k)]^2\left(1+2\nu(k)\right).
\end{aligned} \tag{B.4}$$

Given the conditioned filterbank energy PDF $p(E_x(q)|\boldsymbol{Y})$ (9.22), the first raw moment (mean) of the log-filterbank energy is given by [61]:{4.352-1},

$$\begin{aligned}
E[L_x(q)|\boldsymbol{Y}] &= \hat{L}_x(q) \\
&= \int_0^\infty \log E_x(q).p(E_x(q)|\boldsymbol{Y})dE_x(q) \\
&= \int_0^\infty \log E_x(q).\frac{[E_x(q)]^{\alpha_q-1}\,\exp\left(-\frac{E_x(q)}{\beta_q}\right)}{\beta_q^{\alpha_q}\Gamma(\alpha_q)}dE_x(q) \\
&= \frac{\beta_q^{\alpha_q}\Gamma(\alpha_q)}{\beta_q^{\alpha_q}\Gamma(\alpha_q)}\left(\Psi_0(\alpha_q) - \log\left(\frac{1}{\beta_q}\right)\right) \\
&= \log(\alpha_q\beta_q) - \log(\alpha_q) + \Psi_0(\alpha_q) \\
&= \log\hat{E}_x(q) - \left[\log(\alpha_q) - \Psi_0(\alpha_q)\right].
\end{aligned} \tag{B.5}$$

The second central moment (variance) of the log-filterbank energy is given as [61]:{4.358-2,8.363-8},

$$\begin{aligned}
&E\left[[L_x(q) - \hat{L}_x(q)]^2|\boldsymbol{Y}\right] \\
&\quad= \Sigma_{L_x}(q,q) \\
&\quad= \int_0^\infty [\log E_x(q)]^2.p(E_x(q)|\boldsymbol{Y})dE_x(q) - [\hat{L}_x(q)]^2 \\
&\quad= \frac{\beta_q^{\alpha_q}\Gamma(\alpha_q)}{\beta_q^{\alpha_q}\Gamma(\alpha_q)}\left(\left[\Psi_0(\alpha_q) - \log\left(\frac{1}{\beta_q}\right)\right]^2 + \zeta(2,\alpha_q)\right) - \left[\Psi_0(\alpha_q) - \log\left(\frac{1}{\beta_q}\right)\right]^2 \\
&\quad= \Psi_1(\alpha_q),
\end{aligned} \tag{B.6}$$

where $\Psi_0(.)$, $\Psi_1(.)$ and $\zeta(.)$ are the digamma, trigamma and Riemann zeta functions respectively.

To find the MAP log-filterbank estimate, we are interested in finding the value of $L(q)$ that maximizes $p(L(q)|\boldsymbol{Y})$; i.e. the location of the PDF peak

$$\begin{aligned}\hat{L}_{q-MAP} &= \underset{L(q)}{\operatorname{argmax}}\,[p(L(q)|\boldsymbol{Y})] \\ &= \underset{L(q)}{\operatorname{argmax}}\,[\log p(L(q)|\boldsymbol{Y})] \\ &= \underset{L(q)}{\operatorname{argmax}}\, f(L(q)),\end{aligned} \tag{B.7}$$

where

$$f(L(q)) = [\alpha_q(L(q) - \log\beta_q) - \ \exp(L(q) - \log\beta_q)]\,. \tag{B.8}$$

To find the maxima, we first find the derivative of (B.8) w.r.t $L(q)$,

$$\begin{aligned}\frac{d}{dL(q)}\Big(\alpha_q(L(q) - \log\beta_q) - \ \exp(L(q) - \log\beta_q)\Big).dL(q) \\ = \alpha_q - \ \exp(L(q) - \log\beta_q).\end{aligned} \tag{B.9}$$

then, setting the derivative (B.9) at $L(q) = L_{q-MAP}$ to zero

$$\begin{aligned}&\alpha_q - \ \exp(L_{q-MAP} - \log\beta_q) = 0 \\ &L_{q-MAP} = \log\alpha_q + \log\beta_q \\ &L_{q-MAP} = \log\hat{E}_q.\end{aligned} \tag{B.10}$$

Appendix C

Derivation of lower limit for filterbank energy parameter α

In this section we show that the filterbank shape parameter α_q cannot take values below 1; or more precisely, that it exists on the interval $1 \leq \alpha_q \leq \infty$, when used to model filterbank energies under the assumed noise model. Context for this problem can be found in Chapter 9, Section 9.3. The definition of filterbank shape α_q is given as

$$\alpha_q = \frac{[\hat{E}_x(q)]^2}{\Sigma_{E_x}(q,q)}, \tag{C.1}$$

where filterbank mean $\hat{E}_x(q)$ is given as

$$\hat{E}_x(q) = \sum_k H(k,q)\hat{e}_x(k), \tag{C.2}$$

and filterbank variance is given

$$\Sigma_{E_x}(q,q) = \sum_k [H(k,q)]^2 \Sigma_{e_x}(k,k), \tag{C.3}$$

where $H(k,q)$ is the filterbank gain for the k'th frequency bin and q'th filterbank. The terms $\hat{e}_x(k)$ and $\Sigma_{e_x}(k,k)$ are the estimates for spectral energies and variances respectively (see Section 9.2.1). In order to proceed, we assume clean speech filterbanks have non-zero

energy; i.e., $\hat{e}_x(k) > 0$. Substituting (C.2) and (C.3) into (C.1), we have

$$\begin{aligned}\alpha_q &= \frac{\left[\sum_k H(k,q)\hat{e}_x(k)\right]^2}{\sum_k [H(k,q)]^2 \Sigma_{e_x}(k,k)} \\ &= \frac{\sum_k [H(k,q)]^2 [\hat{e}_x(k)]^2 + \sum_k \sum_{\substack{i \\ i \neq k}} H(k,q)H(i,q)\hat{e}_x(k)\hat{e}_x(i)}{\sum_k [H(k,q)]^2 \Sigma_{e_x}(k,k)}\end{aligned} \tag{C.4}$$

The definition of spectral variance is given as

$$\Sigma_{e_x}(k,k) = [\hat{e}_x(k)]^2 - \left(\frac{\xi(k)}{1+\xi(k)}\right)^4 [e_y(k)]^2. \tag{C.5}$$

