Short-time Phase Spectrum in Human and Automatic Speech Recognition

---

A Dissertation

Presented to

The School of Microelectronic Engineering,

Faculty of Engineering and Information Technology,

Griffith University.

Submitted in Fulfillment

of the Requirements of the Degree of

Doctor of Philosophy.

---

by

Leigh Alsteris, BEng (Hons)

August 2005

Abstract

Incorporating information from the short-time phase spectrum into a feature set for automatic speech recognition (ASR) may possibly serve to improve recognition accuracy. Currently, however, it is common practice to discard this information in favour of features that are derived purely from the short-time magnitude spectrum. There are two reasons for this: 1) the results of some well-known human listening experiments have indicated that the short-time phase spectrum conveys a negligible amount of intelligibility at the small window durations of 20–40 ms used for ASR spectral analysis, and 2) using the short-time phase spectrum directly for ASR has proven difficult from a signal processing viewpoint, due to phase-wrapping and other problems.

In this thesis, we explore the possibility of using short-time phase spectrum information for ASR by considering the two points mentioned above. To address the first point, we conduct our own set of human listening experiments. Contrary to previous studies, our results indicate that the short-time phase spectrum can indeed contribute significantly to speech intelligibility over small window durations of 20–40 ms. Also, the results of these listening experiments, in addition to some ASR experiments, indicate that at least part of this intelligibility may be supplementary to that provided by the short-time magnitude spectrum. To address the second point (i.e., the signal processing difficulties), it may be necessary to transform the short-time phase spectrum into a more physically meaningful representation from which useful features could possibly be extracted. Specifically, we investigate the frequency-derivative (or group delay function, GDF) and the time-derivative (or instantaneous frequency distribution, IFD) as potential candidates for this intermediate representation. We have performed various

experiments which show that the GDF and IFD may be useful for ASR. We conduct several ASR experiments to test a feature set derived from the GDF. We find that, in most cases, these features perform worse than the standard MFCC features. Therefore, we suggest that a short-time phase spectrum feature set may ultimately be derived from a concatenation of information from both the GDF and IFD representations. For best performance, the feature set may also need to be concatenated with short-time magnitude spectrum information.

Further to addressing the two aforementioned points, we also discuss a number of other speech applications in which the short-time phase spectrum has proven to be very useful. We believe that an appreciation for how the short-time phase spectrum has been used for other tasks, in addition to the results of our research, will provoke fellow researchers to also investigate its potential for use in ASR.

Keywords: Short-time Fourier transform, Phase spectrum, Magnitude spectrum, Speech perception, Automatic speech recognition, Overlap-add procedure.

Dedication

For Sharron and my parents, Leon and Lorraine.

Acknowledgments

First and foremost, I wish to thank Professor Kuldip Paliwal for your guidance, perseverance, motivating discussions, and for helping me stay the course.

Thankyou to Sharron, the love of my life, for your unwaivering patience and support. Without you, I would have lived in a pig sty for the last few years and I would have almost certainly suffered from malnourishment!

I also want to thank my parents, Lorraine and Leon, for your confidence in my abilities and the regular check-ups on my progress.

My gratitude also goes to my fellow students in the Signal Processing Laboratory. Thankyou for the technical discussions and for allowing me to seek your input on various ideas. Thanks also for the computer technical support!

Many thanks also go to the volunteers who took part in the human listening tests. These tests demanded a lot of time from the participants and were very repetitive. Some of you even had nightmares and flashbacks about your experiences! Without the kind donation of your time and concerted efforts, I would not have been able to compile the results that form the foundation of this thesis.

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 another person except where due reference is made in the thesis itself.

Leigh Alsteris

Table of Contents

List of Figures

List of Tables

List of Acronyms

ANN	Artificial neural network
ASR	Automatic speech recognition
CMS	Cepstral mean subtraction
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
FBE	Filter-bank energy
FBS	Filter-bank summation
FFT	Fast Fourier transform
GDF	Group delay function
HMM	Hidden Markov model
IF	Instantaneous frequency
IFD	Instantaneous frequency distribution
LP	Linear prediction
LPC	Linear prediction coefficient
LPCC	Linear prediction cepstral coefficient
MAP	Maximum a posteriori
MFCC	Mel frequency cepstral coefficient
MGDF	Modified group delay function
MLE	Maximum likelihood estimation
MODGDF	MGDF-based feature set
MSE	Mean-square error
OLA	Overlap-add

SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform

Chapter 1

Introduction

In the standard Automatic speech recognition (ASR) framework, speech is processed frame-wise using a temporal window duration of 20–40 ms. The short-time Fourier transform (STFT) is used for the signal analysis of each frame. The resulting signal spectrum can be decomposed into the short-time magnitude spectrum and the short-time phase spectrum[1]. Although information about the speech is provided by both components, state-of-the-art ASR systems generally discard the phase spectrum in favour of features that are derived only from the magnitude spectrum[2] [109]. A perusal of the literature reveals that there are two main reasons for this: 1) At such small temporal window durations, it is generally believed that the phase spectrum does not contribute much to speech intelligibility, and 2) the phase spectrum is an abstract representation from which it is difficult to extract information; in comparison, the magnitude spectrum is easy to analyse and parameterise into features.

The first reason for neglecting phase spectrum information in ASR can be attributed to the results of human speech recognition (HSR) studies conducted by Schroeder [127], Oppenheim and Lim [100], and Liu et al. [80]. Schroeder, Oppenheim and Lim infor-

[1]From here inward, the modifier 'short-time' is implied when mentioning the phase spectrum and magnitude spectrum. For clarity, we still explicitly state the modifier in various places throughout the thesis. Note that 'short-time' implies a finite-time window over which the properties of speech may be assumed stationary; it does not refer to the actual duration of the window. We use the qualitative terms 'small' and 'large' to make reference to the duration.

[2]There are other speech processing applications where spectral phase information is overlooked. For example, in speech enhancement it is common practice to modify the magnitude spectrum and keep the corrupt phase spectrum [76, 143].

mally observed that the phase spectrum is important for human intelligibility of speech when the window used for the STFT is large (greater than 1 sec). When the window duration is small (about 20–40 ms), they noted that the phase spectrum conveys little information about the intelligibility of speech. Liu et al. have recently conducted a more formal human speech perception study, the results of which agree with those of previous studies. Note that while the phase spectrum is thought to provide little toward speech intelligibility, it does contribute to the speech quality and naturalness; accordingly, it's information is captured by speech coding algorithms and used for high-quality speech synthesis [15, 34, 69, 110, 111].

The other reason for discarding the phase spectrum in ASR is due to signal processing difficulties such as phase-wrapping and other problems [85,145]. The magnitude spectrum is relatively easy to analyse and parameterise into features. The same analysis and parameterisation techniques, however, can not be used to process the information provided by the phase spectrum. Unlike the magnitude spectrum, the phase spectrum does not explicitly exhibit the system resonances. A physical connection between the phase spectrum and the structure of the vocal apparatus is not immediately apparent.

To avoid confusion, we provide a clarification: From experience, we have learned that when one hears the word 'phase', it is often associated with the quantity explored by Ohm [97] and Helmholtz [55]. This type of 'phase' refers to the changes in arrival time of a signal's frequency components due to variations in the path length between the signal source and the ear [18]. This, however, is a different concept to that of the short-time phase spectrum. The two should not be confused[3].

Although the phase spectrum has yet to be proven useful for ASR, it has successfully been used for many other tasks; some of which we touch upon in the literature review. The literature review leads in to our own work in which we have been exploring the potential for using the phase spectrum to improve ASR performance. Our motivation for research in this direction stems from the results obtained from our own listening tests [107]. Contrary to previous studies by other researchers [80,100,127], our results indicate that the phase spectrum can indeed contribute significantly to speech intelligibility over

[3]To eliminate any ambiguity, throughout this thesis we use the term 'phase spectrum', not 'phase'.

small window durations of 20–40 ms. These results dispel the first aforementioned reason for not using the phase spectrum in ASR. The second reason (i.e., the signal processing difficulties) requires some further investigation. If the phase spectrum is to be useful in ASR, it is necessary that it be transformed into a more tangible representation. If such a representation can be found, we need to determine if it can be used (either in isolation or in combination with magnitude spectrum) to improve ASR performance.

The remainder of this chapter is structured as follows: In Section 1.1, we describe the organisation of this thesis and list the sections that comprise the literature review. In Section 1.2, we list the contributions made in this thesis. The journal and conference papers resulting from this work are listed in Section 1.3.

1.1 Thesis Organisation

1.1.1 Chapter Summary

The outline of this thesis is as follows:

- **Chapter 2** serves to provide a general review of the ASR literature. We discuss the state-of-the-art in ASR and explain the statistical framework on which ASR is based. Each component of a typical ASR system is then described, from the front-end (the part of the system which captures, conditions and parameterises the speech signal) through to the back-end (where the pattern recognition occurs).

- **Chapter 3** introduces the short-time Fourier transform and discusses how it is used to analyse, synthesise and modify a speech signal. We mention two common representations derived from the short-time phase spectrum, briefly describing a number of speech applications in which these representations have successfully been used. By demonstrating the usefulness of the short-time phase spectrum for several applications, we intend to provoke other researchers to ponder its possible use in ASR.

- **Chapter 4** describes several human perception experiments that we have conducted in order to quantify the intelligibility provided by the short-time phase

spectrum and the short-time magnitude spectrum in clean and noisy conditions. This is done with the aid of speech stimuli reconstructed either from the original short-time phase spectra or the original short-time magnitude spectra. These stimuli are respectively referred to as *phase-only* stimuli and *magnitude-only* stimuli. We also attempt to quantify the intelligibility provided by the frequency-derivative of the phase spectrum (group delay function, GDF) and the time-derivative of the phase spectrum (instantaneous frequency distribution, IFD).

- **Chapter 5** follows on from Chapter 4 in that we perform ASR on phase-only stimuli and magnitude-only stimuli. We employ an MFCC-based front-end to see if the ASR recognition scores are consistent with the human intelligibility scores.

- **Chapter 6** provides a tutorial-like review of iterative, one dimensional, signal reconstruction (specifically speech signals). Any researcher with an interest in the phase spectrum should be aware of the flurry of activity that occurred in the 1980's in the area of iterative signal reconstruction. In addition to the review, we provide the results of some further experimentation which may be interesting from an ASR viewpoint.

- **Chapter 7** first reviews the GDF in detail and highlights the problems when using it directly for ASR. We summarise the work by Yegnanarayana and Murthy [145] on the modified GDF (MGDF), which serves to remedy the problems of the GDF. We provide the implementation details of Murthy and Gadde's MGDF-based (MODGDF) features [88]; these features are perhaps the most concerted effort into spectral phase features thus far. We test the MODGDF features on several ASR tasks in additive white and coloured noises.

- **Chapter 8** summarises the work presented in this thesis and presents the conclusions that have been drawn from the work. We also suggest some further research.

1.1.2 Composite Literature Review

The literature review is spread over the following sections:

- The whole of Chapter 2, which provides a general review of the ASR literature.
- The whole of Chapter 3, which covers the theory of the short-time Fourier transform and the uses of the short-time phase spectrum.
- Section 6.1, which is a tutorial-like review of iterative signal reconstruction.
- Sections 7.1 and 7.2, which discuss the GDF and the MGDF respectively.

1.2 Contributions

The work presented in this thesis makes a number of original contributions:

1. Chapter 4: Several human listening experiments have been conducted in order to quantify the intelligibility provided by the short-time phase spectrum and short-time magnitude spectrum.
2. Chapter 5: ASR experiments have been performed on speech which has been reconstructed from only its short-time phase spectra or its short-time magnitude spectra.
3. Section 6.2: Iterative reconstruction of speech from partial short-time phase spectrum information.
4. Section 7.4: We conduct an independent test on the MODGDF features, which were devised by Murthy and Gadde [88].

1.3 Publications Resulting from Research for this Thesis

1.3.1 Journal Articles

1. K.K. Paliwal and L.D. Alsteris, "On the usefulness of STFT phase spectrum in human listening tests", *Speech Communication*, Vol. 45, No. 2, pp. 153–170, Feb. 2005.
2. L.D. Alsteris and K.K. Paliwal, "Further intelligibility results from human listening tests using the short-time phase spectrum", *Speech Communication*, in press.

3. L.D. Alsteris and K.K. Paliwal, "Iterative reconstruction of speech from short-time Fourier transform phase and magnitude spectra", submitted to *Computer Speech and Language*, under review.

4. L.D. Alsteris and K.K. Paliwal, "The short-time phase spectrum in speech processing: a review", submitted to *Digital Signal Processing*, under review.

1.3.2 Conference Papers

1. K.K. Paliwal and L. Alsteris, "On the importance of phase spectrum in speech perception", *Proc. International Conf. Perception and Action*, July 2003.

2. K.K. Paliwal and L. Alsteris, "Usefulness of phase spectrum in human speech perception", *Proc. European Conf. Speech Communication and Technology*, pp. 2117–2120, Sept. 2003.

3. L.D. Alsteris and K.K. Paliwal, "Intelligibility of speech from phase spectrum", *Proc. Microelectronic Engineering Research Conf.*, Nov. 2003.

4. L.D. Alsteris and K.K. Paliwal, "Importance of window shape for phase-only reconstruction of speech", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 573–576, May 2004.

5. L.D. Alsteris and K.K. Paliwal, "ASR on speech reconstructed from short-time Fourier phase spectra", *Proc. International Conf. Spoken Language Processing*, Oct. 2004.

6. L.D. Alsteris and K.K. Paliwal, "Evaluation of the modified group delay feature for isolated word recognition", *Proc. International Symposium on Signal Processing and its Applications*, Sydney, Australia, pp. 715–718, Aug. 2005.

7. L.D. Alsteris and K.K. Paliwal, "Some experiments on iterative reconstruction of speech from STFT phase and magnitude spectra", *Proc. European Conf. Speech Communication and Technology*, pp. 337–340, Sept. 2005.

Chapter 2

Automatic Speech Recognition

ASR is the process by which a computer identifies a sequence of words from an acoustic signal. This sequence of words can then be further processed, depending on the application. For example, the words may be interpreted as commands for a computer program, they may be directly transcribed into a document (i.e., dictation software), or converted to another language and then played through a text-to-speech (TTS) synthesiser.

Fig. 2.1 illustrates the major components of an ASR system, all of which will be further discussed throughout this chapter. Signal acquisition and conditioning involves analog-to-digital conversion of the signal as well as digital filtering to emphasise the important components of the speech. Our primary component of interest is that of feature extraction. This is the process by which the digital speech signal is transformed in to a stream of vectors that represent events in a probability space. Given these vectors, the acoustic models, the pronunciation lexicon, and the language model, we compute the most likely sequence of models that generated the vectors. The sequence of models are then mapped (by using the lexicon) to a sequence of words. The performance of an ASR system can then be evaluated by comparing the decoded utterance to the transcriptions of the actual words spoken.

ASR involves input from many disciplines, including signal processing, pattern recognition, artificial intelligence, information theory, psychoacoustics and psychology. Consequently, the literature on ASR is vast. A full and complete review is beyond the scope of this dissertation. The purpose of this chapter is to review only the most pertinent

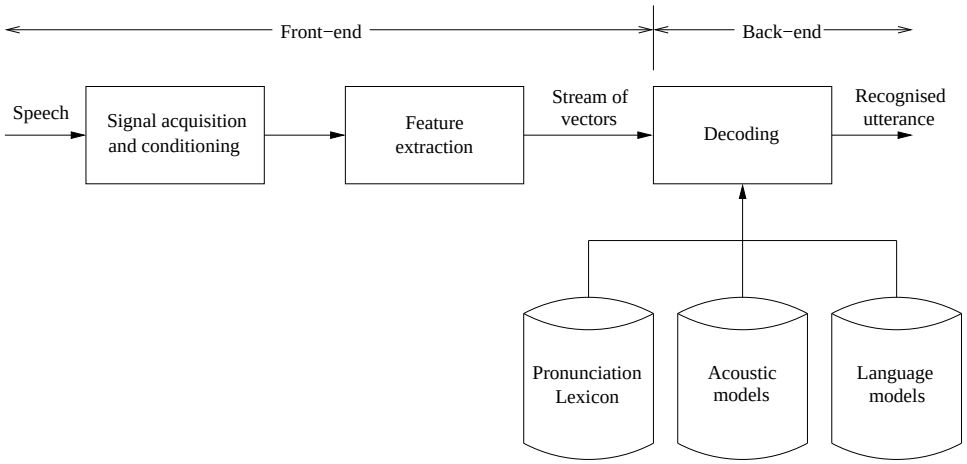


Fig. 2.1: Components of of typical ASR system.

aspects on the subject. In the first section, we discuss the state-of-the-art, giving examples of the systems that are currently in use. In the second section, we set the scene for the remainder of the chapter by introducing the statistical framework on which a typical ASR system is based. Subsequent sections discuss each component of an ASR system in order (as shown in Fig. 2.1), from the front-end (the part of the system which captures, conditions and parameterises the speech signal) through to the back-end (where the pattern recognition/decoding occurs).

2.1 The State of the Art

The ultimate goal in the field of ASR is to create a system that will recognise unrestricted, continuous speech utterances from any speaker, of any language. Since speech recognition comes so naturally to humans (at least for one language), it is easy to see how the difficulty of this task is often underestimated. In fact, over fifty years of research has proven that human speech recognition is an extremely difficult task to emulate on a computer.

Although human-like performance remains in the realm of science fiction, a significant amount of progress has been made toward the realisation of such systems. The first ASR systems in the 1950's recognised vowels, consonants and syllables. The general consensus of researchers was that these basic systems would form the foundation for a

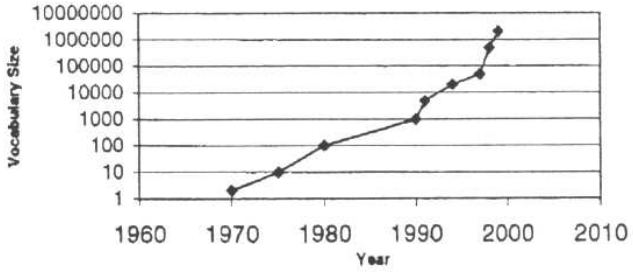


Fig. 2.2: Evolution in vocabulary size for ASR (after [23]).

continuous ASR system, and that building such a system from this foundation would not be an extremely difficult problem. However, time has shown otherwise. The 1960's and 1970's saw mainly digit and word recognition systems being implemented thanks to the use of dynamic programming for time-aligning utterances. The 1980's saw the insurgence of statistical-based methods for recognition, namely Hidden Markov Modeling [121] in addition to the emergence of large speech corpora [79], such as TIMIT [33], Resource Management [118], and Switchboard [46]. Further improvements in statistical modeling have taken place in the 1990's, in addition to improvements in the accompanying search algorithms (which scan the space of possible recognition outcomes). These search algorithms have allowed a remarkable increase in vocabulary size, such that today any vocabulary size can be recognised in real time [23] (Fig. 2.2).

Undeniably, a key factor in the explosion of progress in most recent times has been the advances in computer technology. The phenomenal increase in speed and memory of personal computers has enabled researchers to investigate and implement very complex algorithms. Furthermore, with the promise of continued improvements in computing technology, algorithms that work slower than real-time today may work in real-time soon into the future. In addition, the proliferation of the Internet and e-mail has made the process of information exchange much more efficient. Researchers from different groups around the globe can easily share computer programs and work jointly on projects while being in different geographical locations. Obviously, this has happened in the past, but the Internet promotes much more of this collaboration. Also, there are international competitions in which systems are evaluated, further promoting progress in the field as well as interaction between research labs.

Despite recent advances in ASR and the impressive performance of various systems in laboratory conditions, the fact is that ASR systems are not robust. That is, they are unable to maintain a consistent level of performance when there is a mismatch between training and operating (or testing) conditions. This mismatch is realised through variabilities introduced by speakers, transducer equipment and the acoustic environment (discussed further in Section 2.8). Consequently, all systems that use ASR technology must be designed to compensate for its shortcomings. Programs need to be written to effectively manage when something goes wrong.

Specifically, ASR is currently finding applications in devices such as personal digital assistants, mobile phones [94], kitchen appliances, automobiles [59], toys, etc. It is also used to automate services such telephone banking and call routing. Many software developers are now able to develop speech-enabled software simply by using core speech engines made available through common application programming interface libraries, such as Microsoft's SAPI (speech application programming interface) and Java's J-SAPI. It is possible to navigate the Internet by voice and so-called voice portals are allowing access to Internet-based information via the telephone.

Most ASR systems are of the command-and-control type. These systems use a grammar where the commands are from a list of commands that the application is expecting to hear. Such systems do not provide a natural interaction, which may defeat the purpose of having the speech mode in the first place. For example, telephone ASR systems are constrained by strict dialog systems, where any deviation from what is expected will cause the system to fail. The annoyance of such a failure should be familiar to almost anybody who has attempted telephone banking. Dictation systems, such as Dragon Naturally Speaking and IBM Viavoice, must be trained for each speaker and each microphone. They require specialised language models and the speaking rate must also be limited to a reading pace (or an unnaturally slow speaking rate).

The requirement of the future is mobility – devices are getting smaller and portability is a need. Keyboards and mice are becoming impractical, opening the way for ASR. This requirement is the key driving force behind the recent growth in popularity of the speech modality and is not necessarily indicative of the usability of current ASR

systems. Although all of this progress may seem to indicate that ASR is a mastered art, the truth is that there is still much to be achieved.

2.2 Statistical Framework for ASR

ASR is fundamentally a pattern recognition problem. The majority of ASR systems use a statistical pattern matching technique (i.e., the distance or similarity between a test pattern and a model is expressed in terms of a probability). It makes sense to use such a technique because speech is a statistically fluctuating signal. More precisely, instances of speech that convey the same phonetic content differ slightly due to random variations.

Essentially, we want to take the speech (which is a pattern) and classify it as sequence of previously learned patterns. Given a set of observed acoustic vectors, $O = \{\vec{o}_i\}_{i=1}^{n}$, we wish to determine the most likely sequence of models[1], $\widetilde{\Lambda} = \{\lambda_i\}_{i=1}^{m}$. This set of models has a corresponding set of parameters[2], $\Theta_{\widetilde{\Lambda}} = \{\theta_i\}_{i=1}^{m}$. In principle, the most likely sequence of models is computed as follows:

$$\widetilde{\Lambda} = \arg\max_{\Lambda_i} P(\Lambda_i|O, \Theta_{\Lambda_i}), \qquad \text{for } i = 1, 2, \ldots, M, \tag{2.1}$$

where M represents the number of all possible model sequences[3]. These models correspond to some kind of linguist unit, whether it be a phoneme, syllable, word, a sequence of words or possibly some higher or lower level of abstraction (see Section 2.4.2). In any case, the sequence of models, $\widetilde{\Lambda}$, reveals the most probable utterance that was spoken.

The parameters for each model are determined through a training procedure before recognition takes place. This training is done in a supervised manner; that is, we have predetermined the number and the structure of the models we want to use and we also know which acoustic vector sequences should be used to train which model sequences (e.g., the acoustic vector sequence O_i is used to train the model sequence Λ_i). In other

[1] λ_i is simply a label for model number i. In turn, Λ denotes a sequence of these models, and $\widetilde{\Lambda}$ is the most likely sequence of models.