Substituting (C.5) into (C.4), gives

$$\begin{aligned}\alpha_q &= \frac{\sum_k [H(k,q)]^2 [\hat{e}_x(k)]^2 + \sum_k \sum_{\substack{i \\ i \neq k}} H(k,q)H(i,q)\hat{e}_x(k)\hat{e}_x(i)}{\sum_k [H(k,q)]^2 \hat{e}_x(k)]^2 - \sum_k [H(k,q)]^2 \left(\frac{\xi(k)}{1+\xi(k)}\right)^4 [e_y(k)]^2} \\ &= \frac{T_1(q) + T_2(q)}{T_1(q) - T_3(q)},\end{aligned} \tag{C.6}$$

where terms

$$T_1(q) = \sum_k [H(k,q)]^2 [\hat{e}_x(k)]^2, \tag{C.7}$$

$$T_2(q) = \sum_k \sum_{\substack{i \\ i \neq k}} H(k,q)H(i,q)\hat{e}_x(k)\hat{e}_x(i), \tag{C.8}$$

$$T_3(q) = \sum_k [H(k,q)]^2 \left(\frac{\xi(k)}{1+\xi(k)}\right)^4 [e_y(k)]^2. \tag{C.9}$$

We first note that terms $T_1(q)$, $T_2(q)$ and $T_3(q)$ are non-negative. This can be reasoned by using the fact that $\xi(k)$, $\gamma(k)$, $H(k,q)$, $e_y(k)$ and $\hat{e}_x(k)$ are all non-negative. As a result, the numerator of (C.6) is greater than, or equal to the denominator of (C.6). Secondly, we note that the denominator of (C.6) is also non-negative. This is because the denominator is the filterbank variance – which again is strictly non-negative. From both of these observations, it can be inferred that the value of α_q cannot fall below 1. As the filterbank variance (denominator of (C.6)) tends toward zero, α_q tends toward infinity.

Appendix D

MATLAB code listing

Listing D.1: MATLAB code for phase spectrum retrieval

```
function PH = rgen_phase(MAG, FL, ITERS)
%  PH = rgen_phase(MAG, FL, ITERS)
%     Regeneration of phase spectrum for a given magnitude spectrum.
%     Written by Anthony Stark Nov 2009
%
%   Inputs:
%       MAG - magnitude spectrum values (vector)
%       FL - length of time-domain sequence that is to produce the given
%         magnitude spectrum.
%       ITERS - number of iterations to run the algorithm for (def = 200)
%   Outputs:
%       PH - returned phase spectrum, is of same dimensions as MAG

if nargin < 3    ITERS = 200;   end

MS = reshape([ones(FL,1);zeros(length(MAG)-FL,1)], size(MAG))-1;
GAM = 1.9*i;

% Random anti-symmetric phase bootstrap
PH = fft( 2*pi*rand(size(MAG)) );
PH = PH ./ abs(PH);
XG = MAG.*PH;

% Iterative regeneration of the phase spectrum
for z = 1:ITERS
    %XG = fft( (1-MS).*ifft(XG) ); % GS t-domain update
    XG = XG + GAM * PH .* imag( conj(PH) .* fft( MS.*ifft(XG) ) ) ;
    PH = XG ./ (abs(XG)+eps);
    XG = MAG.*PH;
end
```

Listing D.2: MATLAB code for phase spectrum compensation speech enhancement

```
function xsynth = psc(wavfile, WIN, FD, NFFT, NF)
%  xsynth = PSC(wavfile, WIN, FD, NFFT, NF)
%    Phase spectrum compensation speech enhancement method
%    Written by Anthony Stark Jan 2010
%
%   Inputs:
%       wavfile: the name of the wave file to be enhanced
%       WIN: [Nx1] vector describing the analysis window
%       FD: The difference (in samples) between adjacent frames
%       NFFT: FFT length (must be > N)
%       NF: Number of initial frames used to estimate noise (def = 10)
%
%   Outputs:
%       xsynth: vector containing the samples of the enhanced signal
%

if nargin < 5
   NF = 10;
end

FL = length(WIN);
x = wavread(wavfile);
lambda_val = 3.74;    % Strength of suppression

% Build anti-symmetry function
as_func = ones(NFFT,1);
as_func(NFFT/2+2:end) = -1;
as_func([1,NFFT/2+1]) = 0;

% Frame up the signal and perform STFT analysis
%  FC = frame count, FIDX = frame indicies matrix
FC = 2 + floor( (length(x) - FL) / FD );
FIDX = repmat(1:FL, FC, 1)' + repmat( (0:FC-1)'*FD, 1, FL)';
xpad = [x; zeros(FIDX(end)-length(x),1)];
X_seg = fft( diag(WIN) * xpad(FIDX), NFFT, 1 );

% Very basic noise amplitude estimation
noise_est = sqrt(mean( abs(X_seg(:,1:NF)).^2, 2 ));

% Find the modified phase spectrum and combine with old magnitude
lambda_func = lambda_val * as_func .* noise_est;
new_phase = X_seg + repmat(lambda_func,1,FC);
new_phase = new_phase ./ (abs(new_phase) + eps);

XM_seg = abs(X_seg) .* new_phase;
xm_seg = ifft( XM_seg, NFFT, 1 );
xm_seg = real( xm_seg(1:FL, :) );

% Synthesize the new signal
```

```
fix_term = zeros(size(xpad));
for m=1:FC
   fix_term( FIDX(:,m) ) = fix_term( FIDX(:,m) ) + WIN;
end

xsynth = zeros(size(xpad));
for m=1:FC
    xsynth( FIDX(:,m) ) = xsynth( FIDX(:,m) ) + xm_seg(:,m);
end

% Clean up synthesized signal
xsynth = xsynth ./ fix_term;
xsynth(isinf(xsynth) | isnan(xsynth)) = 0;
xsynth = xsynth(1:length(x));
```

Listing D.3: MATLAB code for IF deviation spectrogram

```
function [psd ifsd ifo] = ifogram(x, WIND, FD, NFFT, lim)
% IFOGRAM(x, wind, fd, nfft, lim)
%   Generates an ifd based plot and corresponding spectrogram
%  ---== Written by Anthony Stark 2008 ==---
%
% Return args:
%   psd:     spectrogram plot normed to 0dB
%   ifsd:    IF spectral plot
%   ifo:     raw IF values
%
% Input args:
%   x:       analysis signal
%   wind:    analysis window
%   fd:      sample delay between consecutive analysis frames
%   nfft:    size of the fft
%   lim:     [1x3] vector, lim(1)=l-bound for IFSD, lim(2)=u-bound for IFSD
%            lim(3)=l-bound for PSD + norm for IFSD.
%            Use [20 60 -50] for default

FL = length(WIND);
S1 = spectrogram(x(1:end-1), WIND, FL-FD, NFFT);
S2 = spectrogram(x(2:end-0), WIND, FL-FD, NFFT);
sz = size(S1);

%-------------------------------------------------------------------------
% Standard PSD periodogram
psd = 20*log10(abs(S1));
psd = psd - max(max(psd));

% Limit the PSD plot (normed to zero)
psd = psd - max(psd(:));
psd(psd<lim(3)) = lim(3);

%-------------------------------------------------------------------------
% IFD spectrogram
offset = exp( -j*repmat((0:NFFT/2)'/NFFT*2*pi, 1, sz(2) ));
ifo = angle(S2 .* conj(S1) .*  offset);

% Force into -pi +pi range (matlab does 0-2pi)
ifo = mod(ifo+pi, 2*pi) - pi;

% Convert from IFD to magnitude
ifpow = 1 ./ (abs(ifo/pi).^2 + eps);

% Create log-IFD plot, and smooth
ifsd = 10*log10(ifpow);
ifsd = conv2( ones(5,1)/5, ifsd); %freq axis smoothing
ifsd = ifsd( 3:end-2, : );
```

```
% Perform limits on Log_IFD plot
ifsd(ifsd>lim(2)) = lim(2);
ifsd(ifsd<lim(1)) = lim(1);
ifsd = ifsd - max(ifsd(:));
ifsd = ifsd / min(ifsd(:)) * lim(3);
```

Listing D.4: MATLAB code for GD deviation spectrogram

```
function GDD = gdgram(x, wind, FD, NFFT, dynrange)
% GDGRAM(x, wind, FD, nfft, dynrange)
%   Generates an gdd based spectrogram
%   ---== Written by Anthony Stark 2008 ==---
%
% Return args:
%   GDD:     matrix of IAGDD spectrogram
%
% Input args:
%   x:           analysis signal
%   wind:        analysis window
%   FD:          sample delay between consecutive analysis frames
%   nfft:        size of the fft
%   dyngrange:   [2x1] min and max vals allowed by the log-GDD

FL = length(wind);
rind = wind .* (0:FL-1)'; % Ramped analysis window

FC = 1 + floor( (length(x)-FL) / FD );
idx = repmat(1:FL, FC, 1) + repmat( (0:FC-1)'*FD, 1, FL);

% STFT spectrums using standard and ramped window functions
A = fft( x(idx) * diag(wind), NFFT, 2);
AR = fft( x(idx) * diag(rind), NFFT, 2);

% Find the group delay deviation
GDD = abs(real(AR./A) - (FL-1)/2 );
GDD = conv2( ones(1,5)/5, GDD); % freq axis smoothing
GDD = GDD( :, 3:end-2 );

% Convert into log domain, and truncate ranges
GDD = -log10(GDD(:,1:NFFT/2+1))';
GDD(GDD<dynrange(1)) = dynrange(1);
GDD(GDD>dynrange(2)) = dynrange(2);
```

Listing D.5: MATLAB code for heuristic SPU spectral energy estimation

```
function [x ceps] = sespu(filename)
% function sespu(filename, mfcname, type, spu)
%    Takes a NIST waveform and writes an enhanced HTK format MFCC_E file
%       x - cleaned speech vector
%       ceps - cepstral coeffs from x

%=========================================================================
% Must load an ensemble spectral energy average into the variable T_seg
% Must be size [NFFT/2+1 x 1] and be properly normed
load TSEG.mat;

%=========================================================================
% Tuneables
regen = 1;          % Force time-domain resynthesis
FS = 16000;         % Sample freq.
flen = 400;         % Frame length
NFFT = 800;         % Discrete Fourier transform size
fshift = 160;       % Frame shift
BANDS = 26;         % # filterbanks
CEPNUM = 12;        % # cepstral coeffs to keep
CEPLIFT = 22;       % liftering coefficient
kappa = 0.5;        % SPU heuristic scale (lower = aggressive)
qmax = 0.4;         % Max SPU aggressivenes (higher = aggressive)
win = hamming(flen);
alph = 0.98;        % Xi estimation parm (higher = more smoothing)
eta  = 0.15;        % Loizous VAD detection theshold
mu   = 0.98;        % Loizous VAD mixer parameter
nframes = 11;       % Number of frames to use for lamd est
xik_min = 10^-3; % Bounding for SNR estimation
gammak_max = 50;

%=========================================================================
fid = fopen(filename,'r','b'); % This is for Aurora2 files
x = fread(fid, 'int16');
fclose(fid);
x = x - mean(x);
fcount = 1 + floor( (length(x) - flen) / fshift );
fidx = repmat(1:flen, fcount, 1)' + repmat( (0:fcount-1)'*fshift, 1, flen)';

 % Basic energy normalization
segs = diag(win) * x(fidx);
max_eng = max(sum(segs.^2));
segs = segs / sqrt(max_eng);

% Corrupt speech DSTFT
X_segs = fft( segs, NFFT, 1 );
X_segs = X_segs(1:NFFT/2+1,:);

lamd = mean( abs(X_segs(:,1:nframes)), 2 ).^2;
```

```
prev_pow = zeros(NFFT/2+1,1);
qk = zeros(NFFT/2+1,1);


for n=1:fcount
    % Estimation of model parameters
    pow     = abs(X_segs(:,n)).^2;
    gammak = min(pow./lamd, gammak_max);
    xik     = max(alph*prev_pow./lamd + (1-alph)*max(gammak-1,0), xik_min);
    nuk     = xik .* gammak ./ (1 + xik);

    % ==== Loizou's VAD code
    log_sigma_k = gammak.* xik./ (1+ xik)- log(1+ xik);
    vad_decision = sum( log_sigma_k) / flen;
    if (vad_decision< eta)
        lamd = mu*lamd + (1-mu) * pow;
    end

    % Calculate the heuristic SPU
    qrat = kappa*(T_seg ./ lamd).^0.5;
    qk = 1 - qrat ./ (1 + qrat);
    qk(qk>qmax) = qmax;
    qrat = (1-qk)./qk;
    SPUrat = qrat .* exp(nuk) ./ (1 + xik);
    SPU = SPUrat ./ (1 + SPUrat);     % SPU standard form: p(H1)

    SPU(isinf(SPU)) = 1;
    SPU(SPU<0) = 0;

    % Apply enhancement
    gain = sqrt(nuk.*(1+nuk).*SPU)./gammak;
    X_segs(:,n) = X_segs(:,n) .* gain;
    prev_pow = abs(X_segs(:,n)).^2;
end

% Force time-domain resynthesis if desired
if regen == 1
    x = zeros(fidx(end), 1);
    X_segs(NFFT/2+2:NFFT,:) = flipud(conj( X_segs(2:NFFT/2,:) ));
    x_segs = ifft( X_segs );
    x_segs = real( x_segs(1:flen,:) );

    fix = zeros(size(x));
    for k=1:fcount
        fix( fidx(:,k) ) = fix( fidx(:,k) ) + win;
        x( fidx(:,k) ) = x( fidx(:,k) ) + x_segs(:,k);
    end

    x = x ./ fix;
    x(isnan(x)) = 0;
    x(isinf(x)) = 0;
```

```
    segs = diag(win) * x(fidx);
    max_eng = max(sum(segs.^2));
    segs = segs / sqrt(max_eng);

    X_segs = fft( segs, NFFT, 1 );
    X_segs = abs(X_segs(1:NFFT/2+1,:));
end

% ========================================================================
% Feature generation stage
nps = NFFT/2+1;
h_range = [0 0.5*FS]; % Start and end band centres (Hz)
h_scale = linspace(0,FS/2,nps)'; % Hertz scale axis for bands - 0Hz to 0.5 FS
m_range = 1125*log(1 + h_range./700);
m_delta = (m_range(2) - m_range(1)) / (BANDS+1); % Mel jump between filters
m_centre = (1:BANDS) * m_delta; % Centre frequencies (Mel)
h_centre = 700 * (exp(m_centre/1125) - 1);

l_slope = 1 ./ ( h_centre - [h_range(1) h_centre(1:end-1)] );
r_slope = 1 ./ ( h_centre - [h_centre(2:end) h_range(2)] );

del = repmat(h_scale,1,BANDS) - repmat(h_centre,nps,1);
bands = (del<0) .* del .* repmat(l_slope, nps, 1) ...
      + (del>0) .* del .* repmat(r_slope, nps, 1) + 1;
bands(bands < 0) = 0;

bands = bands * diag( sum(bands,1).^-1 );


lifter = diag( 1 + CEPLIFT/2 * sin ( (1:CEPNUM)*pi/CEPLIFT ) );

X_segs = abs(X_segs).'.^2;       % Move to power spectrum
lbe = log(X_segs * bands);       % log band energy
total_eng = log(sum(X_segs,2)); % frame log energy
total_eng = total_eng - max(total_eng);

% Cepstral coefficients, apply any liftering and tack on a
% band energy measure
ceps = dct( lbe.').';
ceps = [ceps(:, 2:CEPNUM+1) * lifter,  total_eng];
```