[2] θ_i denotes the parameters for model λ_i. Θ_Λ denotes a sequence of model parameters (as opposed to model labels), corresponding to model sequence Λ.

[3] In practice, only a subspace of all possible model sequences is searched. The possible model sequences must agree with phonetic, syntactic and pragmatic constraints.

words, the data is labeled. The parameters are selected such that they maximise the probability of the models, given the training data. If we let $L = \{\theta_i\}_{i=1}^{K}$ represent a set of all model parameters (where K is the total number of models), then the best set of model parameters, $\widetilde{L}$, is calculated as:

$$\widetilde{L} = \arg\max_{L} \prod_{i=1}^{I} P(\Lambda_i | O_i, \Theta_{\Lambda_i}), \tag{2.2}$$

where I is the number of training sequences. This is referred to as maximum *a posteriori* (MAP) training. However, in practice, a maximum likelihood estimation (MLE) training method is often used. $P(\Lambda|O, \Theta_\Lambda)$ can be expressed as:

$$P(\Lambda|O, \Theta_\Lambda) = \frac{P(O|\Lambda, \Theta_\Lambda)P(\Lambda)}{P(O)}. \tag{2.3}$$

This is referred to as Bayes formula. While MAP training maximises $P(\Lambda|O, \Theta_\Lambda)$, MLE training assumes that $P(\Lambda)$ and $P(O)$ are constant and $P(O|\Lambda, \Theta_\Lambda)$ is maximised instead. During testing, however, $P(\Lambda)$ is assumed to vary and Eq. 2.1 can be rewritten as:

$$\widetilde{\Lambda} = \arg\max_{\Lambda_i} \Big(P(O|\Lambda_i, \Theta_{\Lambda_i}) P(\Lambda_i) \Big), \qquad \text{for } i = 1, 2, \ldots, M. \tag{2.4}$$

For the purposes of efficient calculation, we can maximise the logarithm of the probability instead:

$$\widetilde{\Lambda} = \arg\max_{\Lambda_i} \Big(\log P(O|\Lambda_i, \Theta_{\Lambda_i}) + \log P(\Lambda_i) \Big), \qquad \text{for } i = 1, 2, \ldots, M. \tag{2.5}$$

Note that the logarithm is a monotonically increasing function, thus both Eq. 2.4 and Eq. 2.5 provide the same answer. This is the discriminant function used in ASR. The first term, $\log P(O|\Lambda_i, \Theta_{\Lambda_i})$, is determined by comparing the observed acoustic vectors to the statistical models; in most systems (and in this dissertation), Hidden Markov models are used (see Section 2.4.1). The second term, $\log P(\Lambda)$, contributes prior knowledge of the speech structure via the language model (see Section 2.5).

2.3 Feature Extraction

The process of obtaining the acoustic observation vectors (from herein referred to as feature vectors or feature representation), $\vec{o}_i$, from the speech signal is commonly referred to as feature extraction. The aim of feature extraction is to encapsulate the linguistic information[4] from the speech signal into an efficient and compact representation which can be further processed and statistically analysed. The feature vectors should be good at separating different classes of speech sounds while also suppressing irrelevant sources of variation. The feature vectors can be derived directly from the time domain (i.e., temporal analysis) or from the frequency domain (i.e., spectral analysis).

In the following subsections, we describe a number of feature representations. One would expect that knowledge of how the human auditory and vocal mechanisms work would prove useful in deriving the feature vectors. Thus, to begin this section, we summarise the important findings from studies on human speech perception and production.

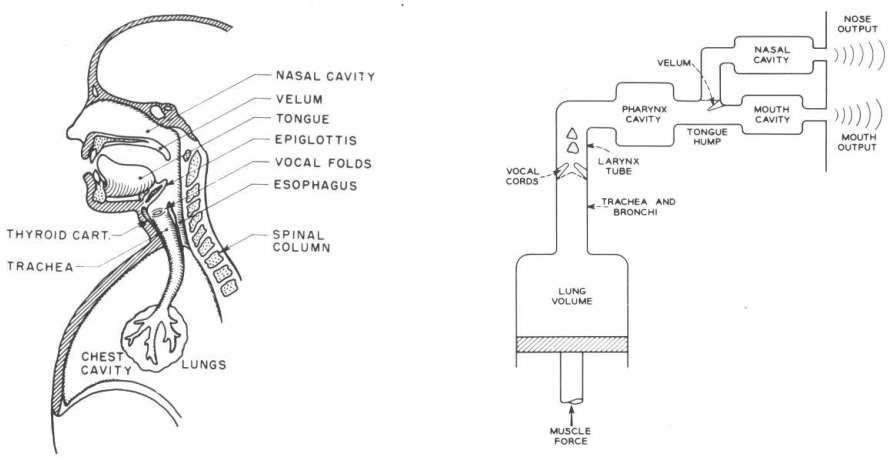
2.3.1 Lessons from Psychoacoustics

Human speech recognition persists under many types of adverse conditions. Replication of this performance is the ultimate goal in ASR research. A common school of thought is that in order to achieve such performance, the structure of the ASR system should be modeled after the human auditory system. Although the auditory system is still not completely understood, various concepts have proven useful in their application to ASR.

An understanding of the speech production mechanism is also essential for speech scientists and engineers who work in the field of ASR. Knowing how speech is produced allows us to make simplifying assumptions about the nature of the speech signal which, in turn, enables the application of some well established signal processing algorithms.

Most feature representations, in one way or another, are influenced by the knowledge that we possess about both the human auditory and speech production mechanisms.

[4]The speech signal consists of both a linguistic component and a non-linguistic component. The linguistic component comprises of any information that aids in identifying the words that have been spoken. The non-linguistic component conveys information that may be useful for other tasks; such as age, gender, and identity determination. Background noise and channel distortions also contribute to the non-linguistic component.



(a) (b)

Fig. 2.3: (a) The speech production mechanism and, (b) its schematic representation (after [35]).

Some of these feature representations employ only some simple concepts [27, 56], while other representations attempt to simulate the auditory system in great detail [45, 70, 130]. In this section, we briefly describe the human (peripheral) auditory and speech production mechanisms, introducing those properties commonly utilised in ASR.

2.3.1.1 Speech Production

Fig. 2.3 presents both a detailed and schematic representation of the vocal apparatus. The lungs provide the excitation (respiration), the vocal cords convert the energy into audible sound (phonation), and the articulators transform the sound into speech (articulation). The articulators consist of the velum, tongue, lips and jaw. The movement of the articulators determines the shape of the vocal tract. Opening and closing of the velum controls the involvement of the nasal tract.

The representation of speech production in Fig. 2.4 is known as the source-filter model. This is a simple but effective mathematical model, which finds use in ASR, speech synthesis and coding. The sound source is representative of the lungs and vocal cords. The signal from the sound source is treated as either a periodic pulse train or random (white) noise. The periodic pulse train is the excitation for voiced sounds and

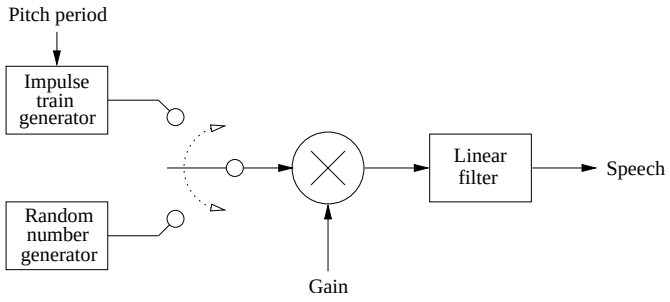


Fig. 2.4: The source-filter model — a simplified model of speech production. The properties of the linear filter (i.e., vocal tract) are assumed to be stationary over short periods of time (20–40 ms).

the random noise is the excitation for non-voiced sounds. The period of the pulse train is referred to as the fundamental frequency (or F0) and it determines the pitch for voiced sounds. The excitation signal passes through the vocal tract, which can be modeled by a time-varying linear filter. This filter also encompasses the effects of the glottal pulse shape and radiation from the mouth (see Section 2.3.2 for further discussion).

Consider a speech segment centered at time t_0. The segment is assumed to be statistically stationary over its duration. If the power spectrum[5] of the excitation over the segment duration is denoted as $P_e(\omega, t_0)$ and the spectral magnitude response of the vocal tract over the segment duration is denoted as $|H(\omega, t_0)|$, then according to the source-filter model, the power spectrum of the resulting speech segment, $P_x(\omega, t_0)$, can be expressed as:

$$P_x(\omega, t_0) = |H(\omega, t_0)|^2 P_e(\omega, t_0). \tag{2.6}$$

This assumes independence between the source and the vocal tract. In reality, however, there is some feedback between the two. Regardless, the independence assumption of the source-filter model works well in practice.

The fine structure in $P_x(\omega, t_0)$ is attributed to $P_e(\omega, t_0)$. Thus, smoothing $P_x(\omega, t_0)$ effectively divides out the effect of $P_e(\omega, t_0)$ (the spectral envelope of the excitation is assumed to be flat). Therefore, $|H(\omega, t_0)|^2$ is estimated by the spectral envelope of the signal[6].

[5] See Section 2.3.4 for a definition of the magnitude spectrum and the power spectrum.

[6] Note that only the power spectral response of the vocal tract is considered in this expression. The phase spectrum is ignored. See Section 2.3.4 for a definition of the phase spectrum.

2.3.1.2 Speech Perception

The hearing process involves both physiological and psychological processing, whose combined purpose is to convert sound pressure waves into a linguistic message. The ear converts the pressure waves into electrical impulses in the auditory nerve (physiological processing), while the brain performs further processing and pattern matching to decode the message (psychological processing).

Fig. 2.5 presents a diagram of the human ear (i.e., the peripheral component of the auditory system). The ear is compartmentalised into three distinct regions:

1. The outer ear: captures the sound waves, funneling them down the auditory canal, toward the eardrum (tympanic membrane).

2. The middle ear: consists of the ear drum and small bone structure (Malleus, Incus and Stapes). The eardrum transforms the sound waves into mechanical movements, which in turn vibrate the bone structure. The mechanism formed by the bones performs impedance matching between the air medium in the outer ear and the fluid medium in the inner ear.

3. The inner ear: contains the cochlea, which acts as a spectrum analyser, converting the mechanical vibrations (received via the oval window) into electrical impulses in the auditory nerve. The fluid inside the cochlea vibrates tiny hairs, called cilia, on the basilar membrane, which are attached to the auditory nerve. Very little, if anything, is understood about what transformations or pattern recognition approaches are applied to this auditory nerve signal.

Next, we summarise some of the most important psycho-acoustic results. There are many good references on the subject [77, 103].

Loudness Versus Intensity It is commonly accepted that there is a logarithmic relationship (to the base 10) between perceived loudness and the actual sound intensity. Consequently, feature representations are often derived from a logarithmic compressed power spectra rather than a linear power spectra [109]. Alternative relationships, such as cubic root compression, have been proposed [3, 75].

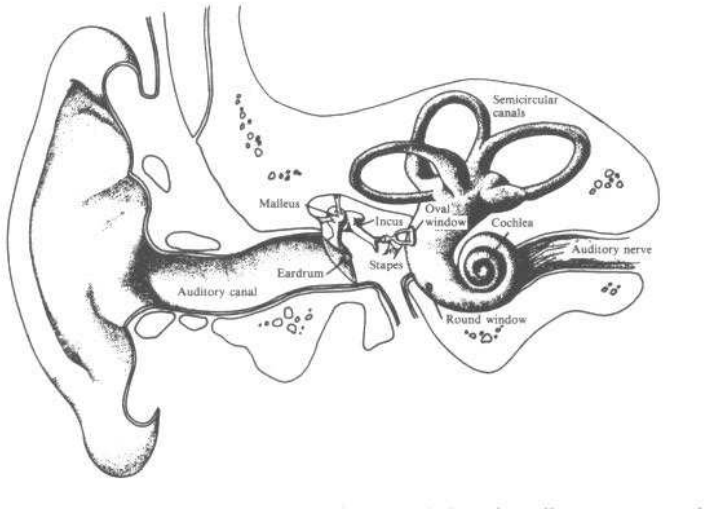


Fig. 2.5: The peripheral auditory system (after [77]).

Frequency Resolution and Critical Bands The basilar membrane is likened to a spectrum analyser. It's frequency response is non-linear; the frequency resolution decreases with increased frequency. Two different approaches used to quantify this non-linear frequency response of the ear are described below. The result is two similar perceptual scales — the Bark scale and the Mel (Melody) scale.

To explain the derivation of the Bark scale, we must first explain the term critical bandwidth – if a band of noise is kept at a constant intensity while its bandwidth is increased, it will be perceived to have constant loudness until the critical bandwidth is reached[7]. The critical bandwidth of filters centered at several frequencies were observed then used to devise the Bark scale. The Bark scale is designed such that critical bandwidths have a constant value of one Bark (as opposed to the linear scale, where the critical bandwidth must be quoted in reference to the center frequency). The Bark scale ranges from one to 24 Barks, which corresponds to 24 contiguous critical bands of hearing. The length of the basilar membrane is typically 35mm, so one Bark also corresponds to a 1.5mm spacing. It should be noted, however, that the critical bands in the

[7]Critical bandwidth is often explained in the context of masking. The threshold of hearing for a single tone (reference tone) is raised in the presence of another tone (the masker). The amount by which the threshold of the reference tone is raised depends on how close the masker tone is in frequency. Frequencies outside the critical bandwidth of the reference tone have a negligible effect on its threshold of hearing [104].

ear are not butted end to end; in fact, every audible tone has a critical band centered on it. The mapping between linear frequency, f (in Hertz), and Bark frequency, B (units of Bark), is expressed as follows [154]:

$$B(f) = 13 \arctan\left(\frac{0.76f}{1000}\right) + 3.5 \arctan\left(\frac{f}{7500}\right)^2. \tag{2.7}$$

Note that this equation is only one approximation to the empirically measured curve. Several other equations that approximate the Bark/Hertz curve have been proposed; for example, [128].

The Mel scale evolved from a number of human perception experiments conducted by Stevens and Volkman [134]. Listeners were given a reference tone and asked to adjust the tone to half and twice of its frequency. The data from many such trials, with many different reference tone frequencies, were used to create the Mel scale. The Mel scale is devised such that a perceived frequency halving (or doubling) is exactly a halving (or doubling) in terms of Mel (i.e., units of the Mel scale). To determine the perceived halving (or doubling) of a reference tone in Hertz, convert the tone's frequency to Mel, halve (or double) the Mel value, then convert it back to Hertz. The Mel scale is linear below 1 kHz and logarithmic above. An analytic approximation of the scale is given by:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{2.8}$$

where f is the linear frequency (in Hertz), and $M(f)$ is the perceived frequency (in Mel).

Although the scales were derived by different methods, it can be observed in Fig. 2.6 that the Bark and Mel scales are very similar. The ASR community has chosen to adopt the Mel scale as the pseudo-standard perceptual warping function, however, the Bark scale is just as appropriate. For example, Mel-frequency cepstral coefficients (see Section 2.3.6.2) are the standard feature representation for ASR. However, Bark-frequency cepstral coefficients provide equally impressive recognition scores [131].

Mean-rate Versus Place Representation Along the length of the basilar membrane there exists nerve fibres. Each of these nerve fibres fire if a stimulus is composed

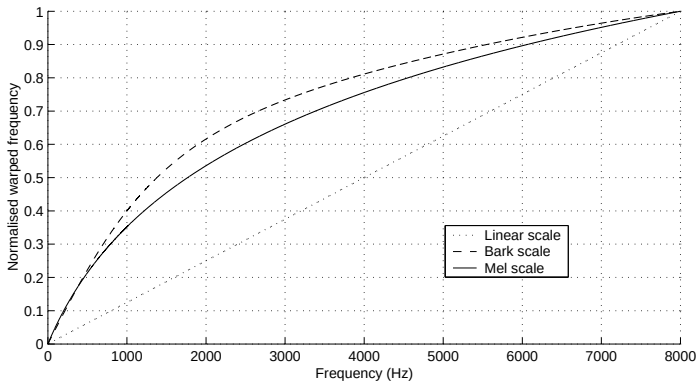


Fig. 2.6: Frequency warping of the Bark and Mel scales. The Bark curve is calculated with Eq. 2.7 and the Mel curve is calculated with Eq. 2.8. Both curves are normalised by their maximum values (at 8 kHz). The normalised linear scale is also shown.

of frequencies within its critical band. The centre frequency depends on the physical location (or place) of the nerve fibre along the basilar membrane. When the nerve fibre fires, it becomes charged. The higher the intensity of the stimulus, the more charge the nerve fibre receives, and in turn it takes longer to discharge. Thus, sound intensity is proportional to the mean discharge rate of the nerve fibres. This provides a measure of the power spectrum. It is referred to as the mean-rate versus place representation [130]. However, this representation is only adequate for low sound pressure levels. At moderate to high sound levels (which are typical of normal conversation) the response of the nerve fibers becomes saturated.

Temporal-synchrony Versus Place Representation The firing rate of a nerve fibre is temporally phase-locked to frequencies of the speech that are within its critical band. A measurement of this phase-locking provides detailed spectrum information. This is referred to as the temporal-synchrony versus place representation [130]. It is thought that this representation could explain the intelligibility of speech in high sound levels and in background noise. Phase-locking is observed for spectral components with frequencies of less than 4 kHz. Above this frequency, the mean-rate phenomenon still occurs. The general consensus is that the human auditory system uses both mean-rate and temporal-synchrony representations to perceive speech signals.

2.3.2 Signal Acquisition and Conditioning

Signal acquisition and conditioning involves sampling the analog signal and performing some preliminary processing. A microphone is required to transform the speech into an electrical signal, $x(t)$. Samples of the continuous electrical signal are taken at intervals of T_s (where the sampling frequency is $f_s = 1/T_s$) and passed through a quantiser (usually 16 bits) to produce a digital signal, $x(n)$.

Voiced segments of this digital signal exhibit a negative spectral slope[8]. Pre-emphasis is the process by which this natural slope is offset, equalising the dynamic range across the entire frequency band. This is done by implementing a simple difference filter (backward difference):

$$H_p(z) = 1 - az^{-1}, \tag{2.9}$$

where a is typically a value between 0.9 and 1.0. In the time domain, the relationship between the input signal and the pre-emphasised signal is given by:

$$x_p(n) = x(n) - ax(n-1). \tag{2.10}$$

Pre-emphasis is only required for voiced speech, however, the effect that it has on unvoiced speech is negligible. Thus, to minimise system complexity, pre-emphasis is applied to all speech.

The movement of the articulators is relatively slow, therefore the vocal tract characteristics can be assumed constant over short intervals of time. From an engineering point of view, the speech signal can be seen as the output of a quasi-stationary random process. Assuming that this is the case, the pre-emphasised stream of digital data is analysed in frames (or blocks) of typically 20–40 ms, at intervals of 10–20 ms. The discontinuities at the ends of each frame cause excessive spectral leakage[9]. The distortion

[8] The voiced source has a high-frequency roll-off of -12dB/octave and the acoustic radiation impedance (of the lips) has a low-frequency roll-off of 6dB/octave. Thus, this total impedance effect produces a -6dB/octave slope over the vocal tract frequency response.

[9] Consider a frame of N samples. The discrete Fourier transform of the frame results in a frequency resolution of f_s/N, where f_s is the sampling frequency. Those frequencies in the signal that are not an integer multiple of this frequency resolution will exhibit non-zeros values over the entire spectrum. This leakage effect is reduced by using windows that possess a reduced discontinuity at their boundaries, such as Hamming, Hanning, Blackman and Gaussian windows.

is reduced by weighting the samples with a tapered window. The most common tapered window is the Hamming window [49]:

$$w_h(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N}, \qquad \text{for } n = 0, 1, \ldots, N-1. \tag{2.11}$$

This window decreases the leakage effect at the expense of reducing frequency resolution. However, in ASR, high resolution is not of primary importance because the formants (whose positions are determined by the configuration of the vocal tract) are relatively far apart and fine spectral harmonics are generally discarded.

In the following sections, we present some popular methods by which the information in these sampled speech frames are represented for ASR.

2.3.3 Linear Prediction Analysis

Linear prediction was introduced by Atal in 1971 for speech coding [14]. This technique subsequently became popular for use in ASR because it provides a compact representation of the spectral envelope. It was the dominant method for feature extraction throughout the 1970s and early 1980s. It has played an important role in the evolution of ASR. However, nowadays there exists other methods that are more resilient to the effects of noise.

In linear prediction, the current speech sample, $x(n)$, is estimated by a linear combination of the past p samples[10]:

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + e(n), \tag{2.12}$$

where $e(n)$ is the error term (or residual signal) and the $\{a_k\}_{k=1}^{p}$ values are referred to as the linear prediction coefficients (LPCs). Further analysis of this equation reveals that it is equivalent to assuming that the vocal tract is modeled by an all-pole filter:

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.13}$$

[10]The value p is generally chosen to be the sampling frequency (in kHz) plus 2.

The LPCs are determined by minimising the error term, $e(n)$. The most popular method of doing this is referred to as the autocorrelation method. This method results in a set of equations called the Yule-Walker equations. The structure of the Yule-Walker equations is such that they can be solved efficiently using the Levinson-Durbin algorithm. This algorithm is recursive and requires $O(p^2)$ operations.

There are a number of other representations of these LPCs, such as reflection coefficients (or partial correlation coefficients), log area ratios, and line spectral-pair frequencies (LSFs) [65]. However, in recent times, LP-derived representations have been discarded in favour of representations derived from the Fourier transform magnitude spectrum which tend to perform better with noisy speech. The LP analysis assumes an all-pole system response. Thus, recognition accuracy becomes worse when a number of zeros exist, such as in nasal and fricative sounds. Drastic performance degradation is observed in the presence of additive noise, which also introduces zeros into the spectrum.

2.3.4 Fourier Transform Analysis

For spectral analysis, two main methods exists; those based on linear predictive coding (Section 2.3.3) and those based on Fourier analysis. For ASR, the latter method has proven to be more popular in recent times. Given a frame of speech that is N samples long, the discrete-time Fourier transform is expressed as[11]:

$$X(\omega) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\omega n}. \tag{2.14}$$

In practice, this spectral estimate is calculated at N equally spaced frequencies:

$$X(k) = X(\omega)|_{\omega=\frac{2\pi k}{N}} \tag{2.15}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi k}{N}n}. \tag{2.16}$$

[11] To be precise, $x(n)$ should be written as $x(n, t_0)$ and $X(\omega)$ should be written as $X(\omega, t_0)$, where t_0 denotes the time at which the speech segment is centered. However, when explaining an algorithm in the context of only one speech frame, the time dependency is often assumed. In other contexts (such as signal reconstruction), the time dependency should be explicitly stated. The reader is referred to Section 3.1, which formally describes the short-time Fourier transform.

This is known as the discrete Fourier transform (DFT). This is often expressed as the multiplication of two components:

$$X(k) = |X(k)|e^{j\psi(k)}, \tag{2.17}$$

where $|X(k)|$ is the magnitude spectrum[12] and $\psi(k) = \angle X(k)$ is the phase spectrum. In feature extraction, the phase spectrum is often ignored and features are derived from the power spectrum. The power spectrum of the frame is calculated as:

$$P(k) = |X(k)|^2. \tag{2.18}$$

The fast Fourier transform (FFT) algorithm is used to efficiently compute the DFT. This algorithm reduces the number of computations from $O(N^2)$ to $O(N \log N^2)$.

2.3.5 Filter-bank Analysis

Historically, filter-bank energies (FBEs) were the first method used for ASR feature extraction. Before the development of the FFT by Cooley and Tukey in 1965, FBEs were commonly used for spectral analysis. This method had the advantage that it could be implemented in real time by using a set of analog bandpass filters. The FBEs were obtained by accumulating the energy in each band over short segments of time.

Nowadays, FBEs are determined by applying filters directly on the DFT-derived power spectrum of the signal. The power spectrum has undesirable harmonic fine structure at multiples of the fundamental frequency. Filter-bank analysis reduces this fine structure by calculating the power over frequency bands. The power in each band (or filter) is calculated as the weighted sum of adjacent power values. The filter-bank representation allows incorporation of perceptually-based frequency scales. For example, Fig. 2.7 illustrates a linear filter-bank, a Mel-warped filter-bank and a Bark-warped filter-bank. The responses of the filters are shifted and frequency-warped versions of

[12]Note that $|X(\omega)|$ is the continuous magnitude spectrum and $|X(k)|$ is the discrete magnitude spectrum. We refer to both as simply the magnitude spectrum. The same applies when we mention the phase spectrum.

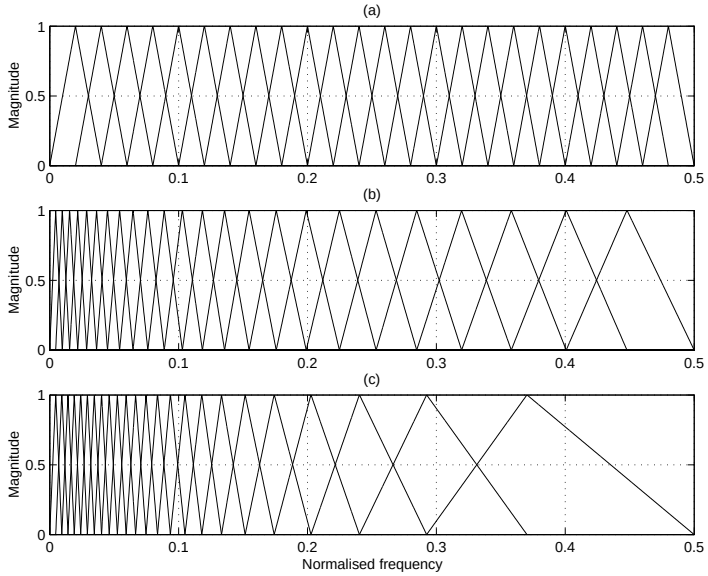


Fig. 2.7: (a) linear filter-bank, (b) Mel-warped filter-bank, and (c) Bark-warped filter-bank. The responses of the filters are shifted and frequency-warped versions of a triangular window. Up to 24 filters are generally used to calculate filter-bank energies.

a triangular window[13]. The magnitude at the center frequency of each filter is unity, linearly decreasing to zero at the center frequencies of the adjacent filters[14]. Note that for the Mel and Bark filter-banks, the center frequencies are equally spaced on their respective perceptual scales. Up to 24 filters are generally used.

There are a number of problems with using FBEs directly for ASR. Since the envelope of the spectrum imposed by the vocal tract is smooth, the FBEs are highly correlated. In addition, a large number of them is generally required. This increases computational time and complexity of the ASR system. Today, FBEs are still used, but as an intermediate step in producing a final set of features.

The smoothing effect of a Mel-spaced filter-bank is demonstrated in Fig. 2.8. We also show the linear prediction spectrum and the power spectrum for the same speech segment.

[13]The triangular filter shape is a very rough engineering approximation to the actual auditory filter shape. Experience has shown that the exact filter shape is not important.

[14]An alternative representation is to normalise each filter so that the area beneath each triangle is equal. However, this makes no significant difference to ASR recognition scores.

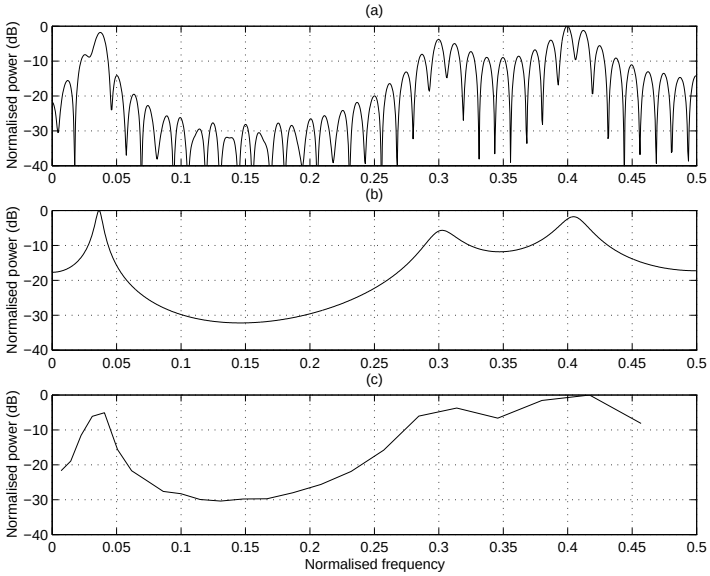


Fig. 2.8: A segment of voiced speech has been analysed to produce: (a) the power spectrum, (b) the LP-derived spectrum, and (c) the FBEs derived from the application of a 24-filter, Mel-warped, triangular filter-bank to the power spectrum. Power has been normalised for comparison purposes. The speech segment was pre-emphasised and weighted with a Hamming window in all cases.

2.3.6 Cepstral Analysis

Recall that the source-filter model of speech production assumes that the speech segment centered at time t_0 is produced when the excitation signal, $e(n, t_0)$, is passed through a linear filter, $h(n, t_0)$. That is, for a small segment of time in which the properties of the speech signal are assumed to be stationary:

$$x(n, t_0) = e(n, t_0) * h(n, t_0). \tag{2.19}$$

The cepstral analysis technique[15] provides an effective method of separating these two components into their sum. The cepstrum[16] of $x(n, t_0)$, denoted by $x_c(m, t_0)$, is the sum of the cepstra of $e(n, t_0)$ and $h(n, t_0)$:

$$x_c(m, t_0) = e_c(m, t_0) + h_c(m, t_0). \tag{2.20}$$

[15]The cepstral analysis technique is a type of homomorphic transformation. Such a transformation maps a convolutional space to a linear space.

[16]The word 'cepstrum' is a result of reversing the first syllable of the word 'spectrum'.

In practice, the cepstrum is computed as the discrete cosine transform (DCT) of the logarithm of the discrete spectral representation, $S(k, t_0)$:

$$x_c(m, t_0) = \frac{1}{K} \sum_{k=0}^{K-1} \log\left(S(k, t_0)\right) \cos\Big(\frac{\pi m(k - 0.5)}{K}\Big), \qquad \text{for } m = 0, 1, \ldots, M-1, \tag{2.21}$$

where, strictly speaking, $S(k, t_0)$ is the magnitude spectrum, $|X(k, t_0)|$. However, it can also be the FBEs. K is the number of spectral samples (in the case of FBEs, usually 24), and M is the total number of cepstral coefficients (usually 13, and $x_c(0, t_0)$ is subsequently discarded). When $S(k, t_0) = |X(k, t_0)|$, m is labeled in units of quefrency (the quefrency has a dimension of time). The periodicity of the excitation source appears as a sharp peak at a quefrency which is the same as the fundamental frequency. The vocal tract characteristics are encoded into the lower quefrencies. The excitation can therefore be removed by windowing (i.e., only keeping the lower cepstral coefficients). This is referred to as liftering.

However, when $S(k, t_0)$ represents the FBEs, there will be very little information in the higher quefrencies because the fine structure is removed during filter-bank analysis. Therefore, the initial motivation for using the cepstrum to remove the excitation component seems to have been accomplished in the filter-bank analysis. Regardless, cepstral analysis is still applied to these FBEs. There are a number of very good reasons for this. The cepstrum of Eq. 2.21 can be broken down into a logarithmic compression operation and a DCT. The logarithmic compression is applied to the FBEs for two reasons: Firstly, their is psychoacoustic evidence that logarithmic compression occurs in the human auditory system (see Section 2.3.1.2). Secondly, the FBEs are based on the linear power, which means the representation only consists of positive values, resulting in a one-sided distribution of the data (Fig. 2.9(a)). Application of the logarithm operation makes the distribution Gaussian-like (Fig. 2.9(b)). This is necessary, since our statistical framework assumes that the feature vectors have such a distribution.

The next step of the cepstrum calculation is the DCT. The DCT is very useful for two reasons. Firstly, it decorrelates the data. Although this decorrelation is not perfect, it is good enough for all intents and purposes. Fig. 2.10(a) shows the covariance matrix for 24

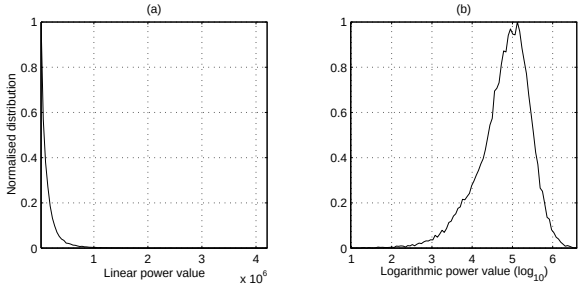


Fig. 2.9: (a) shows the one-sided distribution of the first linear FBE value (from a set of 24 Mel-warped FBEs). (b) shows the distribution after the data vectors are logarithmically compressed. These distributions are derived for the vowel 'aa' from the TIMIT database.

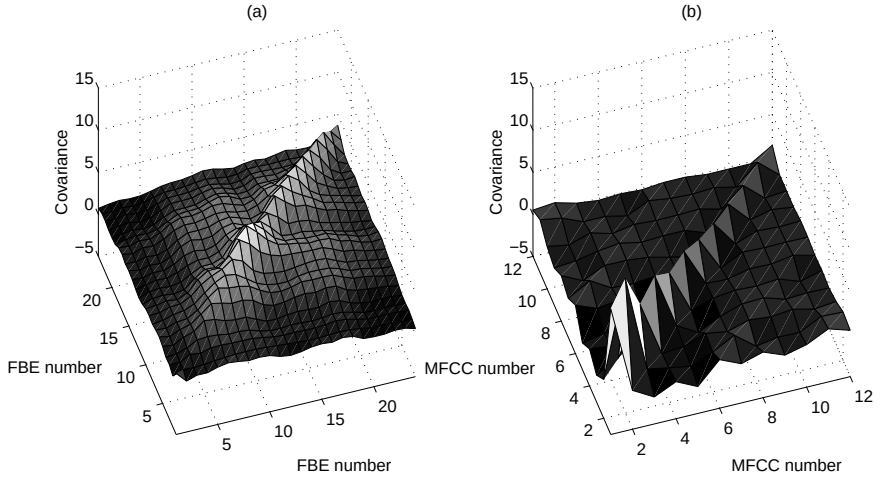


Fig. 2.10: (a) is the covariance matrix for 24 logarithmically compressed Mel-warped FBEs for the vowel 'aa' from the TIMIT database. (b) is the covariance matrix for the (Mel-frequency) cepstral coefficients 1-12 (after a DCT is applied to the log FBEs).

logarithmic compressed FBEs. After a DCT is applied in Fig. 2.10(b), it can be observed that there is very little covariance off the main diagonal. Secondly, the DCT compacts energy. That is, most of the variance in the cepstrum occurs in the lower coefficients. The higher coefficients can be discarded. This makes for an efficient representation. It reduces computation cost and the number of parameters to be estimated in the models.

2.3.6.1 Linear Prediction Cepstral Coefficients

Like the cepstral analysis method, the LP method of analysis also separates the system and the source components. The LP spectrum is a smoothed envelope and represents the vocal tract response, where the residual signal is representative of the source component.

The cepstral transformation, however, is still useful when applied to the LP parameters because it decorrelates them. LPCCs [13] are derived by taking the inverse z-transform of $\ln H(z)$, where $H(z)$ is given by Eq. 2.13. The full derivation can be found in [63]. This results in the following equations which transform the LPC set into a cepstral coefficient set:

$$lpcc(n) = \begin{cases} \ln G & n = 0 \\ a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) c(k) a_{n-k} & 0 < n \leq p \\ \sum_{k=n-p}^{n-1} \left(\frac{k}{n}\right) c(k) a_{n-k} & n > p \end{cases} \tag{2.22}$$

Note that while there is a finite number of LPCs, the number of cepstral coefficients is infinite. However, the cepstrum is a decaying sequence, so a finite number of coefficients are sufficient to approximate it. It is common practice to calculate $c(n)$ for $n = 1, 2, \ldots, 12$.

2.3.6.2 Mel-frequency Cepstral Coefficients

Proposed by Davis and Mermelstein in 1980 [27], MFCCs have consistently been shown to outperform other feature representations for clean speech. MFCCs are also more resistant to noise than conventional LPCCs [81].

MFCC features are calculated as the cepstrum of the Mel-warped filter-bank energies. The process is illustrated in Fig. 2.11. Usually, 24 FBEs are employed and 12 cepstral coefficients (not including the 0^{th} coefficient) are retained.

2.3.7 Energy Measure

It is common to append an energy coefficient to the cepstrum feature vector. The energy is computed as the logarithm of the accumulated frame energy:

$$E_{t_0} = \log\Big(\sum_{n=1}^{N} x^2(n, t_0)\Big). \tag{2.23}$$

In doing so, the 0^{th} cepstral coefficient is discarded because it contains similar information. An analysis of Eq. 2.21 reveals that the 0^{th} coefficient is the sum of the log

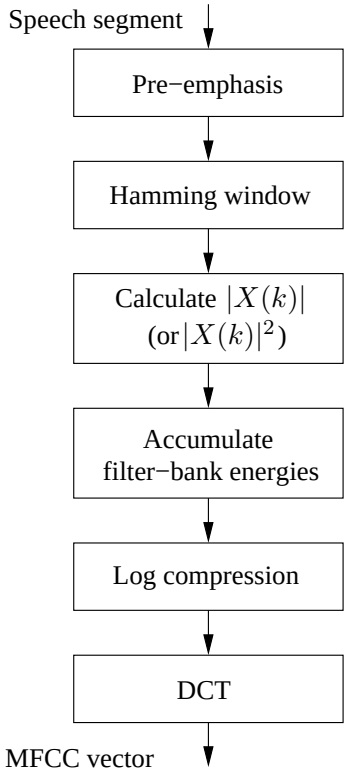


Fig. 2.11: Calculation of MFCC feature vector.

filter-bank energies. Energy is useful since differences in energy are seen among different phonemes. If the speech signal is multiplied by a gain factor, this affects only the energy coefficient. The cepstral coefficients (except for the 0^{th} coefficient) are insensitive to gain factors.

2.3.8 Differential Features

Temporal changes in speech spectra play an important role in perception. This information is captured in the form of velocity coefficients and acceleration coefficients (collectively referred to as differential or dynamic features).

Successive feature vectors of speech are correlated, however, this is not considered in the Hidden Markov model (HMM) framework because HMMs assume independence between frames. The left-right HMM (Fig. 2.12) provides a temporal structure which to some extent models the time-evolution of a speech unit, but in each state the observations are assumed to be independent and identically distributed. This implies that

the spectral-time trajectory of each state is randomly fluctuating around a stationary mean. However, in reality, the spectral-time trajectory clearly has a definite direction as it moves between states. Differential features, first proposed by Furui [39], address this inadequacy and are complementary to the HMM framework. In addition to capturing the time trajectory information, these dynamic features are also less sensitive to slowly varying noise than the static features from which they are calculated.

Differential features are generally not calculated using simple differences because this is too sensitive to random inter-frame variations. Rather, the time derivatives are obtained by linear regression over typically 5 successive frames (where the frame of interest is the center frame). Given that $\vec{o}_i$ is the feature vector at time i (which is usually the 12 cepstral coefficients plus energy), the velocity coefficients are calculated as follows:

$$\Delta\vec{o}_i = \frac{\sum_{l=1}^{L} l\left(\vec{o}_{i+l} - \vec{o}_{i-l}\right)}{2\sum_{l=1}^{L} l^2}, \tag{2.24}$$

where L is the order of the regression ($L = 2$ for 5 frames). Acceleration coefficients are subsequently calculated as a linear regression of the velocity coefficients. Thus, the final length of the feature vector is 39 (12 ceptral coefficients + 1 energy coefficient, giving 13 base coefficients, with Δ and $\Delta\Delta$ coefficients subsequently calculated and appended)[17].

2.4 Acoustic Modeling

Acoustic modeling is based on the assumption that the feature vectors are generated by an underlying multivariate random process. We impose a model structure (HMMs), train the parameters of all the models (using MLE training – see Section 2.2), then use these models to identify previously unseen acoustic vectors.

[17]The energy coefficient is usually written as E. The differential coefficients, Δ and $\Delta\Delta$, are usually written as D and A respectively.

2.4.1 Hidden Markov Modeling

The use of HMMs is a powerful technique for modeling the temporal structure and variability of speech. The majority of ASR systems nowadays, including the system used in this dissertation, use HMMs.

2.4.1.1 HMM Parameters

HMMs describe the sequence of feature vectors, as the output of a stochastic system described by a set of states, a set of state transition probabilities, and probability distributions for each state. The parameters are enumerated as follows:

1. O = sequence of observed feature vectors, where the observation at time t is $\vec{o}_t$.
2. Q = the state sequence vector, where the state at time t is q_t.
3. s_i = the state identifier, where $1 \leq i \leq N$.
4. A = the state transition probability matrix, where the probability of transition between state s_i and s_j is denoted as a_{ij}.
5. $B = (b_1(\vec{o}_t), b_2(\vec{o}_t), \ldots, b_N(\vec{o}_t))$, where $b_i(\vec{o}_t)$ is the feature vector probability distribution in state s_i. Note that this can be of any distribution type, discrete or continuous. In ASR, we commonly assume a continuous multiple mixture Gaussian distribution for each state.
6. π_i = the probability of state s_i being the first state entered.
7. $\theta = (A, B, \pi)$ is used to indicate the parameter set of the model.

The HMM reduces a non-stationary process into a piecewise stationary process, where each state is representative of a stationary region. The HMM structure employed in this dissertation is illustrated in Fig. 2.12. It is used to model a triphone[18] (i.e., a context dependent phoneme). The model consists of three emitting states which represent three discrete temporal regions of the triphone. The outer two states are null

[18]The number of states reflect the duration and complexity of the sound being modeled. A triphone, being an elementary sound unit, is usually only modeled using three emitting states.

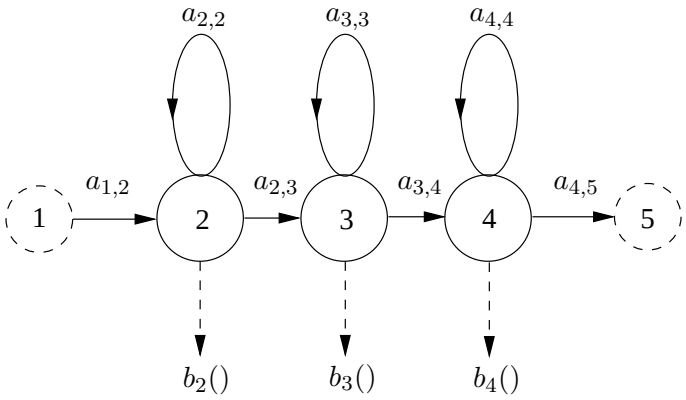


Fig. 2.12: Five state, left-to-right, HMM (adapted from [151]).

states. They allow multiple HMMs to be chained together. Except for those values shown, the remaining state transition probabilities are zero. This imposes a left-to-right structure, ensuring that all states are visited in turn.

2.4.1.2 Determining the Observation Sequence Probability

Consider the following feature vector observation sequence:

$$O = (\vec{o}_1, \vec{o}_2, \ldots, \vec{o}_T), \tag{2.25}$$

where T is the total number of feature vectors. The corresponding state sequence is:

$$Q = (q_1, q_2, \ldots, q_T). \tag{2.26}$$

Given the model parameters, θ, we wish to compute the probability of the observation sequence. If we assume the state sequence is known, the probability of the observation sequence is:

$$\begin{aligned} P(O \mid Q, \theta) &= \prod_{t=1}^{T} P(\vec{o}_t \mid q_t, \theta) && (2.27) \\ &= b_1(\vec{o}_1) b_2(\vec{o}_2) \ldots b_T(\vec{o}_T). && (2.28) \end{aligned}$$

The probability of the state sequence is:

$$P(Q \mid \theta) = \pi_{q_1} a_{1,2} a_{2,3} \dots a_{T-1,T}. \tag{2.29}$$

The probability that the observation sequence is produced by a specific state sequence, given the model, is

$$P(O, Q \mid \theta) = P(O \mid Q, \theta) P(Q \mid \theta). \tag{2.30}$$

It follows that the probability of the observation sequence can then be determined by summing $P(O, Q \mid \theta)$ over all possible state sequences (since multiple state sequences could possibly produce the same observation vector):

$$\begin{aligned} P(O \mid \theta) &= \sum_{all\ Q} P(O \mid Q, \theta) P(Q \mid \theta) && (2.31) \\ &= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_1(\vec{o}_1) a_{1,2} b_2(\vec{o}_2) \dots a_{T-1,T} b_T(\vec{o}_T). && (2.32) \end{aligned}$$

This equation is computationally expensive, requiring $N^T - 1$ additions and $(2T-1)N^T$ multiplications.

The forward-backward method is an efficient procedure by which to calculate $P(O \mid \theta)$. It is based on the first-order Markov assumption, which states that the probability of an observation vector is only dependent on the current state. The forward probability, $\alpha_t(i)$, and the backward probability, $\beta_t(i)$, are defined as:

$$\begin{aligned} \alpha_t(i) &= P(\vec{o}_1, \vec{o}_2, \dots, \vec{o}_t, q_t = s_i \mid \theta) && (2.33) \\ \beta_t(i) &= P(\vec{o}_{t+1}, \vec{o}_{t+2}, \dots, \vec{o}_T \mid q_t = s_i, \theta). && (2.34) \end{aligned}$$

The forward variable is the probability that the partial observation sequence is produced and the state at time t is s_i. The backward variable is the probability of the partial observation sequence from time $t+1$ to T and the state at time t being s_i.