Listing D.6: MATLAB code for reading a HTK HMM file

```
function model = loadhmm(filename)
% Loads a very basic text based HTK hmm file into a MATLAB
%  file format.  Does not load HMM specific parameters
%  like transition probs.
%
% Structure: model
%   model.labels
%   model.variances
%   model.means
%   model.mixweight
%
% Array dimensions are as follows:
%   (lab_id, state_id, mix_id, DIMS)


fid = fopen(filename);
lines = textscan(fid, '%s', 'Delimiter', '\n');
lines = lines{1};
fclose(fid);

model.labels = cell(1);
lab_id = 0;
flag = 0;

for k=1:length(lines)
    if strcmp( lines{k}(1:2), '~h' )
        lab_id = lab_id + 1;
        model.labels{lab_id} = lines{k}(5:end-1);
        continue;
    end

    if lab_id > 0 & findstr( lines{k}, '<VARIANCE>' )
       dim = sscanf( lines{k}(11:end), '%d' );
       model.variances(lab_id, state_id, mix_id, :) = sscanf( lines{k+1}, '%f' );
       continue;
    end

    if findstr( lines{k}, '<MEAN>' )
       dim = sscanf( lines{k}(7:end), '%d' );
       model.means(lab_id, state_id, mix_id, :) = sscanf( lines{k+1}, '%f' );
       continue;
    end

    if findstr( lines{k}, '<GCONST>' )
       model.gconst(lab_id, state_id, mix_id) = sscanf( lines{k}(9:end), '%f' );
       continue;
    end

    if findstr( lines{k}, '<NUMSTATES>' )
```

```
        states = sscanf( lines{k}(12:end), '%d' ) - 2;
        continue;
    end

    if findstr( lines{k}, '<STATE>' )
        state_id = sscanf( lines{k}(9:end), '%d' ) - 1;
        continue;
    end

    if findstr( lines{k}, '<NUMMIXES>' )
        mixcount = sscanf( lines{k}(11:end), '%d' );
        continue;
    end

    if findstr( lines{k}, '<MIXTURE>' )
        temp = sscanf( lines{k}(10:end), '%f', 2 );
        mix_id = temp(1);
        model.mixweight(lab_id, state_id, mix_id) = temp(2);
        continue;
    end
end
```