The forward algorithm is initialised as follows:

$$\alpha_1(i) = \pi_i b_i(\vec{o}_1), \qquad 1 \leq i \leq N. \tag{2.35}$$

Each successive value can then be inductively determined as follows:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)a_{ij}]b_j(\vec{o}_{t+1}). \tag{2.36}$$

The probability of the complete observation sequence is determined by summing $\alpha_T(i)$ over each state:

$$P(O \mid \theta) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.37}$$

The backward probability $\beta_t(i)$ can similarly be used to determine $P(O \mid \theta)$. The initial conditions are:

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N. \tag{2.38}$$

Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(\vec{o}_{t+1})\beta_{t+1}(j), \tag{2.39}$$

Termination:

$$P(O \mid \theta) = \sum_{i=1}^{N} \beta_1(i). \tag{2.40}$$

The computation of $P(O \mid \theta)$ using either of the above methods requires $O(N^2T)$ calculations.

2.4.1.3 Determining the Optimal State Sequence

Given the observation sequence, O, and the model parameters, θ, we need to determine a corresponding state sequence that is optimal in some sense. The Viterbi algorithm is a dynamic programming method that is used to find the single best state sequence path. This algorithm maximises $P(Q \mid O, \theta)$, which is equivalent to maximising $P(Q, O \mid \theta)$.

To begin, we need to define the following quantity:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} [P(q_1, q_2, \ldots, q_t = s_i, \vec{o}_1, \vec{o}_2, \ldots, \vec{o}_t \mid \theta)], \tag{2.41}$$

which is the best score along a single path at time t, accounting for the first t observations

and ending in state s_i. It is initialised as follows:

$$\delta_1(i) = \pi_i b_i(\vec{o}_1) \qquad 1 \leq i \leq N, \tag{2.42}$$

and successive values are recursively determined:

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] b_j(\vec{o}_{t+1}) \qquad 2 \leq t \leq T \text{ and } 1 \leq j \leq N. \tag{2.43}$$

We also need to define an array which holds the state numbers that maximise Eq. 2.43:

$$\psi_1(i) = 0 \tag{2.44}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \qquad 2 \leq t \leq T \text{ and } 1 \leq j \leq N. \tag{2.45}$$

In order to terminate the recursion, we define two more parameters:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{2.46}$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)], \tag{2.47}$$

where P^* is the probability of the optimal state sequence and q_T^* is the most likely final state. The optimal state sequence can be retrieved from ψ, such that:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \ldots, 1. \tag{2.48}$$

2.4.1.4 Model Parameter Estimation

There is no analytic or optimal way to determine the parameters of a HMM. The Baum-Welch method is an iterative procedure based on the expectation-maximisation algorithm [17] that determines the model parameters such that $P(O \mid \theta)$ is locally maximised.

We start by defining the following variable, which is the probability of being in state

s_i at time t, given the observation sequence and the model:

$$\gamma_t(i) = P(q_t = s_i \mid O, \theta), \tag{2.49}$$

which can be expressed in terms of the forward and backward variables:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \theta)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \tag{2.50}$$

where the denominator of the fraction serves to normalise $\gamma_t(i)$, making it a probability measure.

We also need to define $\xi_t(i,j)$, the probability of being in state s_i at time t and state s_j at time $t+1$, given the observation sequence and the model:

$$\begin{aligned}
\xi_t(i,j) &= P(q_t = s_i, q_{t+1} = s_j \mid O, \theta) && (2.51)\\
&= \frac{\alpha_t(i)a_{ij}b_j(\vec{o}_{t+1})\beta_{t+1}(j)}{P(O \mid \theta)} && (2.52)\\
&= \frac{\alpha_t(i)a_{ij}b_j(\vec{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(\vec{o}_{t+1})\beta_{t+1}(j)}, && (2.53)
\end{aligned}$$

where the denominator makes $\xi_t(i,j)$ a probability. Note that the relationship between $\gamma_t(i)$ and $\xi_t(i,j)$ is given by:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{2.54}$$

The Baum-Welch re-estimation formulas are as follows:

$$\begin{aligned}
\bar{\pi}_i &= \gamma_1(i) && (2.55)\\
\bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(j)} && (2.56)\\
\bar{b}_j(\vec{k}) &= \frac{\sum_{t=1(s.t.\vec{o}_t=\vec{v_k})}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, && (2.57)
\end{aligned}$$

where the sum in the numerator of Eq. 2.57 is only calculated when the symbol $\vec{v_k}$ is observed in state s_j. The model is seeded with a set of parameters, then the new

parameters form the seeds for the next iteration. That is, we iteratively use $\bar{\theta}$ in place of θ and repeat the calculation. The probability of the training observation sequences is improved after every iteration. Refer to Rabiner's paper for an excellent discussion on HMMs [121]. Huang et al. [63] also provide a comprehensive review on the subject.

2.4.2 Linguistic Unit Size and Parameter Sharing

The HMMs can be used to represent any level of linguistic unit, such as words, syllables, phonemes, etc. The choice as to what level of model is required depends on the vocabulary size and the amount of training data. For example, whole word models are commonly used for a small vocabulary system. For such systems it is easy to obtain a sufficient number of instances for each word to train the models. However, as the vocabulary size increases it becomes more difficult to obtain the required amount of data to adequately train the models. Sub-word models, such as phonemes, are more appropriate for large vocabulary systems. Phonemes are generalisable because they are the building blocks for words, making it easier to acquire training data for them.

The popular choice for HMMs is at the triphone level (i.e., context dependent phoneme). This is because triphones are very specific units and can be joined together to form any word. However, there are approximately 100,000 triphones in English (45^3). Such a number of models calls for an inordinate amount of training data. Consider this example: if each HMM has 3 states, with 10 mixtures per state, diagonal covariances and a feature vector size of 39, this results in 390 means, 390 variances, and 10 Gaussian weights per state (790 parameters per state), by 3 states (2370 parameters per model), by 100,000 models gives around 237 million parameters. Once again, triphone level models seem impractical due to the lack of adequate training data. However, since many of the triphones are similar, parameter sharing is often employed to work around this problem [64].

Many phonemes have similar effects on neighbouring phonemes. These similar contexts can be trained with the same training data. Let us consider each triphone at a state level. There are three emitting states. Essentially, the first and last emitting state account for the contextual information, while the middle state is representative of

the true identity of the phoneme. It then seems logical to train the middle state of all allophones[19] of a given phoneme with the same data. Further still, if the effects of one left-context are similar to the effects of another left-context on the same phoneme, then we may also choose to train those states with the same training data. The same is true for right-contexts. This process is also referred to as clustering and is often performed by using a question set which asks a set of binary questions in a hierarchical manner in order to cluster similar states into senones[20] for training [151].

2.4.3 Embedded Training for Continuous ASR

For continuous ASR, it is common to perform embedded training. This involves linking the models end-to-end in the order specified by the training transcriptions. For example, if the models are triphones then the transcriptions must be written in terms of their triphone pronunciations. The initial parameters (i.e., the seed values for the Baum-Welch formulae) for all the models are set to the global mean and variance of all the speech training data; this is called a flat start. Another approach is to start with models that have been trained from another system. Effectively, one large HMM is formed for each of the transcribed sentences. All the models are subsequently trained in parallel.

2.5 Language Modeling

As explained in Section 2.2, $\log P(\Lambda)$ introduces apriori information. Depending on the linguistic unit type, this term can introduce information at a number of levels. For example, for phoneme HMMs $\log P(\Lambda)$ enforces constraints (via the lexicon[21]) about what phonemes can follow other phonemes. $\log P(\Lambda)$ also enforces constraints at the word level. These word-level constraints are dictated by the language model.

The language model specifies the most likely subset of words that may be used at any one time. It compensates for the deficiencies in the acoustic pattern matching, serving

[19] An *allophone* is one contextual realisation of a given phoneme.
[20] A *senone* is a set of similar Markov states.
[21] The lexicon contains a list of the words present in the vocabulary and their breakdown into the level of abstraction modeled by the HMMs.

to eliminate unlikely candidates in addition to speeding up the recognition process[22]. Without the language model, the entire vocabulary must be considered at every decision point.

There are many types of language models, all of which (to differing degrees) include syntactic, semantic and pragmatic information. The models differ in the way that they reduce the search space. The more the search space is reduced, the less flexible the system will be. Such constraints on flexibility are acceptable for very specific tasks, however, for unconstrained continuous speech recognition we want as much flexibility as possible.

The most common type of language model used for ASR is the stochastic N-gram model. The prior probability of a sequence of words $W = \{w_i\}_{i=1}^{M}$ is factorised as follows:

$$P(W) = P(w_1, w_2, \ldots, w_M) = P(w_1) \prod_{i=2}^{M} P(w_i \mid w_{i-1}, \ldots, w_1). \tag{2.58}$$

The estimation of this large set of probabilities requires an almost limitless amount of training data. This is unfeasible. We can overcome this by using an N-gram model. The conditional probability of a word w_i at i is evaluated by considering only the $N-1$ preceding words. The probability of Eq. 2.58 is evaluated by considering that:

$$P(w_i \mid w_{i-1}, \ldots, w_1) \approx P(w_i \mid w_{i-1}, \ldots, w_{i-N}). \tag{2.59}$$

The larger that N is, the more training data that is required. Generally, only unigram ($N = 1$), bigram ($N = 2$) and trigram ($N = 3$) models are considered.

2.6 Decoding

Decoding involves the calculation of Eq. 2.5. That is, given a set of observed feature vectors, the set of all models, the lexicon, and the language model, we need to compute the most likely sequence of models that generated the feature vectors. The sequence of

[22]This works well until the speaker goes outside those constraints set by the language model. In such cases, the system will fail because the actual words spoken will not be presented as possible candidates to the pattern matching system.

models can then be mapped to a sequence of words. This is a dynamic process which involves considering many models at each point in time and eliminating unlikely model sequences on the fly.

The calculation of the posterior probability component of Eq. 2.5 is essentially determined by placing the HMMs end-to-end to form one composite HMM (again, this is a dynamic process, but it helps to think of the process like this). The probability that this composite HMM generated the observation sequence is then determined. Strictly speaking, this should be calculated using Eq. 2.31 (or using the more efficient forward-backward algorithm in Eq. 2.37). In practice, however, we use the Viterbi algorithm, which determines the most likely state sequence through the model as well as the corresponding probability for that state sequence. It is this state sequence probability that is used in place of the observation sequence probability. That is, $P(O|\Lambda, \Theta_\Lambda)$ is approximated by the probability of the most likely state sequence through the sequence of models (in this case, the composite model). In effect, the summation in Eq. 2.31 is replaced by maximisation. This works because the maximum term is usually the only significant term in the summation [121].

Also in practice, Eq. 2.5 is slightly modified by the addition of a grammar scale factor, s, and a word-insertion log probability (or fixed transition penalty), p:

$$\widetilde{\Lambda} = \arg\max_{\Lambda \in M} \Big(\log P(O|\Lambda_\Lambda) + s \log P(\Lambda) + p \Big). \tag{2.60}$$

p is only added when an hypothesised word transition is made. Both s and p are fudge factors that are set empirically to maximise word accuracy on development test utterances. Their combined purpose is to regulate the number of deletion and insertion errors.

2.7 Evaluating Performance

Performance evaluation involves a dynamic algorithm which best aligns the hypothesised word sequences and the actual word sequences (supplied with the test utterances). After

alignment of all test sequences, word accuracy is calculated by:

$$Accuracy = \frac{N_t - N_s - N_d - N_i}{N_t}.100\%, \tag{2.61}$$

where N_t is the total number of words, N_s is the number of substitutions, N_d is the number of deletions, and N_i is the number of insertions. A substitution occurs when a word from the actual word sequence is replaced by a different word in the hypothesised word sequence. A deletion occurs when a word from the actual word sequence is skipped in the hypothesised word sequence. An insertion occurs when an extra word appears in the hypothesised word sequence.

2.8 Robustness

State-of-the-art ASR systems provide very impressive recognition performance. However, performance declines dramatically in the presence of adverse variabilities. These variabilities include various sources of additive noise, the effects of linear filtering, as well as variabilities and noises introduced by the speaker. Additive noise can be either stationary or non-stationary, such as door slams, passing cars, air conditioners, music, and other talkers in the background. This noise may ultimately influence how the speaker talks (Lombard effect). Linear filtering effects are introduced by the room acoustics (i.e., echos and reverberation) and also by the electronic equipment used to capture the speech. Speaker variabilities include varying pronunciation of words, non-linguistic sounds such as lip smacks, coughs, etc. The distance of the speaker to the microphone is also of concern; gain variations occur as the speaker moves toward and away from the microphone.

Robustness is a term used to describe an invariance or graceful degradation in system performance in the presence of the aforementioned variabilities. The greater the invariance of an ASR system to such variabilities, the more we say the system is robust. Robustness is important because users of practical systems will demand a continued high performance in a wide variety of acoustic environments.

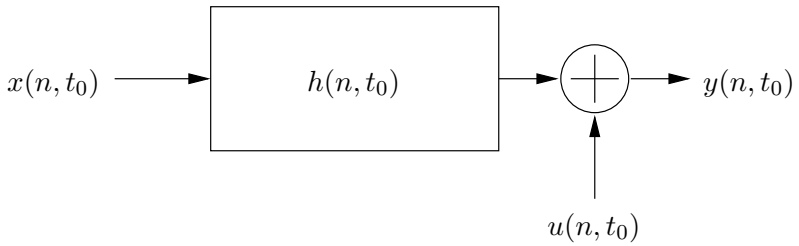


Fig. 2.13: Simplified model of the environment. $x(n, t_0)$ represents the clean speech segment centered at t_0, $h(n, t_0)$ represents convolutional/channel effects, $u(n, t_0)$ represents additive noise, and $y(n, t_0)$ is the resulting acoustic signal segment from which features are to be extracted.

Before discussing common strategies for robustness, it is best to first analyse how the properties of the cepstrum change in response to such variabilities (since the cepstrum is the most common feature representation). The cepstral representation offers many advantages, such as a compact orthogonal feature set, with level independence and good pitch attenuation [102]. As will be shown, however, the cepstral representation is not very robust.

2.8.1 The Effect of Noise on the Cepstrum [62]

It is common in ASR to characterise the environment by the model shown in Fig. 2.13. This is a widely used model, categorising noise as either additive or convolutional (neglecting speaker variabilities). The model assumes that the speech and contributing noise sources are stationary over the duration of the segment. According to the model:

$$y(n, t_0) = x(n, t_0) * h(n, t_0) + u(n, t_0). \tag{2.62}$$

The power spectrum of $y(n, t_0)$ is determined by taking a $2N$-point DFT (where the length of the frame is N-points) and then squaring the magnitude spectrum as follows:

$$\begin{aligned} |Y(k, t_0)|^2 &= \Big[X(k, t_0)H(k, t_0) + U(k, t_0)\Big]\Big[X^*(k, t_0)H^*(k, t_0) + U^*(k, t_0)\Big] && (2.63) \\ &= |X(k, t_0)|^2|H(k, t_0)|^2 + |U(k, t_0)|^2 + Z(k, t_0), && (2.64) \end{aligned}$$

where $k = 0, 1, \ldots, N-1$ and $Z(k, t_0)$ is a cross-term defined by:

$$|Z(k,t_0)|^2 = X(k,t_0)H(k,t_0)U^*(k,t_0) + X^*(k,t_0)H^*(k,t_0)U(k,t_0) \quad (2.65)$$

$$= 2|X(k,t_0)||H(k,t_0)||U(k,t_0)|\cos(\theta(k,t_0)), \quad (2.66)$$

where $\theta(k, t_0)$ is the phase angle between the noise and the filtered signal. The expected value of $Z(k, t_0)$ is zero because $x(n, t_0)$ and $u(n, t_0)$ are independent. In practice, this term is not zero; however, it approaches zero when an average is taken over a range of frequency bins. This averaging process is used in filter-bank analysis (see Section 2.3.5).

$$|Y(k_i,t_0)|^2 = \sum_{k=k_i-\delta_i}^{k_i+\delta_i} w(k)|Y(k,t_0)|^2 \quad (2.67)$$

$$|Y(k_i,t_0)|^2 = |X(k_i,t_0)|^2|H(k_i,t_0)|^2 + |U(k_i,t_0)|^2 \quad (2.68)$$

where $w(k)$ is a weighting function (usually triangular), k_i and $2\delta_i$ are the filter center frequency and bandwidth respectively, and $i = 0, 1, \ldots, K-1$, where K is the number of filters.

After performing some algebraic manipulations, we take the logarithm of both sides:

$$|Y(k_i,t_0)|^2 = |X(k_i,t_0)|^2|H(k_i,t_0)|^2\left(1 + \frac{|U(k_i,t_0)|^2}{|X(k_i,t_0)|^2|H(k_i,t_0)|^2}\right) \quad (2.69)$$

$$\log|Y(k_i,t_0)|^2 = \log|X(k_i,t_0)|^2 + \log|H(k_i,t_0)|^2$$
$$+ \log\Big(1 + \exp\big(\ln|U(k_i,t_0)|^2 - \log|X(k_i,t_0)|^2 - \log|H(k_i,t_0)|^2\big)\Big). \quad (2.70)$$

If $\vec{y}_{t_0}$ is the cepstral feature vector corresponding to $y(n, t_0)$ (calculated by applying the DCT to $\log|Y(k_i, t_0)|^2$), then Eq. 2.70 can be written in terms of the cepstrum:

$$\vec{y}_{t_0} = \vec{x}_{t_0} + \vec{h}_{t_0} + C\ln\Big(1 + \exp\big(C^{-1}(\vec{u}_{t_0} - \vec{x}_{t_0} - \vec{h}_{t_0})\big)\Big), \quad (2.71)$$

where C is the DCT matrix.

The clean speech cepstrum, $\vec{x}_{t_0}$, is assumed to have a Gaussian distribution. How-

ever, Eq. 2.71 demonstrates that the noisy speech cepstrum $\vec{y}_{t_0}$ is no longer Gaussian because of the non-linear combination of $\vec{x}_{t_0}$ with $\vec{h}_{t_0}$ and $\vec{u}_{t_0}$. As noise decreases, the non-linear term in Eq. 2.71 tends toward zero and the effect of additive noise becomes negligible. That is,

$$\vec{y}_{t_0} \approx \vec{x}_{t_0} + \vec{h}_{t_0}, \tag{2.72}$$

for high SNRs. Consequently, at high SNRs the effect of the channel, $\vec{h}_{t_0}$, can be subtracted by a process called cepstral mean subtraction (see Section 2.8.2.3) and $\vec{y}_{t_0} \approx \vec{x}_{t_0}$.

It can be observed in Fig. 2.14 how the value of a cepstral coefficient changes over the length of an utterance at various values of SNR. The addition of white noise reduces the variance (or dynamic range) and dramatically changes the values of the cepstral coefficient. Alternatively, we can examine the distributions directly. Fig. 2.15 demonstrates how the distribution of a cepstral coefficient changes with additive white noise. Specifically, we can see that the addition of white noise results in a mean shift and a variance reduction. A detailed analysis has been performed by Openshaw and Mason [102], where they demonstrated that cepstral distributions can also become multi-modal in noise.

2.8.2 Strategies for Robustness

To achieve optimum performance, systems are often trained in the environment in which they will be used [26]. However, retraining systems for each particular environment is expensive because a lot of time is required to collect the training data and to retrain the system. In addition, retraining in a particular environment also has its limitations. The noise introduces more variability in to the training data. As a consequence, the variance of the distributions of the sound classes also increases. This results in increased classification errors because the classes are harder to separate due to the increased overlap between distributions. Also, if the noise is non-stationary, training on such noise will not help because it may not be representative of the noise in the test speech. Thus, more sophisticated methods for robustness must be used.

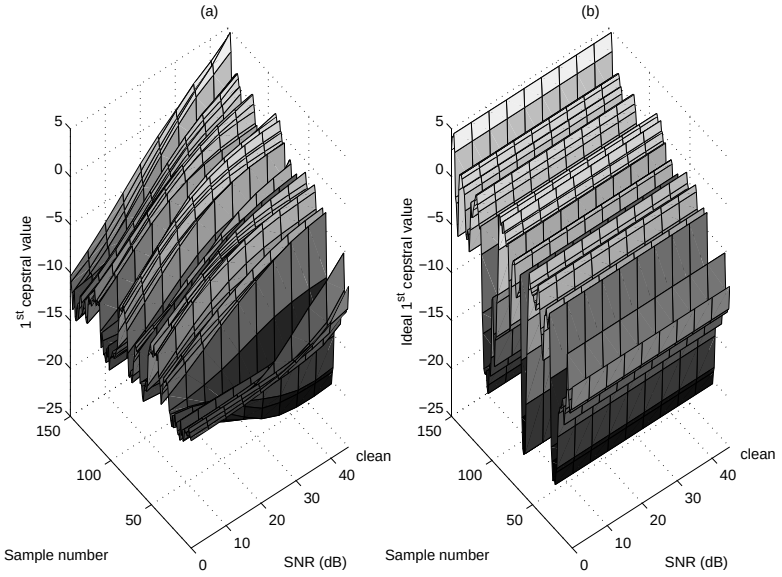


Fig. 2.14: (a) shows how the first Mel-frequency cepstral coefficient varies over the length of an utterance at SNRs from ∞ down to 0 dB. (b) shows an ideal, invariant, response to the white noise (a fictitious plot provided for comparison purposes).

Strategies for robustness range from narrow to generally applicable solutions. The methods of robustness can be classified into the following categories, which are further discussed in subsequent subsections[23]:

1. Speech enhancement: Retrieval of the original speech from corrupt speech is attempted before parameterisation.

2. Robust feature extraction: Designing features that are invariant to the noise. The design of the feature extractor is usually motivated by the inner workings of the human auditory system.

3. Mapped features: Noisy features are mapped to a 'cleaner' feature space.

4. Noise compensated models: The trained models are adapted to suit the testing conditions.

We briefly describe some of these approaches in the following sections.

[23]There are many ways to classify approaches to robustness.

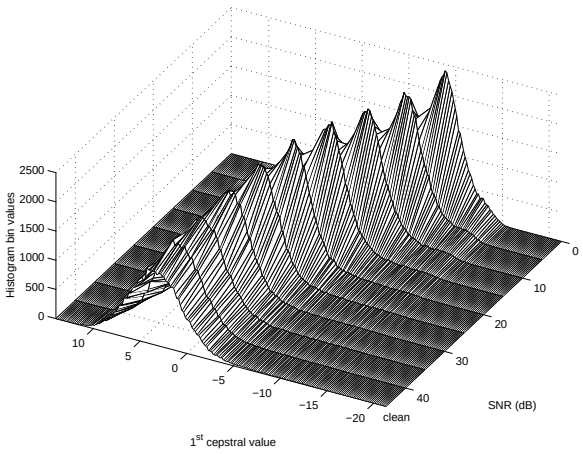


Fig. 2.15: The distribution of the first Mel-frequency cepstral coefficient for the vowel 'aa' from the TIMIT database in various amounts of additive white noise.

2.8.2.1 Speech Enhancement

In the context of ASR, speech enhancement techniques are commonly referred to as signal space techniques. Such techniques are intended to recover the speech waveform embedded in noise. The idea behind using speech enhancement is that if we can enhance the speech, we can assume it is clean, then use the same system (i.e., feature type and models) for recognition. In general, however, traditional speech enhancement offers little improvement to recognition [84]. ASR performance is often improved in combinations with other techniques.

Perhaps the mother of all speech enhancement techniques is spectral subtraction (SS). This technique, introduced by Boll [19] in 1979, does exactly what its name implies. Assuming additive noise is uncorrelated with speech, SS attempts to remove its effect by subtracting an estimated noise spectrum from the corrupt spectrum to produce an approximated clean spectrum. The noise estimate is obtained by averaging the periodogram over a number of noise-only frames (estimated during non-speech activity). Therefore, a robust procedure to detect these non-speech frames is required. The output speech is significantly less noisy, although there is a presence of musical noise. This is mainly due to the fact that SS depends on the assumption of independence of spectral estimates across frequencies. Also, the average noise spectrum may have values greater

than the noisy signal magnitude spectrum. Thus, when the noise spectrum is subtracted, negative magnitude values may result. These values must be zeroed. In [36], Flores and Young have compensated for the introduced distortion by adapting the models using the parallel model combination technique (see Section 2.8.2.4).

Singular value decomposition (SVD) [28,66] is another such technique that assumes the noise is additive and uncorrelated with the speech (like SS). In this technique, the signal is reconstructed from a subset of the singular values which are retained from a decomposition of an over determined data matrix. The singular values that are ignored are assumed to be associated with the noise. As with SS, SVD seems to improve the SNR of the speech, but this does not translate to an improvement in ASR performance.

A number of adaptive filtering techniques have also been used for speech enhancement. Wiener filtering and Kalman filtering techniques assume an autoregressive model of speech production in order to design a noise suppression filter. Once again, these techniques have succeeded in improving the SNR of the noisy speech but not necessarily its intelligibility or recognition accuracy. Comb filtering [132] effectively multiplies the frequency response of the observed signal by a sequence of Dirac functions whose interval is determined by the period of the speech signal. The problem with this method is that it is difficult to make accurate estimates of the period when the speech is noisy and also when the fundamental frequency varies rapidly [37].

Noise floor normalisation [71] is a primitive form of noise reduction. It involves measuring the noise in a non-speech region. Values of the speech spectrum that are below this average noise spectrum are set to this value.

Artificial neural networks (ANNs) [78] have been used to remove noise in the signal space [135]. To train the ANN, segments of corrupt speech are mapped to segments of clean speech. Essentially, the characteristics of the noise are learned by the ANN. The trained ANN is subsequently used to clean unseen noisy speech.

One class of techniques involves the use of multiple microphones [67, 98, 144]. The microphones are positioned so that the combination of the signals can be used to isolate the speech. However this technique depends largely on the position of the microphones in order to cancel the noise [66]. In our work, we are concerned with techniques that

require only a single microphone.

2.8.2.2 Robust Feature Extraction

Robust feature extraction attempts to derive a feature set which provides a consistent representation in the face of undesired variabilities. The idea is that, if a feature set is robust, then the models that were trained for the clean speech are equally valid for the corrupt speech.

Robustness is generally attempted by incorporating as much as possible about the human auditory system into a model, which is then used to derive these features. Such models are often referred to as auditory models. Among the best known auditory models are the joint synchrony/mean-rate model [130], the Ensemble Interval Histogram (EIH) model [45], and the zero-crossing peak amplitude (ZCPA) model [70]. There are many other types of auditory models [42,136]. In general, the models show an improvement in noisy conditions, however, their performance does not match MFCC features for clean speech. In addition, auditory models are computationally expensive. Current auditory models are simplifications, therefore their poor performance must not discount the use of them altogether. A better understanding of the auditory system in years to come may result in less simplified models which may provide promising results.

There are many more feature sets that have been proposed in the context of their robustness. Some of these include sub-band autocorrelation analysis (SBCOR) features [68], sub-band spectral centroid histograms (SSCH) [40], and perceptual harmonic cepstral coefficients (PHCC) [48]. However, as with their detailed auditory counterparts, these features do not provide performance comparable to MFCCs in clean conditions.

A well known feature set that is competitive with the performance of MFCCs are perceptual linear prediction (PLP) coefficients [56]. PLP is a method of deriving a more auditory-like spectrum based on LP analysis of speech. The output of this analysis is a set of cepstral coefficients. Other LP-based methods include short-term modified coherence (SMC) [82] and one-sided autocorrelation linear predictive coding (OSALPC) [59]. Again, none of these methods is universally superior to MFCCs.

2.8.2.3 Mapped Features

These techniques, also referred to as feature space techniques, involve the application of some kind of transformation which maps the corrupt vectors to clean vectors. Various techniques are presented in the literature.

ANNs have been used to map noisy cepstral vectors to clean cepstral vectors [16, 152]. These transformations are promising because the mapping is performed without knowledge of the distortion. Huang has used an ANN to transform speech data between two speakers so that models of the reference speaker can be used for recognition [61].

Vocal tract normalisation (VTN) is used to reduce inter-speaker variability, mainly caused by the differences in vocal-tract lengths. Essentially a warping factor is calculated for each speaker (determined from a single input utterance). There are a number of methods used to calculate the warping factors. The warping is intended to shift the formants to the locations in the spectrum where they would have been if the spectrum was produced by the reference speaker (the reference speaker is used to train the models). This warping factor is applied on the speech spectra before further processing is applied (such as filter-bank energy calculation and DCT).

Linear discriminant analysis (LDA) [30] finds a transformation matrix which, when applied to the feature vectors, minimises the within-class scatter and maximises between-class scatter (i.e., class separability). However, the transformation is data dependent. Therefore, LDA is useful to improve the performance at a particular SNR. It does not provide robustness.

A number of techniques have been designed specifically for additive noise, such as probabilistic optimum filtering (POF) [93] and codeword-dependent cepstral normalisation (CDCN) [2]. However, these methods depend on the availability of a stereo database, which may be hard to come by.

Cepstral mean subtraction (CMS) is a normalisation method used to remove channel effects. It amounts to the application of a high-pass filter to the time-trajectory in the cepstral domain. It is very simple, yet effective. Thus, the following paragraphs are devoted to its description. A similar technique, called relative spectral processing (or

RASTA) [58], involves the application of a bandpass filter to the time trajectory in the log-spectral domain. The theory behind the use of such a filter comes from the observation that temporal properties of the environment are usually quite different to the temporal properties of speech. RASTA and CMS provide for similar recognition performance. We will only discuss CMS in detail here, since we employ this in the experiments for this dissertation.

Cepstral Mean Subtraction In Section 2.8.1, we showed that convolutional noise (i.e., the channel component) becomes additive in the cepstral domain. CMS uses this property to remove the channel effect. First proposed by Atal in 1974 [13], CMS[24] is one of the most effective and popular environmental compensation algorithms.

Let $\{\vec{x}_t\}_{t=1}^{T}$ represent the cepstral features extracted from the clean training data. T is generally the number of feature vectors extracted from one utterance. The mean of these vectors is calculated by:

$$\vec{x}_{mean} = \frac{1}{T}\sum_{t=0}^{T-1}\vec{x}_t. \tag{2.73}$$

This mean is then subtracted from each vector, $\vec{x}_t$, to obtain a set of mean-subtracted cepstral vectors:

$$\vec{x}_{t,CMS} = \vec{x}_t - \vec{x}_{mean}. \tag{2.74}$$

Ignoring the effect of additive noise, a corrupt cepstral vector is the result of adding the channel cepstral vector, $\vec{h}_t$, to the desired clean cepstral vector, $\vec{x}_t$. That is:

$$\vec{y}_t = \vec{x}_t + \vec{h}_t. \tag{2.75}$$

By assuming that the channel response is constant, the mean of the corrupt vectors, $\vec{y}_{mean}$, can be expressed as:

$$\vec{y}_{mean} = \vec{x}_{mean} + \vec{h}, \tag{2.76}$$

[24] Also referred to as cepstral mean normalisation (CMN).

and its normalised cepstrum is given by:

$$\begin{aligned} \vec{y}_{t,CMS} &= \vec{y}_t - \vec{y}_{mean} \\ &= (\vec{x}_t + \vec{h}) - (\vec{x}_{mean} + \vec{h}) \qquad (2.77) \\ &= \vec{x}_t - \vec{x}_{mean} \\ &= \hat{x}_{t,CMS} \qquad (2.78) \end{aligned}$$

CMS completely removes the effect of the channel, but only if the assumption that the channel response remains constant is correct. The procedure is performed on both training and testing data so that it is the mean-subtracted feature vectors that are compared for recognition. CMS has no effect on temporal derivatives, thus dynamic features are appended to the feature set after the static features are normalised.

It has been empirically determined that at least two seconds of speech are required in order to attain a reasonable estimation of the mean [62]. If the utterance is too short, the mean may be similar to stationary sounds within the utterance, thus most of the relevant information needed for recognition will be removed.

CMS can be viewed as subtracting the output of a low-pass filter, where all T coefficients have a value of $1/T$, from each feature vector. The subtraction of the mean from the feature set can alternatively be interpreted as a high-pass filtering operation. The above derivation of CMS imposes the constraint that the complete utterance is required before the mean can be calculated. This is because the length of the high-pass filter is as long as the utterance. CMS needs to be modified so it can be used in a real-time system, that is, we must reduce the length and alter the coefficients of the high-pass filter so the vectors can be normalised on the fly. An exponential filter is commonly used for real-time CMS:

$$\vec{x}_{t,CMS} = \alpha\vec{x}_t + (1-\alpha)\vec{x}_{t-1}. \qquad (2.79)$$

2.8.2.4 Noise Compensated Models

In this approach to robustness, the mismatch between the training and testing conditions is corrected in the model space. The major disadvantage with these techniques is that the models need to be re-trained/adapted for different testing environments. Also, training noise models and integrating noise statistics into the clean models is only effective as long as the noise does not induce the Lombard reflex.

One simple method involves training some noise models in addition to the phoneme/word models [62]. This method, however, only works well when the noise is present in periods between speech.

Speech and noise decomposition (SND) [140] requires a three-dimensional Viterbi decoder. Each HMM state is decomposed into a number of states for noise. This approach is computationally expensive, although in theory it can handle non-stationary noises quite well.

Using a separate model for the noise, parallel model combination (PMC) [41] transforms the clean linguistic HMMs to noisy linguistic HMMs, assuming that the noise is additive. It does this by converting the model statistics from the cepstral domain to the linear spectral domain. In this domain, the means and variances of the clean speech models and the noise models can be added. The resulting mean and variance statistics are converted back to the cepstral domain in order to obtain the noisy speech model parameters. This results in a modification to both the mean and variances of each state. PMC is essentially an extension to another method called state-dependent Weiner filters [141], in which only the means of the HMMs are modified.

The HMM model parameters can also be altered by a MAP adaptation method. The model parameters are adapted by maximising the posterior probability of the model parameters, given some adaptation data. This results in a weighted sum of prior parameters (determined during initial training) and a maximum-likelihood estimate computed using the adaptation data [43]. Note that adaptation only occurs for those models represented in the training data.

Maximum likelihood linear regression (MLLR) [73] is another approach that utilises

some adaptation data. This method determines a set of linear transformations that can be applied to the mean and variance parameters of the HMMs. Unlike MAP, model parameters are altered even if they are not represented in the training data. If only a small amount of adaptation data is present, one set of transforms is applied to all HMMs. As more adaptation data becomes available, transforms for subsets of HMMs and/or Gaussian components can be applied.

Chapter 3

Short-time Phase Spectrum

In this chapter, we formally introduce the short-time Fourier transform and discuss how it is used to analyse, synthesise and modify a speech signal. We introduce two common representations derived from the short-time phase spectrum, briefly describing a number of speech applications in which these representations have successfully been used.

3.1 Short-time Fourier Transform

The short-time Fourier transform (STFT) [4, 5, 25, 34, 47, 111–115, 120, 123, 126] is the result of applying the Fourier transform at different points in time on finite length (i.e., short time) sections of a signal. This algorithm is fundamental to signal analysis because it introduces a time dependency, which the Fourier transform of the whole signal does not have. The separate Fourier transforms can be analysed for ASR feature extraction, or they can be individually processed and recombined to form a new processed signal (as we do in Chapter 4).

The continuous STFT of a signal, $x(t)$, is given by:

$$X(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(t-\tau)e^{-j\omega\tau}d\tau, \tag{3.1}$$

where $w(t)$ is the analysis window. In speech processing, the Hamming window function is typically used and its duration is normally 20–40 ms.

We can decompose $X(t,\omega)$ as follows:

$$X(t,\omega) = |X(t,\omega)|e^{j\psi(t,\omega)}, \tag{3.2}$$

where $|X(t,\omega)|$ is the short-time magnitude spectrum and $\psi(t,\omega) = \angle X(t,\omega)$ is the short-time phase spectrum. The signal, $x(t)$, is completely characterized by its short-time magnitude and phase spectra[1].

In order to implement the STFT, it must be discrete in both time and frequency. The discrete-time STFT is:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j\omega m}, \tag{3.3}$$

where n is the discrete time variable. The discrete STFT (which is discrete in both time and frequency) is given by:

$$X(n,k) = X(n,\omega)|_{\omega=2\pi k/N}, \tag{3.4}$$

where N is the number of frequency samples (or the length of the DFT) and $0 \leq k < N$. We can rewrite this as:

$$X(n,k) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j2\pi km/N}. \tag{3.5}$$

Usually, the time variation is decimated by a factor of L, such that we obtain $X(nL,k)$. L is changed depending on the time resolution required. If each speech segment is M points long, then the limits of the summation can be replaced to sum from $-\frac{M}{2}+1$ to $\frac{M}{2}$ (assuming M is even, as it usually is).

[1]This statement is always true for the continuous STFT. For the discrete STFT, this statement is only true under certain constraints (to be discussed).

3.2 Synthesis from the STFT

In Chapter 4 we use an STFT framework to reconstruct speech from its magnitude spectra and phase spectra[2]. Here, we introduce the theory behind reconstruction from the STFT. In particular, we discuss two classical methods that have been widely used for short-time synthesis; the filter-bank summation method and the overlap-add method.

3.2.1 Filter-bank Summation Method

The STFT can be viewed as the output of a set of filters, where the analysis window, $w(n)$, determines the filter impulse response[3]. To demonstrate this, we fix the value of ω at $\omega_k = 2\pi k/N$ and rewrite the expression for the discrete STFT as:

$$X(n,k) = \sum_{m=-\infty}^{\infty} [x(m)e^{-j\omega_k m}]w(n-m), \tag{3.6}$$

$$= [x(n)e^{-j\omega_k n}] * w(n), \tag{3.7}$$

where $*$ represents the convolution operation. It can also be written as:

$$X(n,k) = [x(n) * w(n)e^{j\omega_k n}]e^{-j\omega_k n}. \tag{3.8}$$

In Eq. 3.7, $x(n)$ is modulated by $e^{-j\omega_k n}$, then a baseband filter, $w(n)$, is applied (see Fig. 3.1(a)). In Eq. 3.8, $x(n)$ is filtered with a bandpass filter, $w(n)e^{j\omega_k n}$. The filtered signal is then modulated (to baseband) by $e^{-j\omega_k n}$ (see Fig. 3.1(b)). In both cases, the outcome is the same.

To construct a signal, $y(n)$, which ideally should be equal to the original signal, $x(n)$, we modulate each baseband signal, $X(n,k)$, with a complex exponential, $e^{j\omega_k n}$, and these outputs are summed at each time instant to obtain the original time sample.

[2]Recall that the modifier 'short-time' is implied when mentioning the magnitude spectrum and the phase spectrum.

[3]Throughout the discussions in this chapter we will alternate between the STFT forms (i.e., continuous, discrete-time and discrete versions). The specific forms we use in each section are deemed the most appropriate to convey the concepts.

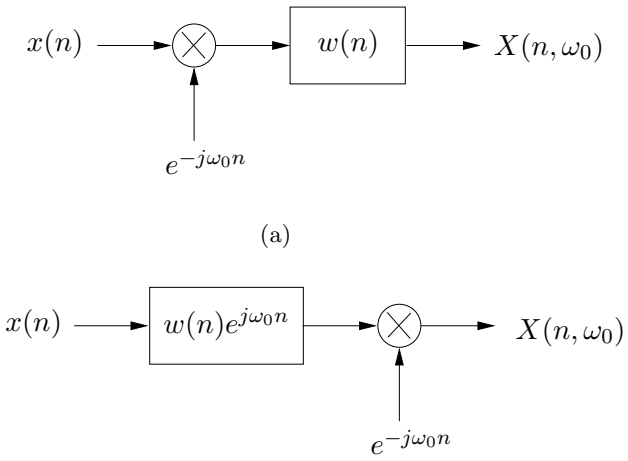


Fig. 3.1: Two filtering views of the STFT analysis. (a) Complex exponential modulation followed by low pass filtering, or (b) bandpass filtering followed by a complex exponential modulation.

That is,

$$y(n) = \frac{1}{Nw(0)} \sum_{k=0}^{N-1} X(n,k) e^{j2\pi nk/N}, \tag{3.9}$$

where $w(0)$ is the centre value of the analysis window (i.e., the weighting applied during the analysis at time n). Eq. 3.9 is referred to as the filter-bank summation (FBS) method. Substituting the definition for the discrete STFT into this equation, we obtain:

$$\begin{aligned} y(n) &= \frac{1}{Nw(0)} \sum_{k=0}^{N-1} \sum_{m=-\infty}^{\infty} x(m) w(n-m) e^{-j2\pi mk/N} e^{j2\pi nk/N} && (3.10) \\ &= \frac{1}{Nw(0)} \sum_{m=-\infty}^{\infty} x(m) \sum_{k=0}^{N-1} w(n-m) e^{j2\pi (n-m)k/N} && (3.11) \\ &= \frac{1}{Nw(0)} x(n) * \sum_{k=0}^{N-1} w(n) e^{j2\pi nk/N}, && (3.12) \end{aligned}$$

where the second term is the composite filter response. Eq. 3.12 can be rewritten as:

$$y(n) = \frac{1}{Nw(0)} x(n) * w(n) N \sum_{r=-\infty}^{\infty} \delta(n - rN). \tag{3.13}$$

Therefore, for $y(n) = x(n)$, the following constraint must be met:

$$w(n)N \sum_{r=-\infty}^{\infty} \delta(n - rN) = Nw(0)\delta(n). \tag{3.14}$$

This equation is referred to as the FBS constraint. It requires that the frequency sampling factor, N, be at least as large as the segment window size, M (which is the length of the window, $w(n)$). Alternatively, this constraint can be expressed in the frequency domain:

$$\sum_{k=0}^{N-1} W(\omega - 2\pi k/N) = Nw(0), \tag{3.15}$$

which ensures that the frequency responses of the filters sum to a constant across the entire frequency band.

To deal with temporal decimation, temporal interpolation filtering must be performed on the discrete STFT to restore the decimation factor to unity before modulation and addition is performed. Note that this can only be done if the decimation factor, L, is no larger than the window length (such that no samples are missed during the analysis).

3.2.2 Overlap-add Method

To construct a signal, $y(n)$, which ideally should be the same as the original signal, $x(n)$, the overlap-add (OLA) method requires that the inverse DFT be taken for each segment in the discrete STFT. Each of the short-time sections are then overlapped and added:

$$y(n) = \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} [\frac{1}{N} \sum_{k=0}^{N-1} X(p,k) e^{j2\pi kn/N}], \tag{3.16}$$

where

$$W(0) = \sum_{n=-\infty}^{\infty} w(n). \tag{3.17}$$

We can rewrite Eq. 3.16 as:

$$y(n) \quad = \quad \frac{1}{W(0)} \sum_{p=-\infty}^{\infty} x(n)w(p-n) \tag{3.18}$$

$$= \frac{1}{W(0)} x(n) \sum_{p=-\infty}^{\infty} w(p-n). \tag{3.19}$$

Therefore, for $y(n) = x(n)$, the following constraint must be met:

$$\sum_{p=-\infty}^{\infty} w(p-n) = W(0). \tag{3.20}$$

Further, if the STFT is decimated by factor L, then,

$$\sum_{p=-\infty}^{\infty} w(pL-n) = \frac{W(0)}{L}. \tag{3.21}$$

This is the OLA constraint. This constraint requires that the sum of the analysis windows (which are obtained by sliding $w(n)$ by L time samples) add up to the same value at each discrete point in time. This results in the elimination of the analysis window from the synthesised sequence. Thus, $x(n)$ can be reconstructed by:

$$x(n) = \frac{L}{W(0)} \sum_{p=-\infty}^{\infty} [\frac{1}{N} \sum_{k=0}^{N-1} X(pL,k) e^{j2\pi kn/N}]. \tag{3.22}$$

This process is graphically depicted in Fig. 3.2.

Recall that when viewing $X(n,k)$ as the filter output for a fixed frequency (Fig. 3.1), it has all the properties of a filtered sequence. Therefore, its bandwidth is smaller or equal to the bandwidth of the analysis filter, $w(n)$. Thus, the decimation factor, L, can not be arbitrarily large, otherwise we may undersample $X(n,k)$ across n. As we will show, the cutoff frequency of the analysis filter, ω_c, must be less than or equal to π/L. This constraint on L also ensures that the OLA constraint of Eq. 3.21 is satisfied. This is a direct result of the sampling theorem and is explained as follows. Given the maximum frequency of the analysis filter, ω_c, we need to sample $X(n,k)$ across n at a frequency of at least $2\omega_c$. This equates to a decimation factor of no more than:

$$L = \frac{1}{2f_c} = \frac{\pi}{\omega_c}, \tag{3.23}$$

since $w_c = 2\pi f_c$.

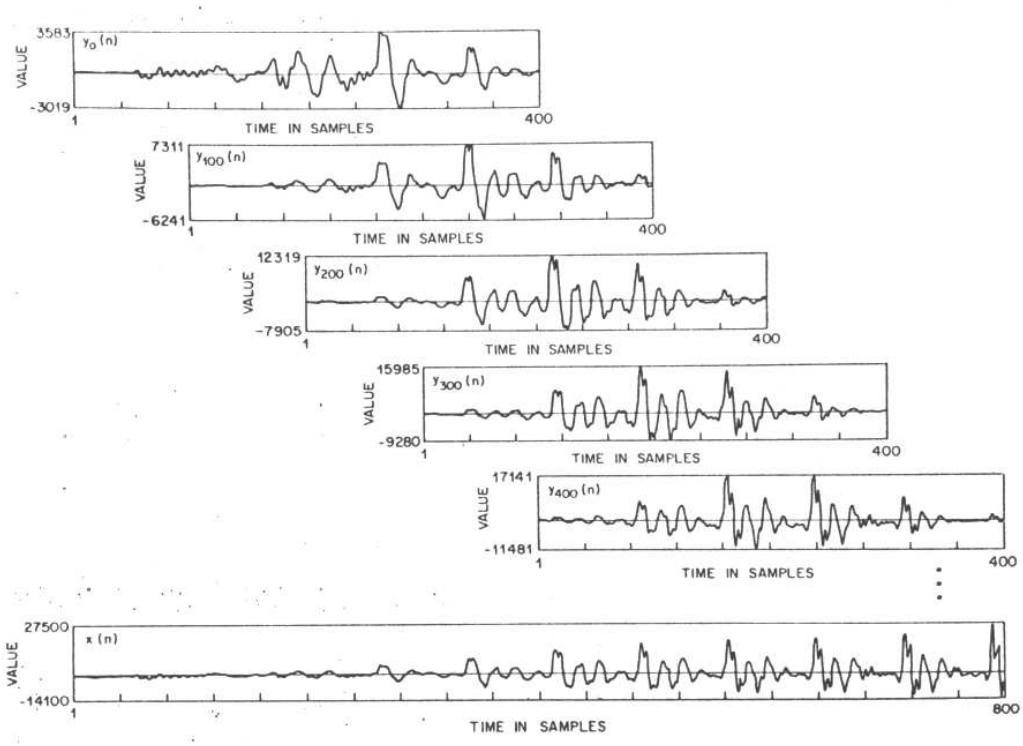


Fig. 3.2: Graphical interpretation of the OLA synthesis method. Each of the short-time segments are weighted with a Hamming window. The resulting summation is shown in the bottom panel (after [4]).