Listing D.7: MATLAB code for MMSE MFCC estimation

```
function [ceps psegs] = mfcc_mmse(filename, spu, prior)
% MFCC MMSE estimator implementation
%  INPUTS:
%    filename - input filename
%    spu - any static SPU to use (0 = none)
%    prior - speech prior if any, use function loadhmm
%      to load a compatible prior variable
%
%  OUTPUTS:
%    ceps = cepstral coefficient matrix
%    psegs = spectral energies matrix

%==========================================================================
% Tuneables
BANDS = 26;              % Number of Mel-bands
FS = 16000;              % Sampling frequency (Hz)
flen = 400;              % Frame length: #samples
fshift = 160;            % Frame shift: #samples
win = hamming(flen);     % Analysis window
NFFT = 800;              % FFT size
CEPNUM = 12;             % Number of cepstral coeffs
CEPLIFT = 22;            % Cepstral liftering

alph = 0.98;             % DD XI estimation parm
eta  = 0.15;             % Loizous VAD detection theshold
mu   = 0.98;             % Loizous VAD mixer parameter
nframes = 11;            % Number of frames to use for lamd est
xik_min = 10^-3;         % Minimum allowed XI


%==========================================================================
% NIST file format expected here
x = read_NIST_file(filename); % sub desired wavreading routine here
x = x - mean(x);
fcount = 1 + floor( (length(x) - flen) / fshift );
fidx = repmat(1:flen, fcount, 1)' + repmat( (0:fcount-1)'*fshift, 1, flen)';

segs = diag(win) * x(fidx);
max_eng = max(sum(segs.^2));
segs = segs / sqrt(max_eng);
X_segs = fft( segs, NFFT, 1 );
X_segs = X_segs(1:NFFT/2+1,:);

pvari = zeros(size(X_segs));
lamd = mean( abs(X_segs(:,1:nframes)), 2 ).^2;
prev_pow = zeros(NFFT/2+1,1);
qk = zeros(NFFT/2+1,1);
```

```
%=========================================================================
% Standard spectral energy estimator stage, with additional variance est
for n=1:fcount

    % Estimation of model parameters
    pow = abs(X_segs(:,n)).^2;
    gammak = pow./lamd;
    xik    = max(alph*prev_pow./lamd + (1-alph)*max(gammak-1,0), xik_min);

    % ==== Loizou's VAD code
    log_sigma_k = gammak.* xik./ (1+ xik)- log(1+ xik);
    vad_decision = sum( log_sigma_k) / flen;
    if (vad_decision< eta)
        lamd = mu*lamd + (1-mu) *pow;
    end
    % ===end of vad===

    qk = spu;
    qrat = (1-qk)./qk;
    nuk = xik .* gammak ./ (1 + xik);
    SPUrat = qrat .* exp(nuk) ./ (1 + xik);
    SPU = SPUrat ./ (1 + SPUrat);
    SPU(SPU>1|isnan(SPU)) = 1;
    SPU(SPU<eps) = eps;

    lamk =  pow .* nuk ./ gammak.^2;
    gain = sqrt(nuk.*(1+nuk).*SPU)./gammak;

    % Additional variance calculation
    newpow = gain.^2.*pow;
    newxi =  -(2*pow ./ ( lamd - sqrt(lamd.^2 + 4*pow.*newpow ) ) + 1).^-1;
    pvari(:,n) = newpow.^2 - ( newxi ./ (1+newxi) ).^4 .* pow.^2;

    X_segs(:,n) = X_segs(:,n) .* gain;
    prev_pow = abs(X_segs(:,n)).^2;
end


%=========================================================================
% Feature generation stage
psegs = abs(X_segs).'.^2;
bands = mel_bands(BANDS, FS, (NFFT/2)+1);
lifter = diag( 1 + CEPLIFT/2 * sin ( (1:CEPNUM)*pi/CEPLIFT ) );

% Filterbank level variables
be = (psegs * bands);              % Filterbank means
bevar = pvari' * (bands.^2) + eps; % Filterbank vars (diag)
alph = (be.^2 + 1e-5) ./ bevar;    % Alpha value for FBEs
alph(alph<1) = 1;                          % Fix alpha for precis loss

% Total fbe energy
```

```
total_eng = log(sum(psegs,2));
total_eng = total_eng - max(total_eng);

% log-filterbank variables
lbfix = 0.5./(alph+0.045) + 0.108./(alph+0.045).^2;  % MAP diff term
lbvar = 1./(alph-0.1157) + 0.404./(alph-0.1157).^2;  % LFBE variance
lbe = log(be) - lbfix;  % MMSE log-filterbank energy

% Cepstral coefficients, apply any liftering and tack on a
% band energy measure
ceps = dct( lbe.').';
ceps = [ceps(:, 2:CEPNUM+1) * lifter,  total_eng];

% Do same for cepstral variances
DMAT = dct(eye(BANDS));
cepvar = zeros(size(lbvar));
for n=1:fcount
    bevar = bands'*diag(pvari(:,n))*bands; % FB cov matrix
    dgvar = diag( be(n,:).^-1 );

    % First ord. estimate for log-FBE cov, replace diags with lbvar
    lbvarfull = dgvar*bevar*dgvar;
    lbvarfull = lbvarfull - diag(diag(lbvarfull)) + diag(lbvar(n,:));

    % Find cepstral cov, only keep diags
    cepvar(n,:) = diag(DMAT*lbvarfull*DMAT');
end
cepvar = [cepvar(:, 2:CEPNUM+1) * lifter.^2, repmat(1e-5, size(total_eng))];


%==========================================================================
% Mix with a prior speech model
[FC DIM] = size(ceps);
PMIX=16;
DIDX = 1:CEPNUM;
for frame=1:FC%*0          % TURN ON / OFF prior model mixing

    B = cepvar(frame, 1:CEPNUM)';
    BI = B.^-1; % Inverse data covar (diag covar needed!)
    BI(BI>1e6) = 1e6;
    b = ceps(frame, 1:CEPNUM)';

    wgts = zeros(PMIX,1); % Weights for each Gaussian product - 1 for each mix
    bits = zeros(length(DIDX), PMIX); % Means of each Guassian product
    wgtsum = -Inf;

    for k=1:PMIX
        A = squeeze(prior.variances(1,1,k,DIDX));
        AI = A.^-1; % Inverse prior covar (diag covar needed!)
        AI(AI>1e6) = 1e6;
        a = squeeze(prior.means(1,1,k,DIDX));
```

```
        % Covariance and mean of the Gaussian product
        C = (AI + BI).^-1;
        c = C.*(AI.*a + BI.*b);

        % Gaussian product normalization constant (with mix weighting)
        % Have to use log-liklihoods to avoid precision loss
        wgts(k) = log(prior.mixweight(1,1,k)) + lmvnpdf( a, b, diag(A+B) );
        bits(:,k) = c;
        tmp = sort([wgts(k) wgtsum], 'descend');
        wgtsum = tmp(1) + log(1 + exp(tmp(2)-tmp(1)));
    end

    ceps(frame, 1:CEPNUM) = bits*exp(wgts - wgtsum);
end
```

Bibliography

[1] ISO 226:2003. Acoustics-Normal equal-loudness level contours.

[2] T. Abe, T. Kobayashi, and S. Imai. Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '96*, volume 1, pages 1277–1280, May 1996.

[3] A. Acero, L. Deng, T. Kristjansson, and J. Wang. HMM adaptation using vector taylor series for noisy speech recognition. In *Proceedings Interspeech*, 2000.

[4] A. Acero and R.M. Stern. Environmental robustness in automatic speech recognition. In *Proceedings Interspeech*, pages 849–852, 1990.

[5] Y. Akiri, S. Mizuta, M. Nagata, and T. Sakai. Spoken word recognition using dynamic features analysed by two-dimensional cepstrum. In *Proceedings IEE communications, speech and vision*, pages 133–140, 1989.

[6] H. Al-Nashi. Phase unwrapping of digital signals. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1693–1702, Nov 1989.

[7] J.B. Allen. How do humans process and recognize speech? *Speech and Audio Processing, IEEE Transactions on*, 2(4):567–577, Oct 1994.

[8] J.B. Allen and L.R. Rabiner. A unified approach to short-time Fourier analysis and synthesis. *IEEE, Proceedings on*, 65(11):1558–1564, 1977.

[9] L. Alsteris. *Short-time phase spectrum in human and automatic speech recognition.* PhD in Electrical engineering, Griffith University, 2005.

[10] L. Alsteris and K.K. Paliwal. Further intelligibility results from human listening tests using the short-time phase spectrum. *Speech Communication*, 48(6):727–736, 2006.