Consider the case for a Hamming analysis window of duration T_w (in seconds),

$$w(t) = 0.54 + 0.46\cos(2\pi t/T_w), \tag{3.24}$$

where $-T_w/2 \leq t \leq T_w/2$. Using a 42 dB criterion, the highest frequency in the signal is $f_c = 2/T_w$ [4]. This means we need to sample at a frequency of at least $4/T_w$. Therefore, the largest decimation factor for a Hamming analysis window, L_{ham}, is:

$$L_{ham} = \frac{1}{2f_c} = \frac{T_w}{4}. \tag{3.25}$$

That is, the frame shift between adjacent analysis frames should be no more than 1/4 of the frame length when a Hamming analysis window is used. By doing this, aliasing is avoided upon reconstruction.

For a large decimation rate, the OLA method is significantly more efficient than the

FBS method because the OLA method can use the decimated STFT directly (given that the above constraints are satisfied).

3.3 Synthesis from a Modified STFT

There are times when the need to reconstruct a signal from a modified STFT arises. The modifications to the STFT may be unintentional, such as in the case of quantisation errors, or they may be intentional, such as in the case of spectral subtraction for speech enhancement (see Section 2.8.2.1). However, an arbitrary change to an STFT does not necessarily result in a valid STFT. The definition of the STFT imposes a structure on time and frequency variations. Due to the overlap of short-time segments, adjacent segments cannot have arbitrary variations. If the phase spectra or magnitude spectra are modified, the STFT is only valid if the adjacent reconstructed sections are consistent in their region of overlap.

The OLA and FBS methods of reconstruction discussed in the previous section assume a valid STFT. Although there is no theoretical justification for using these methods to reconstruct a signal from a modified STFT, they are usually applied in a brute force manner. In most cases, these reconstruction methods provide reasonable results [74].

Rather than heuristically applying the OLA or FBS methods, one can use least-squares signal estimation from the modified STFT [47]. In this method, a signal is estimated which has an STFT which is closest in a least-squares sense to the modified STFT. That is, we want to minimise the mean squared error (MSE) between the modified STFT and the resulting STFT of the reconstructed signal.

Consider the following distance metric between the modified STFT, $X_{Mod}(pL, \omega)$, and the STFT of the reconstructed signal, $X_{Rec}(pL, \omega)$:

$$D[X_{Mod}(pL, \omega), X_{Rec}(pL, \omega)] = \sum_{p=-\infty}^{\infty} \frac{1}{2\pi} \int_{\omega=-\pi}^{\pi} |X_{Mod}(pL, \omega) - X_{Rec}(pL, \omega)|^2 d\omega. \tag{3.26}$$

Due to Parseval's theorem, we can rewrite the above equation as:

$$D[X_{Mod}(pL,\omega), X_{Rec}(pL,\omega)] = \sum_{p=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} [x_{Mod}(pL,n) - x_{Rec}(pL,n)]^2, \quad (3.27)$$

and minimising with respect to $x_R(n)$:

$$x_{Rec}(n) = \frac{\sum_{p=-\infty}^{\infty} w(pL-n)x_{Mod}(pL,n)}{\sum_{p=-\infty}^{\infty} w^2(pL-n)}. \quad (3.28)$$

Therefore, for the MSE solution, the modified frames must be weighted with the analysis window before overlap and addition. Also, the resulting signal must be normalised by the overlap and addition of $w^2(pL-n)$ rather than $w(pL-n)$ (see Eq. 3.21 and Eq. 3.22). Note that this method reduces to OLA if the analysis window is rectangular. Thus, applying the OLA method in the case of a rectangular analysis window results in a least-squares estimate of the signal [74].

3.4 Difficulties with Processing of the Phase Spectrum

There are a number of signal processing difficulties with using the phase spectrum directly for ASR. We discuss two of the most critical problems here; phase-unwrapping and time dependency.

3.4.1 Phase-Unwrapping

In an effort to make more sense of the phase spectrum, it is often unwrapped. However, the problem with this is that unwrapping is an heuristic process.

The 'principal' phase spectrum of $X(t,\omega)$ is denoted by $\text{ARG}[X(t,\omega)]$. That is:

$$-\pi < \text{ARG}[X(t,\omega)] \leq \pi. \quad (3.29)$$

Note that the principal phase spectrum values fall in the range $+/-\pi$. It can be

obtained directly by using the arctangent function (4-quadrant version):

$$\mathrm{ARG}[X(t,\omega)] = \arctan(\frac{X_I(t,\omega)}{X_R(t,\omega)}), \tag{3.30}$$

where the subscripts R and I denote the real and imaginary parts respectively[4]. As the principal phase spectrum values exceed the $+/-\pi$ limits, the values may change abruptly from negative to positive or vice versa. The principal values are said to be 'wrapped' around these limits.

Phase-unwrapping algorithms seek to determine a 'continuous' phase spectrum, denoted by $\arg[X(t,\omega)]$. Estimates of the continuous phase spectrum are often referred to as the 'unwrapped' phase spectrum. The fundamental difficulty in finding the continuous phase spectrum is that any multiple of 2π can be added to the principal phase spectrum without changing the values of the complex number $X(t,\omega)$. Thus, there are an infinite number of ways to unwrap the principal phase spectrum. There is, however, only one correct continuous phase spectrum. An example of where the continuous phase spectrum is required is in the definition for the complex cepstrum [99], which is fundamental to homomorphic signal processing. The complex cepstrum of $X(t,\omega)$, denoted by $\hat{X}(t,\omega)$, is defined as:

$$\begin{aligned} \hat{X}(t,\omega) &= \log X(t,\omega), & (3.31)\\ &= \log|X(t,\omega)| + j\arg[X(t,\omega)]. & (3.32) \end{aligned}$$

It can be seen that the complex cepstrum is only defined for the continuous phase spectrum. A different phase spectrum (even with some values only modified by the addition of 2π) would result in a different complex cepstrum.

The derivative of the continuous phase spectrum is well defined [101]:

$$\frac{d\arg[X(t,\omega)]}{d\omega} = \frac{X_R(t,\omega)X_I'(t,\omega) - X_I(t,\omega)X_R'(t,\omega)}{|X(t,\omega)|^2}, \tag{3.33}$$

[4]We employ the same nomenclature as used in [101]. Specifically, $\angle X(t,\omega)$ denotes the ambiguous phase spectrum, $\mathrm{ARG}[X(t,\omega)]$ denotes the principal phase spectrum, and $\arg[X(t,\omega)]$ denotes the continuous phase spectrum.

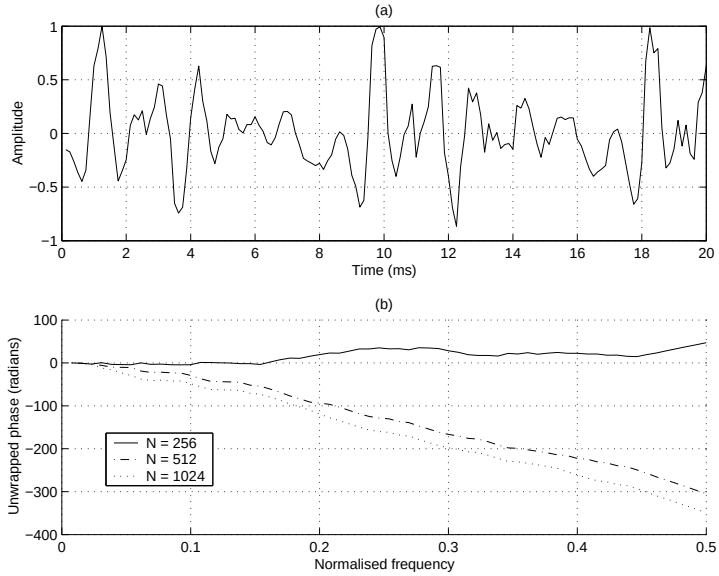


Fig. 3.3: The 20 ms speech segment in (a), which is 160 samples long (sampled at 8 kHz), is analysed with a DFT of length $N = 256, 512,$ and 1024. The respective unwrapped phase spectra are shown in (b). The unwrapped phase spectrum is calculated using the Matlab function unwrap(). *This example demonstrates that the unwrapped phase spectrum is dependent on the DFT bin spacing.*

where the prime denotes $d/d\omega$. $\arg[X(t,\omega)]$ can thus be defined as:

$$\arg[X(t,\omega)] = \int_0^{\omega} \arg\prime[X(t,\eta)]d\eta, \tag{3.34}$$

with initial conditions given by:

$$\arg[X(t,0)] = 0. \tag{3.35}$$

Although the continuous phase spectrum is precisely defined, it can not be exactly implemented on a computer. Often, the calculated discrete phase spectrum values are not equal to the corresponding continuous phase spectrum values. There are a number of methods used to estimate the continuous phase spectrum:

- One method involves the numerical integration of the principal phase spectrum derivative. This method is heavily dependent on the size of the integration step, $2\pi/N$ (where N is the DFT length).

- Another more popular method determines a phase spectrum which constrains the absolute differences between adjacent principal phase spectrum values to be less than a pre-defined tolerance. This tolerance is normally chosen to be π. When the absolute difference between the phase spectral values of adjacent bins is greater than π, the values are adjusted by adding or subtracting a multiple of 2π such that the resultant values differ by no more than π (see the code used in the Matlab function *unwrap()*). Again, as demonstrated in Fig. 3.3, the resulting unwrapped phase spectrum is dependent on the DFT bin spacing.

- Yet another technique, Tribolet's method [138], adapts the integration step size until a consistent estimation of the unwrapped phase spectrum is found.

There are several other phase-unwrapping methods [44,95]. In general, phase-unwrapping is an heuristic process.

3.4.2 Time Dependency

The phase spectrum is highly dependent on the exact positioning of the short-time analysis window. No matter how small the time shift of the analysis window, the phase spectrum values will dramatically change. This is demonstrated in Fig. 3.4. This is not desirable for an ASR feature representation.

In ASR, we require a consistent representation for similar instances of speech, independent of their position in time. The magnitude spectrum representation meets this requirement, thus it has proven to be popular for ASR (compare Fig. 3.4(c) to Fig. 3.4(d)). If the phase spectrum is ever to be used for ASR, it will need to be transformed into a more robust representation.

3.5 Representations Derived from the Short-time Phase Spectrum

The short-time phase spectrum has two independent variables: frequency and time. Thus, while there may be many ways to represent the information present in the phase

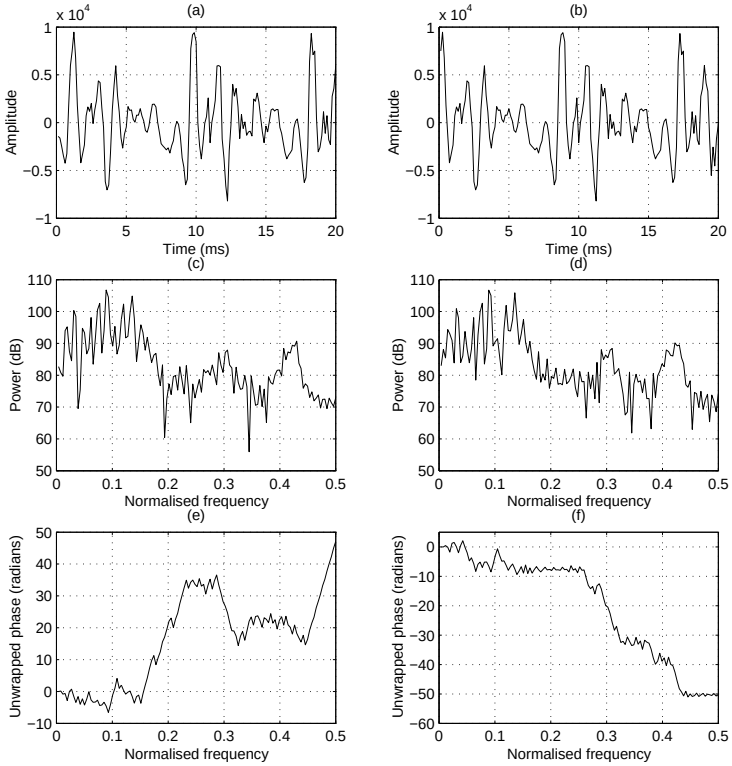


Fig. 3.4: The two 20 ms segments of speech in (a) and (b) are separated in time by only 1 ms. The magnitude spectrum for the first segment is shown in (c) and the magnitude spectrum of the second segment is shown in (d). They are very similar. The respective unwrapped phase spectra (calculated using the Matlab function unwrap()*) are shown in (e) and (f). This example demonstrates the consistency of the magnitude spectrum representation and the time dependency of the phase spectrum.*

spectrum, two representations that first come to mind are those that can be obtained either by calculating its frequency-derivative or its time-derivative.

3.5.1 Frequency-derivative of the Phase Spectrum

The group delay function (GDF) is a measure of the non-linearity of the phase spectrum [101]. It is defined as the negative derivative of the continuous phase spectrum[5]:

$$\tau(t,\omega) = -\frac{d}{d\omega}\arg[X(t,\omega)] \quad = \quad -\frac{X_R(t,\omega)X'_I(t,\omega) - X_I(t,\omega)X'_R(t,\omega)}{|X(t,\omega)|^2}, \quad (3.36)$$

which is the negative of Eq. 3.33.

[5]Note that this expression is normally written without the time dependency. Time is explicitly specified here, since we are discussing the short-time phase spectrum.

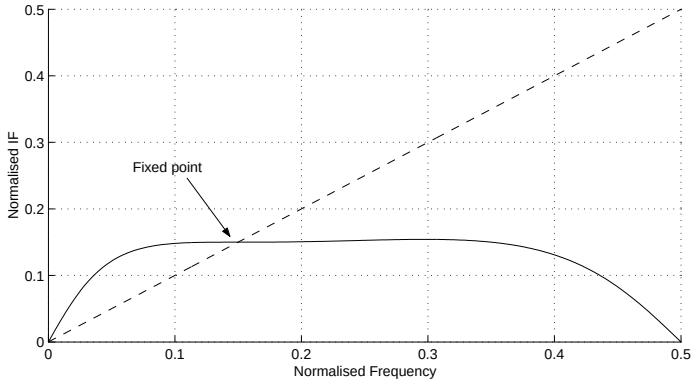


Fig. 3.5: The IFD of a segment of a sinusoid. The best estimate of the sinusoidal frequency is where the IFD crosses the diagonal line (i.e., the fixed-point).

In practice, a discrete version of Eq. 3.36 is used to estimate the GDF. The difficulties with using this GDF estimation method for ASR are discussed in Section 7.1.

The GDF is further discussed in Chapter 6, where it is considered as a representation for ASR feature extraction.

3.5.2 Time-derivative of the Phase Spectrum

A useful interpretation of the short-time phase spectrum is its first-order time-derivative, called the instantaneous frequency distribution (IFD):

$$IF(t,\omega) = \frac{1}{2\pi}\frac{d\psi(t,\omega)}{dt}. \tag{3.37}$$

In practice, the differentiation in Eq. 3.37 is performed by the simple difference method; the difference is taken between the values of two (short-time) phase spectra separated in time by one time sample[6]. The division by 2π provides a normalised measure of the instantaneous frequency (IF). The normalised IF can be multiplied by the sampling frequency (in Hertz) to obtain the actual IF (in Hertz).

The IFD is discussed further in the pursuing sections.

[6]The differentiation can be calculated by multiplying the DFT of the delayed frame by the complex conjugate of the DFT of the first frame. The phase spectrum of the resulting 'cross-spectrum' vector is in fact the difference between the two original phase spectra.

3.6 Some Uses of the Short-time Phase Spectrum

Although the phase spectrum has yet to be proven useful for ASR[7], it has successfully been used for many other tasks, such as F0 extraction [1, 22, 89], determination of significant instants [87, 133], formant extraction [31, 38, 86, 116], and iterative signal reconstruction [47, 50, 83, 90, 100, 119, 137, 139, 147, 150]. In the following subsections we discuss how the GDF and IFD are used in some of these applications. Our intention is not to provide implementation details; the reader is referred to the cited references for this. A separate chapter is devoted to a discussion on iterative signal reconstruction (Chapter 5).

3.6.1 Determination of Fundamental Frequency

There are many ways to estimate the fundamental frequency (F0) of a speech signal. F0 is usually extracted from the autocorrelation function, the cepstrum, zero-crossing rates, or even the magnitude spectrum [60, 122]. The phase spectrum, through its IFD representation, is also used to extract estimates of F0.

In general, IF values for each DFT bin give an estimate of the dominant frequency within the vicinity of each bin center frequency. In Fig. 3.5 we calculate the IFD (i.e., the IF for each DFT bin from $\omega = 0$ to π) of a segment of a sinusoid. The IF values for most of the bins are almost equal to the frequency of the sinusoid. As can be observed, the best estimate of the sinusoidal frequency is where the IFD crosses the diagonal line. This cross-over point is often referred to as a fixed-point.

However, the speech signal is much more complicated than a simple sinusoid. For a typical ASR analysis duration of 20–40 ms, the IF of each DFT bin locks on to the frequency of a speech harmonic. For example, consider the signal in Fig. 3.6(a); it is a 25 ms segment from an instance of the voiced phoneme 'iy' (as in the English word 'me'). The segment encapsulates just over two pitch periods. The power spectrum is shown in Fig. 3.6(b). The IFD in Fig. 3.6(c) is obtained by plotting the IF for each bin against their respective bin center frequencies. We zoom in on this IFD in Fig. 3.6(d). The IFD

[7] There have been some attempts at using the phase spectrum as a representation for ASR feature extraction [10, 29, 53, 54, 88, 108, 117, 142].

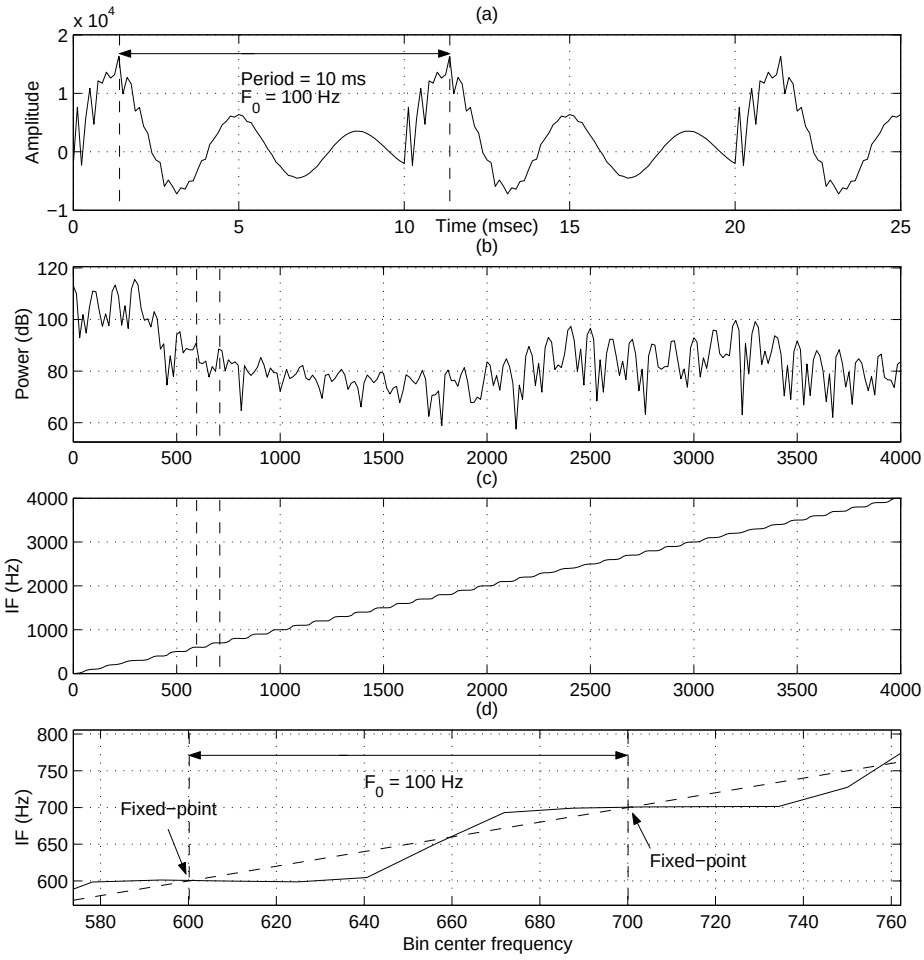


Fig. 3.6: Examples to aid the discussion of the IFD in Section 3.6.1. (a) A 25 ms segment from an instance of the voiced phoneme 'iy' (as in the English word 'me'), (b) the power spectrum calculated over the 25 ms segment, (c) the IFD calculated over the 25 ms segment, and (d) inset of IFD showing some fixed points.

resembles a staircase. Each small step (flat region) corresponds to a speech harmonic (see this by comparing Fig. 3.6(b) and Fig. 3.6(c)). The best estimates of the harmonic frequencies occur at the fixed-points. The averaging of the separation between these points provides a robust method of determining FO information [1, 22, 89]. In fact, it is well known that the IFD carries information about the vocal-tract excitation [34, 114].

It is interesting to note that this locking of bin frequencies is similar to the 'phase-locking' phenomenon observed in the auditory system (Section 2.3.1.2).

3.6.2 Formant Extraction

Recall that the DFT can be interpreted as a bank of uniformly-spaced band-pass filters. These band-pass filters have small bandwidths. When the spectrum exhibits F0 harmonics, the narrow filters are locked to these F0 harmonics and are effectively blind to the formant resonances. However, the IF values lock on to the formant resonances when the F0 harmonics are removed from the spectrum through some smoothing operation (such a cepstral smoothing and all-pole modeling). In other words, in the absence of source-induced harmonics, the IFD will resolve formant frequencies.