[11] B. Atal and S. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50(2):637–655, 1971.

[12] J. Barker, L. Josifovski, M. Cooke, and P. Green. Soft decisions in missing data techniques for robust automatic speech recognition. In *Proceedings ICSLP*, pages 373–376, 2000.

[13] V. Beatie and S. Young. Hidden Markov model state-based cepstral noise compensation. In *Proceedings ICSLP*, pages 519–522, 1992.

[14] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 4, pages 208–211, 1979.

[15] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[16] B. Boashash. Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. *Proceedings of the IEEE*, 80(4):520–538, Apr 1992.

[17] S.F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech, Signal Processing, IEEE Transactions on*, ASSP-27(2):113–120, 1979.

[18] A.C. Bovik, P. Maragos, and T.F. Quatieri. AM-FM energy detection and separation in noise using multiband energy operators. *Signal Processing, IEEE Transactions on*, 41(12):3245–3265, Dec 1993.

[19] B. Bozkurt, L. Couvreur, and T. Dutoit. Chirp group delay analysis of speech signals. *Speech Communication*, 49(3):159–176, 2007.

[20] A. Buzo, A. Gray, R. Gray, and J. Markel. Speech coding based upon vector quantization. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 28(5):562–574, Oct 1980.

[21] O. Cappe. Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor. *Speech and Audio Processing, IEEE Transactions on*, 2(2):345 –349, Apr 1994.

[22] R. Carlson and G. Fant. *Two-formant models, pitch and vowel perception*. Academic press, 1975.

[23] F. Charpentier. Pitch detection using the short-term phase spectrum. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 11, pages 113–116, Apr 1986.

[24] D. Chazan, R. Hoory, G. Cohen, and M. Zibulski. Speech reconstruction from Mel frequency cepstral coefficients and pitch. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pages 1299–1302, 2000.

[25] E. Choi. On compensating the mel-frequency cepstral coefficients for noisy speech recognition. In *Proceedings of the 29th Australasian Computer Science Conference, ACSC*, pages 49–54, 2006.

[26] I. Cohen. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *Signal Processing Letters, IEEE*, 9(4):113–116, Apr 2002.

[27] M. Cooke, P. Green, L. Josifofski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech communication*, 34:267–285, 2000.

[28] M. Cooke, A. Morris, and P. Green. Missing data techniques for robust speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International conference on ICASSP*, page 863, 1997.

[29] R.E. Crochiere. A weighted overlap-add method of short-time Fourier analysis / synthesis. *Acoustics, Speech, Signal Processing, IEEE Transactions on*, ASSP-28(2):99–102, 1980.

[30] S.B. Davis and P. Mermelstein. *Readings in speech recognition: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[31] L. Deng, A. Acero, and X. Huang. Large vocabulary speech recognition under adverse acoustic environments. In *Proceedings ICSLP*, 2000.

[32] L. Deng, J. Droppo, and A. Acero. Estimating cepstrum of speech under the presence of noise using joint prior of static and dynamic features. *Speech and Audio Processing, IEEE Transactions on*, 12(3):218,233.

[33] L. Deng and D. Yu. Use of differential cepstra as acoustic features in hidden trajectory modeling for phonetic recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, 4:IV–445–IV–448, 15-20 April 2007.

[34] D.-V. Dimitriadis, P. Maragos, and A. Potamianos. Robust AM-FM features for speech recognition. *Signal Processing Letters, IEEE*, 12(9):621–624, Sep 2005.

[35] J. Droppo, A. Acero, and L. Deng. Uncertainty decoding with splice for noise robust speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 1, pages 57–60, 2002.

[36] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[37] G. Duncan, B. Yegnarayana, and H.A. Murthy. A nonparametric method of formant estimation using group delay spectra. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 1, pages 572–575, May 1989.

[38] Y. Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *Signal processing, IEEE transactions on*, 40:725–735, 1992.

[39] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(6):1109–1121, Dec 1984.

[40] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):443–445, Apr 1985.

[41] Y. Ephraim, D. Malah, and B. Juang. On the application of hidden Markov models for enhancing noisy speech. *Acoustics, speech and signal processing, IEEE transactions on*, 34:1846–1856, 1989.

[42] Y. Ephraim and H.L. Van Trees. Constrained iterative speech enhancement with application to speech recognition. *Signal processing, IEEE Transactions on*, 39(4):795–805, Apr 1991.

[43] Y. Ephraim and H.L. Van Trees. A signal subspace approach for speech enhancement. *Speech Audio Processing, IEEE Transactions on*, 3:251–266, July 1995.

[44] A. Erell and M. Weintraub. Energy conditioned spectral estimation for recognition of noisy speech. *Speech and Audio Processing, IEEE Transactions on*, 1(1):84 –89, Jan 1993.

[45] G. Evangelista and S. Cavaliere. Discrete frequency warped wavelets : Theory and applications. *Signal Processing, IEEE Transactions on. Special issue on Theory and Applications of Filter Banks and Wavelets.*, 1998.

[46] J.L. Flanagan and R.M. Golden. Phase vocoder. *Bell System Technical Journal*, 45:1493–1509, Nov 1966.

[47] H. Fletcher. *Speech and hearing in communication, ASA edition.* The acoustical society of America, 1995.

[48] D. Friedman. Formulation of a vector distance measure for the instantaneous-frequency distribution of speech. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 12, pages 1748–1751, Apr 1987.

[49] D. Friedman. Instantaneous-frequency distribution vs. time: An interpretation of

the phase structure of speech. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, 10:1121–1124, Apr 1985.

[50] M. Fujimoto and Y. Ariki. Noisy speech recognition using noise reduction method based on Kalman filter. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 3:1727–1730, 2000.

[51] D. Gabor. Theory of communication. *IEE*, 93:429–457, 1946.

[52] B. Gajic and K.K. Paliwal. Robust speech recognition in noisy environments based on subband spectral centroid histograms. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):600–608, Mar 2006.

[53] L.F.J. Gales. *Model-based techniques for robust speech recognition*. PhD thesis, University of Cambridge, UK, 1995.

[54] A. Garnero, B. Drygajlo. Perceptual speech coding using time and frequency masking constraints. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, 1997.

[55] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, and D.S. Pallett. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93:27403–+, 1993.

[56] J-L. Gauvain and C-H Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *Audio Processing, IEEE Transactions on*, 2:291–298, 1994.

[57] R. Gemello, F. Mana, and R.De Mori. Automatic speech recognition with a modified Ephraim-Malah rule. *IEEE Signal processing letters*, 13(1):56–59, Jan 2006.