If we take a small segment of speech, starting from the beginning of a glottal pulse and finishing approximately half way through the pulse, its Fourier transform will have minimal contamination from source information. The power spectrum calculated from the 6 ms signal in Fig. 3.7(a) is shown in Fig. 3.7(b). The IFD for this shorter segment is shown in Fig. 3.7(c). This is called the wide-band IFD. The spurious values in the wide-band IFD are a direct result of the ambiguous nature of the phase spectrum values (as discussed in Section 3.4.1). Ignoring these spurious values (which can be smoothed out if desired), there are three distinct flat regions, corresponding to formant frequencies. The formant frequency values provided by the wide-band IFD and the power spectrum for this 6 ms segment are in agreement. The wide-band representation, however, requires the use of a small analysis window duration (of about half the pitch period) positioned at the start of the glottal pulse, so that source information is minimised. If features are to be extracted from such a representation, a pitch-synchronous analysis is required. Such a representation can not be used in the standard ASR framework, which obtains features from a pitch-asynchronous analysis. Ideally, a representation for ASR feature extraction should be independent of the F0.

Fig. 3.8 shows a histogram of the wide-band IFD from Fig. 3.7(c). Friedman [38] shifts a 4 ms Hanning window over a speech utterance, then plots a topographical view of IFD histograms for each segment in order to create a spectrogram-like plot. When the 4 ms window is at the start of a pitch period, this IF-spectrogram reveals the formant frequencies. When the 4 ms window is between the points of excitation, the IF-

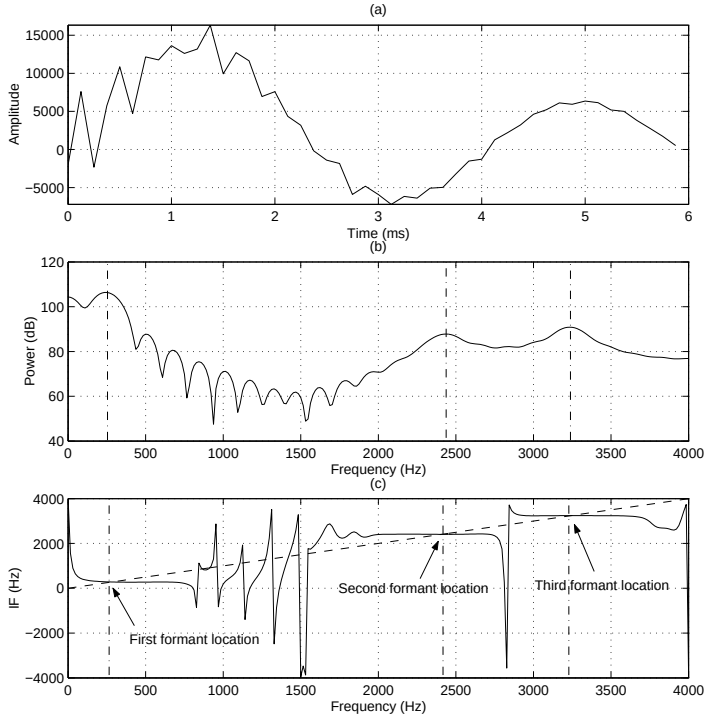


Fig. 3.7: Example to aid the discussion of the IFD in Section 3.6.2. (a) The first 6 ms of the signal in Fig. 3.6, (b) the wide-band power spectrum, and (c) the wide-band IFD. The vertical dashed lines indicate the estimated formant locations.

spectrogram provides spurious values. In order to remove the spurious values, Friedman applies two-dimensional linear smoothing.

Nelson [91,92] has proposed an alternative spectral representation which is useful for accurate measurement of vocal tract resonances. By plotting the spectral magnitude or spectral power against the IF for each DFT bin (rather than the bin center frequencies), he obtains a 'phase re-parameterised (PR) spectrum'. An example of the PR spectrum is shown in Fig. 3.9. As with the IFD, the formants are only resolved when using a short (wide-band) analysis window.

Duncan et al. [31] have suggested a method of formant estimation through use of the GDF. As mentioned above, the wide-band IFD technique of formant estimation relies on the small size and the precise location of the analysis window. This is explained by Duncan et al. as follows: speech is assumed to be produced by a minimum-phase system[8]. However, the arbitrary placement (usually every 10–20 ms) and size (usually

[8]See [101] for an explanation of minimum-phase and maximum-phase systems.

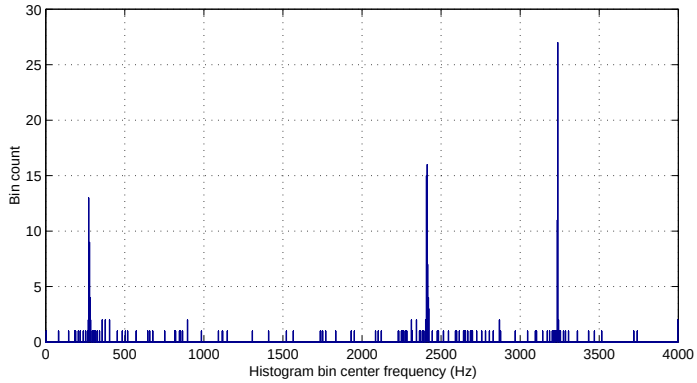


Fig. 3.8: Histogram of the wide-band IFD shown in Fig.3.7(c). This is an alternative view of the IFD which clearly shows the three formant locations.

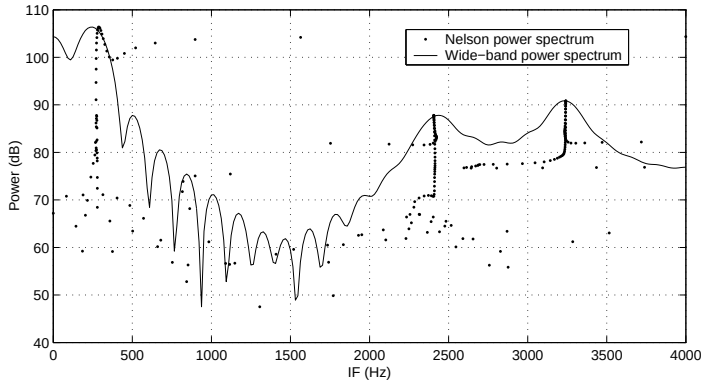


Fig. 3.9: The Nelson power spectrum (or phase re-parameterised spectrum) for the 6 ms speech segment shown in Fig. 3.7. The Nelson power spectrum is determined by plotting power spectrum values against the IF values for each DFT bin.

20–40 ms) of the analysis window generally results in a mixed-phase segment. There are two reasons for this; firstly, the zero-time reference point of the observation window does not align with start of the minimum-phase impulse response, and secondly, the length of the analysis window encompasses more than one point of excitation. Since the observed response is mixed-phase, there are zeros both inside and outside the z-plane unit circle. The GDF of this mixed-phase signal exhibits spurious peaks which render it useless for formant extraction (the reasons for this are discussed in detail in Section 7.1). In order to work around this problem, Duncan et al. derive a minimum-phase signal from the mixed-phase speech segment. This is done by taking the inverse Fourier transform of the co-phase (zero-phase) magnitude spectrum. The significant instant in this reconstructed signal is at the beginning of the segment. A smaller (wide-band)

window is then applied to only the beginning of this reconstructed segment. The GDF of this new signal has significant peaks that correspond to formant frequencies.

Consider the 25 ms segment of speech provided in Fig. 3.10(a). The mixed-phase nature of the signal results in a GDF which is useless (Fig. 3.10(e)). The zero-phase signal is shown in Fig. 3.10(b). The significant instant for this zero-phase signal is at the start of the window. The GDF in Fig. 3.10(f), calculated from a smaller analysis window positioned at the start of this zero-phase signal, is much more informative than that calculated by the mixed-phase speech segment. The three largest peaks of the GDF coincide with the position of the formants in the wide-band power spectrum (Fig. 3.10(d)). For interest, we demonstrate that a wide-band IFD (Fig. 3.10(h)) can also be calculated from the zero-phase signal (again, using a small analysis window at the start of the zero-phase signal and another shifted by one sample).

Rather than obtain the GDF from a minimum-phase signal (as Duncan et al. did [31]), Murthy et al. [86] smooth the phase spectrum of the mixed-phase speech segment before computing its GDF. Additionally, the GDF calculated from the smoothed phase spectrum is cepstrally smoothed. The smoothing is done in an effort to reduce the large fluctuations in the phase spectrum introduced by the zeros that are close to the z-plane unit circle. The position of the formants are clearly visible in the smoothed GDF. In later work [85, 145], Murthy et al. have modified the GDF definition in order to make it less volatile to the effects of zeros. This modified GDF (MGDF) is discussed in detail in Section 7.2.

3.6.3 Determining the Instants of Major Excitation

Yegnanarayana et al. [133, 148], have used the GDF to determine the instants at which glottal closures occur. These instants are called significant instants. It is useful to find the locations of the significant instants so that wideband analysis windows can be accurately placed for formant estimation.

The basic idea behind the algorithm is conveyed through the following example. Consider a unit impulse at time τ. Its Fourier transform is $e^{-j\omega\tau}$. The phase function is $\phi(\omega) = -\omega\tau$ and the group delay function is $-\phi\prime(\omega) = \tau$. The average group delay

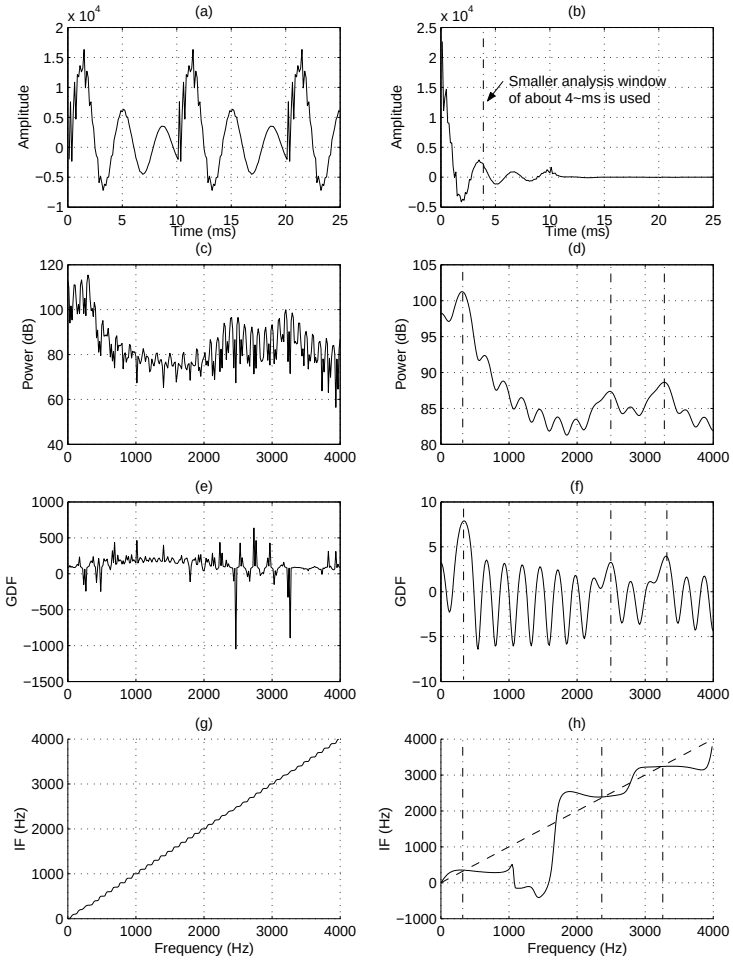


Fig. 3.10: The left column shows the (a) 25 ms speech segment and its associated (b) power spectrum, (c) GDF, and (d) IFD. The right column shows the (a) zero-phase equivalent of the speech segment and its associated (b) power spectrum, (c) GDF, and (d) IFD. The power spectrum, GDF and IFD for the zero-phase speech segment are calculated by using a 4 ms analysis window positioned at the beginning of the segment.

(or phase slope), τ, is equal to the delay of the unit impulse. The delay, and thus the phase slope, changes with respect to the analysis window position. Yegnanarayana et al. plot the average value of the GDF as a function of the analysis window position. They call this the phase-slope function. On a simple signal, such as a unit impulse train, the points at which the phase-slope function crosses zero are considered to be the time at which the unit impulses occur (in this case, the significant instants are when the unit impulses occur).

Although speech is much more complex than a unit impulse train, Yegnanarayana et al. have demonstrated that the phase-slope function of a speech signal can still provide

quite reasonable estimates of where the significant instants occur. When this method is applied to speech, it is best to apply it to the LP residual. The LP residual maintains all of the source information, but it reduces the truncation effects of windowing [148]. The robustness of this method is demonstrated in [87].

Chapter 4

Human Listening Experiments

In this chapter, the usefulness of the phase spectrum is explored in human speech perception[1]. We have a long-term goal of utilising phase spectra in an effort to improve ASR performance. It is common practice in ASR to discard the phase spectrum in favour of features that are derived purely from the magnitude spectrum. In the ASR framework, speech is processed frame-wise using a temporal window of duration 20–40 ms. If the phase spectrum is to be of any use for ASR applications, it should provide some information about speech intelligibility using small window durations (20–40 ms) in a human perception experiment.

A few studies have been reported in the literature which discuss whether the phase spectrum provides any information which can contribute to intelligibility for human speech recognition (HSR). Schroeder [127], and Oppenheim and Lim [100] performed some informal perception experiments, concluding that the phase spectrum is important for intelligibility when the window duration of the short-time Fourier transform (STFT) is large ($T_w > 1$ s), while it seems to convey negligible intelligibility at small window durations (20–40 ms). However, Cox and Robinson [24] have informally observed that important waveform features of speech are retained when speech is reconstructed from short-time phase information at a window duration of 25.6 ms.

Liu et al. [80] have recently investigated the intelligibility of phase spectra through a more formal human speech perception study. They recorded 6 stop-consonants from

[1]Example audio files are available at http://maxwell.me.gu.edu.au/spl/research/phase/project.htm

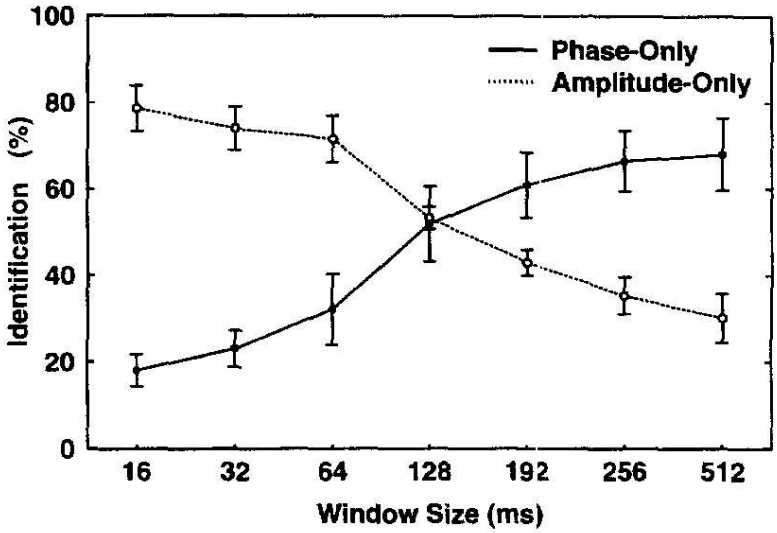


Fig. 4.1: Average identification performance and standard deviation as a function of window size for phase-only and magnitude-only stimuli, from the paper by Liu et al. (after [80]).

10 speakers in vowel-consonant-vowel context. Using these recordings, they created *magnitude-only* and *phase-only* stimuli. Magnitude-only stimuli were created by analysing the original recordings with a STFT, replacing each frame's phase spectra with random phase values, then reconstructing the speech signal using the overlap-add method. In the case of phase-only stimuli, the original phase of each frame was retained, while the magnitude of each frame was set to unity for all frequency components. The stimuli were created for various window lengths from 16 ms to 512 ms. These were played to subjects, whose task was to identify each as one of the 6 consonants. Their results (Fig. 4.1) show that intelligibility of magnitude-only stimuli decreases while the intelligibility of the phase-only stimuli increases as the window duration increases. For small window durations ($T_w < 128$ ms), magnitude-only stimuli are significantly more intelligible than phase-only stimuli (while the opposite is true for larger window lengths). This implies that for small window durations (which are of relevance for ASR applications), the magnitude spectrum contributes much more toward intelligibility than the phase spectrum.

Our initial intention was to reproduce Liu's results; in doing so, we have made a

number of modifications to Liu's analysis-modification-synthesis procedure. The modifications produce results which are different from Liu's results and more interesting from an ASR application's viewpoint. The first suggested modification is that of the analysis window type. Liu and his collaborators employed a Hamming window for construction of both the magnitude-only and phase-only stimuli. In our experiments, we find that the intelligibility of phase-only stimuli is improved significantly and becomes comparable to that of magnitude-only stimuli when a rectangular window is used. The second suggested modification is the choice of analysis frame shift; Liu et al. used a frame shift of $T_w/2$. As shown by Allen and Rabiner [4, 5], in order to avoid aliasing errors during reconstruction, the STFT sampling period (or frame shift) must be at most $T_w/4$ for a Hamming window (see Section 3.2.2). In our experiments, to be on the safer side, we use a frame shift of $T_w/8$. Our study also differs from Liu's study with respect to the number of consonants used (16 for this study compared to 6 for Liu et al). The design parameters are discussed in further detail later in this chapter. Our results indicate that even for small window durations ($T_w < 128$ ms), the phase spectrum can contribute to speech intelligibility as much as the magnitude spectrum if the analysis-modification-synthesis parameters are properly selected.

The outline of this chapter is as follows: In Section 4.1, we detail the analysis-modification-synthesis technique used to create the phase-only and magnitude-only stimuli. In Section 4.2, we describe a number of experiments which evaluate the importance of short-time phase spectra and short-time magnitude spectra in human speech perception:

- In Experiment 1 (Section 4.2.1), we demonstrate that intelligibility of phase-only stimuli is improved significantly when a rectangular analysis window is used, and it becomes comparable with that of magnitude-only stimuli even for small window durations.

- In Experiment 2 (Section 4.2.2), we construct magnitude-only and phase-only stimuli for window sizes ranging from 16 ms to 2048 ms, using both Liu's parameter settings and our parameter settings (discussed in Experiment 1) in order to

compare their intelligibility.

- In Experiment 3 (Section 4.2.3), we ascertain the contribution that each analysis-modification-synthesis parameter provides toward the intelligibility of signals reconstructed from phase spectra.

- In the aforementioned experiments, magnitude-only stimuli are created by randomising each frame's phase spectra. It is also possible to create magnitude-only stimuli by setting all phase values for each frame to zero. Thus, in Experiment 4 (Section 4.2.4), we address the issue of using random-phase or zero-phase and determine if a significant difference exists between magnitude-only stimuli constructed with one or the other.

- In Experiment 5 (Section 4.2.5), we explore the use of partial phase spectrum information, in the absence of all the magnitude spectrum information, for intelligible signal reconstruction. We create two types of stimuli; one in which the phase spectrum frequency-derivative (i.e., GDF) is preserved and another in which the phase spectrum time-derivative (i.e., IFD) is preserved. We do this to determine the contribution that each component of the phase spectrum provides toward intelligibility. If we obtain significant intelligibility from either component, then it would be wise to investigate the component's potential as a basis for an ASR representation. Conversely, if we obtain poor intelligibility, perhaps we should consider other phase spectrum representations.

- Ultimately, the speech community is searching for ASR features that are robust to noise. Hence, in Experiment 6 (Section 4.2.6) we attempt to quantify the intelligibility of stimuli reconstructed from the phase spectrum and the magnitude spectrum of noisy speech.

Publications resulting from this research: [7–9, 106, 107].

4.1 STFT Analysis-modification-synthesis Technique

The aim of the experiments in Section 4.2 is to determine the contribution that the phase and magnitude spectra provide toward speech intelligibility. Accordingly, stimuli are created either from phase or magnitude spectra[2]. In order to construct, for example, an utterance with only phase spectra, the signal, $x(n)$, is processed through a discrete-STFT analysis to obtain $X(n,k)$. The magnitude spectrum is made unity in the modified STFT $\hat{X}(n,k)$; that is[3],

$$X_{Mod}(n,k) \quad = \quad e^{j\psi(n,k)}. \tag{4.1}$$

The modified STFT (with unity magnitude spectra) is then used to synthesize a signal, $\hat{x}(n)$, using the OLA method[4]. The synthesized signal, $\hat{x}(n)$, contains all of the information about the short-time phase spectra contained in the original signal, $x(n)$, but will have no information about the short-time magnitude spectra. We refer to this procedure as the STFT *phase-only synthesis* and the utterances synthesized by this procedure as the *phase-only* utterances. Similarly, for generating *magnitude-only* utterances, we retain each frame's magnitude spectrum and randomise each frame's phase spectrum; that is, the modified STFT is computed as follows:

$$X_{Mod}(n,k) \quad = \quad |X(n,k)|e^{j\phi(n,k)}, \tag{4.2}$$

where ϕ is a random variable uniformly distributed between 0 and 2π. It may also seem plausible to set ϕ to zero for all values of n and k (i.e., time and frequency). In Experiment 4, reported later in Section 4.2, we test the intelligibility of magnitude-only stimuli constructed with zero-phase.

[2]Matlab code is provided in Appendix D.

[3]Recall that the phase spectrum, $\psi(n,k)$, can be wrapped or unwrapped. Either way, the value of the $X_{Mod}(n,k)$ will be the same.

[4]In the following experiments, we use Allen and Rabiner's OLA reconstruction method [4] (Section 3.2.2) rather than Griffin and Lim's LSE method [47] (Section 3.3). The methods are identical when using a rectangular window. We have performed some experiments with a Hamming window which indicate that there is no significant difference in intelligibility between stimuli constructed from either method.

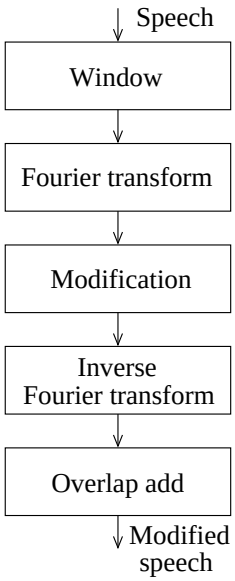


Fig. 4.2: Speech analysis-modification-synthesis system.

In the case of phase-only stimuli, the reader may wonder why we do not replace the magnitude spectra with random values. We have observed that doing so renders the reconstructed speech unintelligible. We propose that this occurs for the following reason: Consider the original signal. Where the magnitude spectrum values are low, the phase spectrum values are noisy. Where the original magnitude spectrum values are high, the phase spectrum values are information bearing. By modifying the signal such that the magnitude spectrum is unity, the noisy phase spectrum values are emphasised somewhat such that their weighting is now equal to the information bearing phase spectrum values. That is, we are emphasising noisy phase spectrum values while retaining the information bearing phase spectrum values. However, if we randomise the magnitude spectrum values, the information bearing phase spectrum values are randomly suppressed (which we do not want to do).

In the STFT-based speech analysis-modification-synthesis system (shown in Fig. 4.2), there are 4 design issues that must be addressed.

1. **Analysis window type:** This refers to the type of window function, $w(n)$, used for computing the STFT. A tapered window function (such as Hanning, Hamming or triangular) has been used in earlier studies [80]. Considering these studies

have found the phase spectrum to be unimportant at small window durations, a rectangular (non-tapered) window function is investigated in this study in addition to a Hamming window function.