[58] M.J.T. George, E.B. Smith. Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model. *Speech and Audio Processing, IEEE Transactions on*, 5(5):389–406, 1997.

[59] O. Ghitza. Auditory nerve representation criteria for speech analysis/synthesis. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(6):736–740, Jun 1987.

[60] O. Ghitza. Auditory models and human performance in tasks related to speech coding and speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 2(1):115–132, Jan 1994.

[61] I.S. Gradshteyn and I.M. Ryzhik. *Table of Integrals, Series and Products*. Elsevier, 2007.

[62] D.W. Griffin and J.S. Lim. Signal estimation from modified short-time Fourier transform. *Acoustics, Speech, Signal Processing, IEEE Transactions on*, ASSP-32(2):236–243, 1984.

[63] M. Grimaldi and F. Cummins. Speaker identification using instantaneous frequencies. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(6):1097–1111, Aug 2008.

[64] F.J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of IEEE*, 66(1):51–83, 1978.

[65] M. Hayes, J Lim, and A. Oppenheim. Phase-only signal reconstruction. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 5, pages 437–440, Apr 1980.

[66] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Acoustical Society of America Journal*, 87:1738–1752, April 1990.

[67] K. Hermus, P. Wambacq, and H. V. Hamme. A review of signal subspace speech enhancement and its application to noise robust speech recognition. *EURASIP J.Appl.Signal Process.*, 2007(1):195–197, 2007.

[68] P.V. Hove, M. Hayes, J. Lim, and A. Oppenheim. Signal reconstruction from signed Fourier transform magnitude. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 31(5):1286–1293, Oct 1983.

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

[70] K.M. Indrebo, R.J. Povinelli, and M.T. Johnson. Minimum mean-squared error estimation of mel-frequency cepstral coefficients using a novel distortion model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(8):1654–1661, Nov. 2008.

[71] F. Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 23(1):67–72, Feb 1975.

[72] L. Josifovski, M. Cooke, P. Green, and A. Vizinho. State based imputation of missing data for robust speech recognition and speech enhancement. In *Proceedings Eurospeech*, pages 2837–2840, 1999.

[73] S. Kay. A fast and accurate single frequency estimator. *Acoustics Speech and Signal Processing, IEEE Transactions on*, 37(12):1987–1990, Dec 1989.

[74] B. Kedem. On frequency detection by zero-crossings. *Signal Processing*, 10(3):303–306, 1986.

[75] D-S. Kim, S-Y. Lee, and R.M. Kil. Auditory processing of speech signals for robust speech recognition in real-world noisy environments. *Speech and Audio Processing, IEEE Transactions on*, 7(1):55–69, Jan 1999.

[76] Y. Kubo, S. Okawa, A. Kurematsu, and K. Shirai. A study on temporal features derived by analytic signal. In *Interspeech Antwerp*, pages 1130–1133, 2007.

[77] G. Lathoud, M. Magimai-Doss, B. Mesot, and H. Bourlard. Unsupervised Spectral Subtraction for Noise-Robust ASR. In *Proceedings of the 2005 IEEE ASRU Workshop*, pages 343–348, 2005.

[78] A.V. Lim, J.S. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67(12):1586–1604, Dec. 1979.

[79] L. Liu, J. He, and G. Palm. Effects of phase on the perception of intervocalic stop consonants. *Speech Communication*, 22(4):403–417, 1997.

[80] P. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[81] J. Makhoul. Spectral analysis of speech by linear prediction. *Audio and Electroacoustics, IEEE Transactions on*, 21(3):140–148, Jun 1973.

[82] D. Malah, R. V. Cox, and A. J. Accardi. Tracking speech-presence uncertainty to improve speech enhancement in non-stationary noise environments. In *Proceedings of the Acoustics, Speech, and Signal Processing, IEEE International Conference ICASSP*, pages 789–792, Washington, DC, USA, 1999. IEEE Computer Society.

[83] L. Mandel. Interpretation of instantaneous frequencies. *American Journal Physics*, 42:840–846, 1974.

[84] J. Marques and L. Almeida. A background for sinusoid based representation of voiced speech. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, 11:1233–1236, Apr 1986.

[85] R. McAulay and M. Malpass. Speech enhancement using a soft-decision noise suppression filter. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(2):137–145, Apr 1980.

[86] R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 34(4):744–754, Aug 1986.

[87] P. Moreno, B. Raj, and R. Stern. A vector Taylor series approach for environment-independent speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 733–736, 1996.

[88] P.J. Moreno. *Speech recognition in noisy environments*. PhD thesis, Carnegie Mellon University, 1996.

[89] N. Morgan and H. Hermansky. RASTA extensions: robustness to additive and convolutional noise. In *Proceedings ESCA workshop of speech processing in adverse conditions*, pages 115–118, 1992.

[90] H.A. Murthy and V. Gadde. The modified group delay function and its application to phoneme recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 1, pages I–68–71, Apr 2003.

[91] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo. A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments. *Speech Communication*, 50(3):203–214, 2008.

[92] T. Nakatani and T. Irino. Robust and accurate fundamental frequency estimation based on dominant harmonic components. *Journal of the Acoustical Society of America*, 116(6):3690–3700, 2004.

[93] D.J. Nelson. Invertible time-frequency surfaces. In *Time-Frequency and Time-Scale Analysis, IEEE-SP International Symposium*, pages 13–16, Oct 1998.

[94] K. Nie, G. Stickney, and F-G. Zeng. Encoding frequency modulation to improve cochlear implant performance in noise. *Biomedical Engineering, IEEE Transactions on*, 52(1):64–73, Jan 2005.

[95] A.V Oppenheim. Speech spectrograms using the fast Fourier transform. *IEEE Spectrum*, pages 57–62, Aug 1970.

[96] A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, May 1981.

[97] A.V. Oppenheim and R.W Schafer. *Digital Signal Processing*. Prentice–Hall, 1975.

[98] D.H. Oppenheim, A.V. Johnson. Discrete representation of signals. *Proceedings of the IEEE*, 60(6):681–691, June 1972.

[99] K. Paliwal. Spectral subband centroid features for speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '88*, volume 2, pages 617–620, May 1998.

[100] K. Paliwal. On the use of filter-bank energies as features for robust speech recognition. *Signal Processing and Its Applications, 1999. ISSPA '99. Proceedings of the Fifth International Symposium on*, 2:641–644 vol.2, 1999.

[101] K. Paliwal and B. Atal. Frequency-related representation of speech. In *Proceedings Eurospeech*, 2003.

[102] K. Paliwal and A. Basu. A speech enhancement method based on Kalman filtering. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 12, pages 297–300, 1987.

[103] K. Paliwal and D. Zhu. Product of power spectrum and group delay function for speech recognition. *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pages 125–128, 2004.

[104] D. Pearce, H-G. Hirsch, and Ericsson Eurolab Deutschland Gmbh. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA ITRW ASR2000*, pages 29–32, 2000.

[105] S.R. Pillai and A. Antoniou. A robust representation of linear prediction coefficients. *Circuits and Systems, IEEE International Symposium on*, 2:53–56, 1996.

[106] A. Potamianos and P. Maragos. A comparison of the energy operator and the Hilbert transform approach to signal and speech demodulation. *Signal Processing*, 37(1):95–120, 1994.

[107] A. Potamianos and P. Maragos. Speech formant frequency and bandwidth tracking using multiband energy demodulation. *Journal of Acoustical Society of America*, 99:3795–3806, 1996.

[108] A. Potamianos and P. Maragos. Speech analysis and synthesis using an AM-FM modulation model. *Speech Communication*, 28(3):195–209, 1999.

[109] A. Potamianos and P. Maragos. Time-frequency distributions for automatic speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 9(3):196–200, Mar 2001.

[110] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett. The darpa 1000-word resource management database for continuous speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pages 651–654 vol.1, Apr 1988.

[111] T. Quatieri and A. Oppenheim. Iterative techniques for minimum phase signal reconstruction from phase or magnitude. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(6):1187–1193, Dec 1981.

[112] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. pages 267–296, 1990.

[113] L. Rabiner. On the use of autocorrelation analysis for pitch detection. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 25(1):24–33, Feb 1977.

[114] L. Rabiner and B-H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[115] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[116] B. Raj and R.M. Stern. Missing-feature approaches in speech recognition. *Signal Processing Magazine, IEEE*, 22(5):101 – 116, Sep 2005.

[117] A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual Evaluation of Speech Quality (PESQ) – a new method for speech quality assessment of telephone networks and codecs. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, volume 2, pages 749–752, 2001.

[118] M.R. Schroeder. Models of hearing. *IEEE*, 63:1332–1350, 1975.

[119] S.C. Sekhar and T.V. Sreenivas. Novel approach to AM-FM decomposition with applications to speech and music analysis. In *Acoustics, Speech, and Signal*

Processing, IEEE International Conference on ICASSP, volume 2, pages ii–753–6, May 2004.

[120] S.C. Sekhar and T.V. Sreenivas. Auditory motivated level-crossing approach to instantaneous frequency estimation. *Signal Processing, IEEE Transactions on*, 53(4):1450–1462, Apr 2005.

[121] B.J. Shannon. *Speech recognition and enhancement using autocorrelation domain processing*. PhD in Electrical engineering, Griffith University, 2006.

[122] R. Smits and B. Yegnanarayana. Determination of instants of significant excitation in speech using group delay function. *Speech and Audio Processing, IEEE Transactions on*, 3(5):325–333, Sep 1995.

[123] S. So and K. Paliwal. A long state vector Kalman filter for speech enhancement. In *Proceedings Interspeech*, pages 391–394, 2008.

[124] I.Y. Soon, S.N. Koh, and C.K. Yeo. Improved noise suppression filter using self-adaptive estimator of probability of speech absence. *Signal Processing*, 75(2):151 – 159, 1999.

[125] J.L. Spouge. Computation of the gamma, digamma, and trigamma functions. *SIAM Journal on Numerical Analysis*, 31(3):931–944, 1994.

[126] A. Stark and K. Paliwal. Speech analysis using instantaneous frequency. In *Proceedings Interspeech*, pages 2602–2605, 2008.

[127] A. Stark and K. Paliwal. Group-delay-deviation based spectral analysis of speech. In *International Conf. on Spoken Language Processing*, pages 1083–1086, 2009.

[128] A. Stark and K. Paliwal. Speech analysis using instantaneous frequency deviation. In *International Conf. on Spoken Language Processing*, pages 2602–2605, 2009.

[129] A. Stark, K. Wójcicki, J. Lyons, and K. Paliwal. Noise driven short-time phase spectrum compensation procedure for speech enhancement. In *International Conf. on Spoken Language Processing*, pages 2602–2605, 2009.

[130] V. Stouten. *Robust speech recognition in time-varying environments*. PhD thesis, Katholieke Universiteit Leuven, 2006.

[131] M. Sun and R.J. Sclabassi. Discrete-time instantaneous frequency and its computation. *Signal Processing, IEEE Transactions on*, 41(5):1867–1880, May 1993.

[132] T. Thiruvaran, E. Ambikairajah, and J. Epps. Analysis of band structures for speaker-specific information in FM feature extraction. In *Interspeech international conference*, 2009.

[133] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai. Mel-generalised cepstral analysis – a unified approach to speech spectral estimation. In *Proceedings ICSLP*, pages 1043–1046, 1994.

[134] A. Varga and H.J.M. Steeneken. Assessment for automatic speech recognition II: NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12:247–251, 1993.

[135] Y. Wang, J. Hansen, G.K. Allu, and R. Kumaresan. Average instantaneous frequency (AIF) and average log-envelopes (ALE) for ASR with the aurora 2 database. In *Eurospeech international conference*, pages 25–28, Sep 2003.

[136] T. Wesker, B. Meyer, K. Wagener, J. Anemuller, A. Mertins, and B. Kollmeier. Oldenburg logatome speech corpus (ollo) for speech recognition. In *In Proceedings of Interspeech*, pages 1273–1276, 2005.

[137] N. Wiener. *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*. Wiley, New York, 1949.

[138] K. Wójcicki, M. Milacic, A. Stark, J. Lyons, and K. Paliwal. Exploiting conjugate symmetry of the short-time Fourier spectrum for speech enhancement. *IEEE Signal Processing Letters*, 15:461–464, 2008.

[139] E. Wong and S. Sridharan. Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification. *Intelligent*

Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on, pages 95–98, 2001.

[140] B. Yegnanarayana and H.A. Murthy. Significance of group delay functions in spectrum estimation. *Signal Processing, IEEE Transactions on*, 40(9):2281–2289, Sep 1992.

[141] B. Yegnanarayana, D. Saikia, and T. Krishnan. Significance of group delay functions in signal reconstruction from spectral magnitude or phase. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 32(3):610–623, Jun 1984.

[142] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.0*. Cambridge University Press, 2000.

[143] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero. Robust speech recognition using a cepstral minimum-mean-square-error-motivated noise suppressor. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):1061 –1070, Jul 2008.