2. **Analysis window duration:** Over the course of the experiments, we investigate 8 window durations (16, 32, 64, 128, 256, 1024, and 2048 ms).

3. **STFT sampling period (frame shift):** In order to avoid aliasing during reconstruction, the STFT must be adequately sampled across the time axis. The STFT sampling period is decided by the window function, $w(n)$, used in the analysis. For example, for a Hamming window, the sampling period should be at most $T_w/4$ [4] (see Section 3.2.2). To be on the safer side, we have used a sampling period of $T_w/8$. In discrete terms, if M is the number of samples in a frame, then the frame shift is $M/8$ samples. Although the rectangular window can be used with a larger sampling period, we use the same sampling period (i.e., $T_w/8$) to maintain consistency. We also refer to the STFT sampling period as the frame shift.

4. **Zero-padding:** For a windowed frame of length M (where M is a power of 2), the Fourier transform is computed using the fast Fourier transform (FFT) algorithm with a FFT size of $N = 2M$ points. This is equivalent to appending M zeros to the end of the M-length frame prior to performing the FFT. The resulting STFT is modified, then each frame is inverse Fourier transformed to get reconstructed frames of length N. Only the first M points of each frame are used in the OLA procedure, while the last M points are discarded. This is done in order to minimise aliasing effects. Zero-padding is used in the construction of all stimuli in these experiments, unless otherwise stated.

Table 4.1: Consonants used in all perception testing.

a-Consonant-a	As in
aba	bat
ada	deep
afa	five
aga	go
aka	kick
ama	mum
ana	noon
apa	pea
asa	so
ata	tea
ava	vice
aza	zebra
adha	then
asha	show
atha	thing
azha	measure

4.2 Human Listening Experiments

4.2.1 Experiment 1

In this experiment we compare the intelligibility of magnitude-only and phase-only stimuli using two window types: 1) a rectangular window, and 2) a Hamming window[5]. This comparison is done at a small window duration of 32 ms as well as a large window duration of 1024 ms.

4.2.1.1 Recordings

We record 16 commonly occurring consonants in Australian English in aCa context (Table 4.1) spoken in a carrier sentence "Hear aCa now". For example, for the consonant /d/, the recorded utterance is "Hear ada now". These 16 consonants in the carrier sentence are recorded for 4 speakers: 2 males and 2 females, providing a total of 64 utterances. The recordings are made in a silent room with a SONY ECM-MS907 microphone (90 degree position). The signals are sampled at 16 kHz with 16-bit precision.

[5]A triangular window function was also investigated. Results are similar to those provided by the Hamming window in all test conditions. Therefore, we do not report these results.

Table 4.2: Stimuli for Experiment 1 (with frame shift of $T_w/8$).

Type of stimuli	Retained Spectrum	Window Type	Window Duration (ms)
A1	Magnitude	Hamming	32
B1	Magnitude	Rectangular	32
C1	Phase	Hamming	32
D1	Phase	Rectangular	32
E1	Magnitude	Hamming	1024
F1	Magnitude	Rectangular	1024
G1	Phase	Hamming	1024
H1	Phase	Rectangular	1024

Table 4.3: Detailed listing of settings for stimuli construction in Experiment 1.

T_w (ms)	32	1024
M (samples)	512	16384
N (FFT length)	1024	32768
$T_w/8$ (ms)	4	128
$M/8$ (samples)	64	2048

The duration of each recorded signal is approximately 3 seconds[6].

4.2.1.2 Stimuli

Each of the recordings are processed through the STFT-based speech analysis-modification-synthesis system to retain either only phase information or only magnitude information. There are 8 types of stimuli for Experiment 1. The description of each type is provided in Table 4.2. Some extra details for stimuli construction are presented in Table 4.3.

4.2.1.3 Subjects

As listeners, we use 12 native Australian English speakers with normal hearing, all within the age group of 20–35 years. The subjects are different from those used for recording the speech stimuli.

[6]This time is inclusive of leading and trailing silence periods.

Table 4.4: Experiment 1: Consonant intelligibility (or, identification accuracy) of magnitude-only and phase-only stimuli for a small window duration of 32 ms (with $T_w/8$ frame shift).

Type of stimuli	Intelligibility (in %) for	
	Hamming window	Rect. window
Original	89.9	89.9
Magn. only	84.2 (A1)	78.1 (B1)
Phase only	59.8 (C1)	79.9 (D1)

4.2.1.4 Procedure

The perception tests for this experiment are conducted over 2 sessions. In the first session, the original speech signals and stimuli types A1, B1, C1, and D1 are presented. In the second session we present the original speech signals again, in addition to stimuli types E1, F1, G1, and H1.

The subjects are tested in isolation in a silent room. The reconstructed signals and the original signals (a total of 320 for each session) are played in random order via SONY MDR-V5000DF earphones at a comfortable listening level. The task is to identify each utterance as one of the 16 consonants. This way, we attain consonant identification accuracy (or, intelligibility) for each subject for different conditions. In both sessions, the subjects are first familiarised with the task through a short practice test. Session 1 (small window) results are provided in Table 4.4 and session 2 (large window) results are provided in Table 4.5. Results are averaged over the 12 subjects. The intelligibility of the original recordings is averaged over both sessions.

Responses are collected through software. The software displays the 16 aCa possibilities as well as an extra option for a null response. Participants are instructed to only choose the null response when they have no clue as to what the consonant may be. Responses are input via the keyboard in the form of numbers (1–17). Each audio file is presented once. No feedback is provided.

4.2.1.5 Results and Discussion

The following observations can be made from Tables 4.4 and 4.5:

Table 4.5: Experiment 1: Consonant intelligibility (or, identification accuracy) of magnitude-only and phase-only stimuli for a large window duration of 1024 ms (with $T_w/8$ frame shift).

Type of stimuli	Intelligibility (in %) for	
	Hamming window	Rect. window
Original	89.9	89.9
Magn. only	14.1 (E1)	13.3 (F1)
Phase only	88.0 (G1)	89.3 (H1)

1. For the large window duration of 1024 ms, the phase spectrum provides significantly more information than the magnitude spectrum for both the Hamming window function ($F[1, 11] = 2880.57$, $p < 0.01$) and the rectangular window function ($F[1, 11] = 1582.38$, $p < 0.01$). This observation is consistent with the results reported earlier in the literature [80, 100, 127].

2. The difference in intelligibility between magnitude-only stimuli constructed with a Hamming window and magnitude-only stimuli constructed with a rectangular window at a large window duration of 1024 ms is insignificant ($F[1, 11] = 0.63$, $p < 0.01$). The same can also be said for phase-only signals constructed with either window type at the large window duration ($F[1, 11] = 1.18$, $p < 0.01$).

3. For the small window duration of 32 ms, intelligibility of magnitude-only stimuli is significantly better than the phase-only stimuli when the Hamming window function is used ($F[1, 11] = 17.4$, $p < 0.01$), but these are comparable when the rectangular window function is used ($F[1, 11] = 2.91$, $p < 0.01$). Thus, if a rectangular window function is used in the STFT analysis-modification-synthesis system, the phase spectrum carries as much information about the speech signal as the magnitude spectrum, even for small window durations, which are typically used in speech processing applications.

4. For a small window duration of 32 ms, the Hamming window provides better intelligibility than the rectangular window for magnitude-only stimuli ($F[1, 11] = 29.38$, $p < 0.01$); while the rectangular window is better than the Hamming window for the construction of phase-only stimuli ($F[1, 11] = 176.30$, $p < 0.01$).

5. For a small window duration of 32 ms, the best intelligibility results from magnitude-only stimuli (obtained by using a Hamming window) are significantly better than the best results from phase-only stimuli (obtained using a rectangular window) ($F[1, 11] = 17.14$, $p < 0.01$).

These results can be explained as follows. The multiplication of a speech signal with a window function is equivalent to the convolution of the speech spectrum $X(k)$ with the spectrum $W(k)$ of the window function. The window's magnitude spectrum[7] $|W(k)|$ has a big main lobe and a number of side lobes. This causes two problems: 1) frequency resolution problem and 2) spectral leakage problem. The frequency resolution problem is caused by the main lobe of $|W(k)|$. When the main lobe is wider, a larger frequency interval of the speech spectrum gets smoothed and the frequency resolution problem becomes worse. The spectral leakage problem is caused by the sidelobes; the amount of spectral leakage increases with the magnitude of the side lobes. For magnitude-only utterances, we want to preserve the true magnitude spectrum of the speech signal. For the estimation of the magnitude spectrum, frequency resolution as well as spectral leakage are serious problems. Since the Hamming window has a wider main lobe and smaller side lobes in comparison to the rectangular window, the Hamming window provides a better trade-off between frequency resolution and spectral leakage than the rectangular window and, hence, it results in higher intelligibility for the magnitude-only utterances. For the estimation of the phase spectrum, it seems that the side lobes do not cause a serious problem; the smoothing effect caused by the main lobe appears to be more serious. It is because of this that the rectangular window results in better intelligibility than the Hamming window for phase-only utterances. Reddy and Swamy [124] have also recommended the use of a rectangular window function in the computation of the group delay spectrum, which is a frequency derivative of the phase spectrum.

For magnitude-only stimuli constructed with a small window duration, the best intelligibility is obtained for a Hamming window (type A1). For phase-only stimuli constructed with a small window duration, the best intelligibility is obtained when a

[7]The window's phase spectrum $\angle W(k)$ is a linear function of frequency and, hence, does not cause a problem in estimating the speech spectrum $X(k)$.

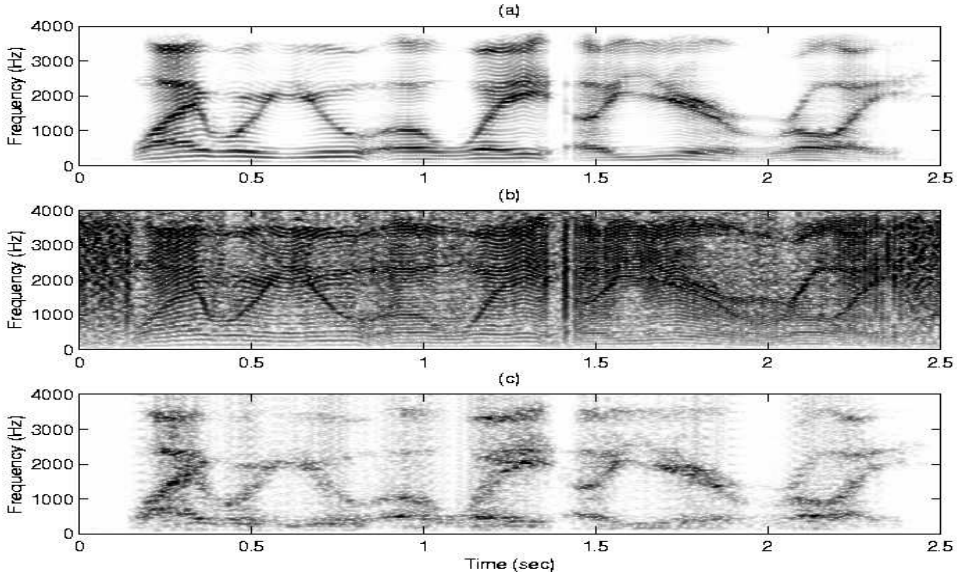


Fig. 4.3: (a) Spectrogram of the original speech sentence "Why were you away a year Roy?", (b) phase-only (type D1) spectrogram, and (c) magnitude-only (type A1) spectrogram.

rectangular window is used (type D1). In order to provide some details about the acoustic properties of these stimuli, we present, in Fig. 4.3, a spectrogram[8] for a sentence of speech and the corresponding magnitude-only (type A1) and phase-only (type D1) spectrograms[9]. The magnitude-only spectrogram is visually more similar to that of the original spectrogram than the phase-only spectrogram. In keeping the magnitude information, we also maintain the frame energies; thus, in the magnitude-only reconstruction, the short-time energy contour is preserved. The image contrast, therefore, in the magnitude-only spectrogram is similar to that of the original spectrogram[10]. In phase-only reconstruction, however, setting each frame's magnitude spectra to unity suppresses energy information, resulting in an almost constant energy contour over the

[8]Unless otherwise stated, all spectrograms in this chapter are constructed using a Hamming analysis window of length 32 ms, a time shift of 1 ms, a pre-emphasis coefficient of 0.97 and a dynamic range of 50 dB.

[9]Refer to Appendix A for an explanation of why we can see formant structure in the "phase-only" stimuli.

[10]When constructing a magnitude-only signal, the short-time phase spectra are replaced by random values. The magnitude spectra for these segments are identical to the original signal. However, when these short-time segments are overlapped and added during synthesis, their magnitude spectra are changed because the samples in the overlapping regions between the frames are no longer consistent (due to phase spectrum changes). Thus, upon re-analysis, the magnitude-spectra for the short-time segments differ to those of the original signal. Consequently, the magnitude-only spectrogram is not identical to the original signal spectrogram.

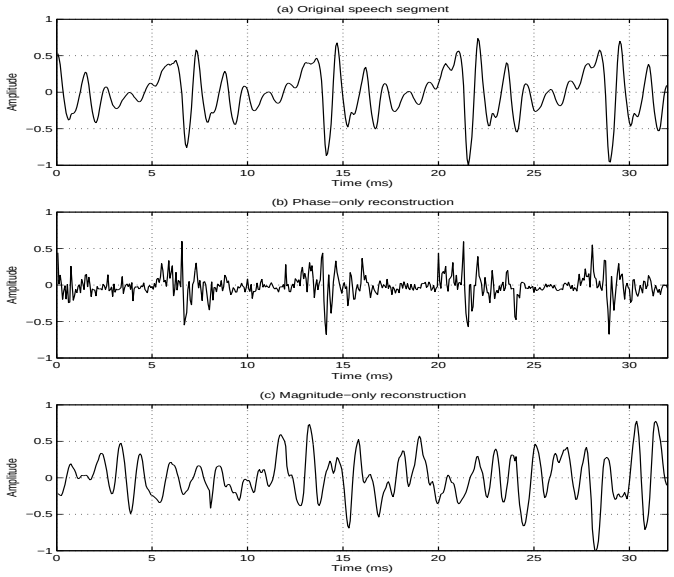


Fig. 4.4: (a) 32 ms segment of speech, (b) phase-only (type D1) reconstruction, and (c) magnitude-only (type A1) reconstruction.

duration of the reconstructed signal. This results in the silent parts at the beginning and end of the original utterance being heard as loud as the speech parts in the reconstructed signal.

Another interesting point can be observed through the time-domain plots in Fig. 4.4. This figure compares a 32 ms time frame of speech to its reconstructed magnitude-only (type A1) and phase-only (type D1) signals. In the phase-only reconstruction (Fig. 4.4(b)), pitch epochs are preserved, while they are lost in the magnitude-only reconstruction (Fig. 4.4(c)). Thus, the phase-only reconstruction preserves the pitch-related timing aspects.

As noted previously, the choice of analysis window type for the construction of magnitude-only and phase-only stimuli for large window durations is unimportant. For consistency, however, we recommend that the best analysis window functions used for constructing magnitude-only and phase-only stimuli at small window durations (as in stimuli types A1 and D1) should also be used for construction at large window durations (as in stimuli types E1 and H1). With this in mind, we introduce Fig. 4.5, which presents magnitude-only (type E1) and phase-only (type H1) reconstructions of the same speech. At large analysis window durations (we use $T_w = 1024$ ms), phase-only stimuli (type

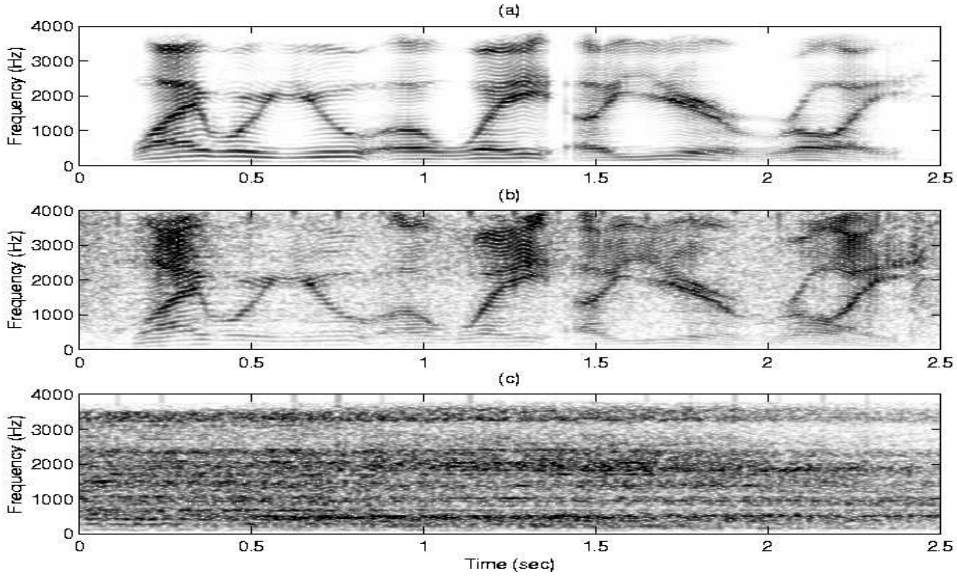


Fig. 4.5: (a) Spectrogram of the original speech sentence "Why were you away a year Roy?", (b) phase-only (type H1) spectrogram, and (c) magnitude-only (type E1) spectrogram.

H1) provide much better intelligibility than magnitude-only stimuli (type E1). Formant tracks are visible in the phase-only spectrogram (Fig. 4.5(b)), and are absent in the magnitude-only spectrogram (Fig. 4.5(c)). This explains the better intelligibility of phase-only stimuli over magnitude-only stimuli for large window durations.

We have performed a detailed analysis of confusion matrices for consonant identification obtained from Experiment 1. However, we have not been able to observe any consistent pattern. The confusion matrices are attached in Appendix B.

4.2.2 Experiment 2

In this experiment, we investigate more closely how intelligibility varies with window duration for magnitude-only and phase-only stimuli. We construct magnitude-only and phase-only stimuli using the analysis-modification-synthesis parameters used by Liu et al. [80] and compare the intelligibility scores to stimuli constructed using our best parameter settings suggested in Experiment 1. This comparison is made over a number of analysis window durations, ranging from 16 ms through to 2048 ms.

Table 4.6: Stimuli for Experiment 2.

Type of Stimuli	Retained Spectrum	Window Type	Frame Shift
A2	Phase	Hamming	$T_w/2$
B2	Magnitude	Hamming	$T_w/2$
C2	Phase	Rectangular	$T_w/8$
D2	Magnitude	Hamming	$T_w/8$

4.2.2.1 Stimuli

In their experiments, Liu et al. used a Hamming window and a frame shift of $T_w/2$ for construction of both phase-only and magnitude-only stimuli. These parameter selections differ from those suggested here, where we use a rectangular window for phase-only reconstruction, a Hamming window for magnitude-only reconstruction, and a frame shift of $T_w/8$ for both types of stimuli.

In this experiment we have 4 types of stimuli to compare at 8 analysis window durations (16, 32, 64, 128, 256, 512, 1024, and 2048 ms). Table 4.6 details the parameters used to construct each type of stimulus and the names subsequently used to reference them. Stimuli types A2 and B2 are constructed with Liu's settings and stimuli types C2 and D2 are constructed using the best settings suggested in Experiment 1.

4.2.2.2 Procedure

The experiment is split into 2 parts: in the first part, the intelligibility of stimuli types A2 and B2 are compared, while in the second part we compare the intelligibility of stimuli types C2 and D2. The details of the experimental setup are the same as those used in Experiment 1.

4.2.2.3 Results and Discussion

The intelligibility of stimuli types A2 and B2 over all analysis window durations are compared in Fig. 4.6(a). The intelligibility of magnitude-only (type B2) stimuli is almost 2 times better than that of the phase-only (type A2) stimuli at small analysis window durations. The intelligibility of magnitude-only (type B2) stimuli decreases

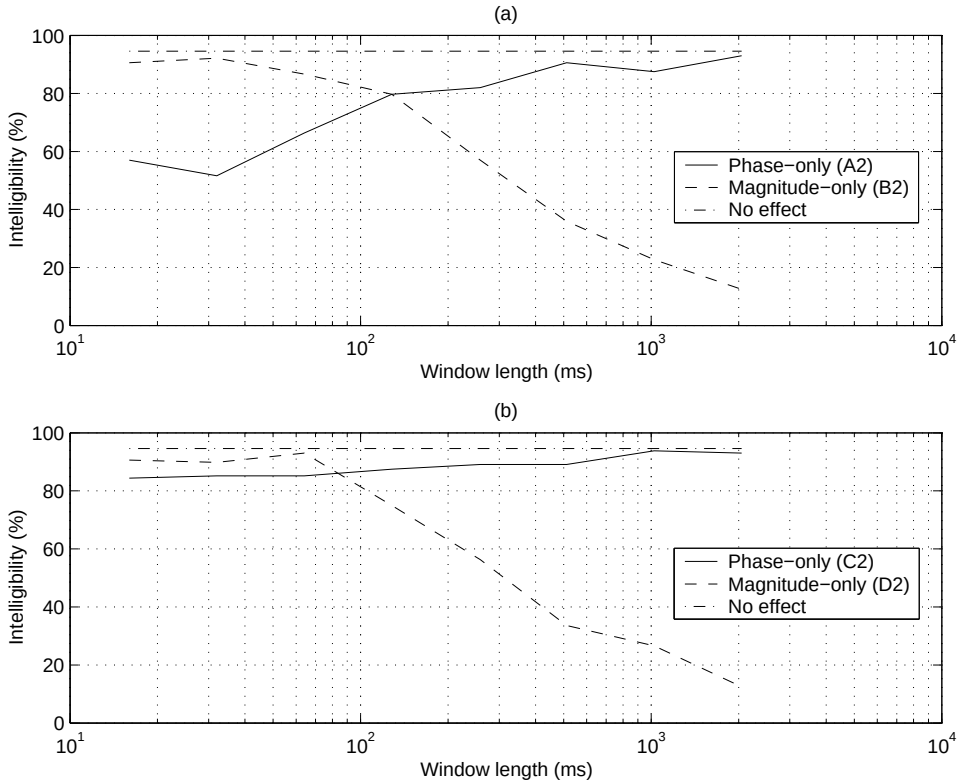


Fig. 4.6: Consonant identification performance (or, intelligibility) as a function of window duration for the magnitude-only and phase-only stimuli of Experiment 2. Intelligibility for the original utterances (without any modification) is shown by horizontal dot-dashed line. (a) Stimuli type A2 versus stimuli type B2, (b) Stimuli type C2 versus stimuli type D2.

while intelligibility of phase-only (type A2) stimuli increases as the analysis window duration increases. The crossover point is around 128 ms. The trends observed here are similar to those observed by Liu and his colleagues [80].

Intelligibility results for stimuli types C2 and D2 are shown in Fig. 4.6(b). It can be observed from this figure that for magnitude-only (type D2) stimuli, the intelligibility decreases with an increase in window duration. The trend of this relationship is similar to that for type B2 stimuli (Liu's method). For phase-only stimuli of type C2 (our method of constructing phase-only stimuli), the intelligibility scores are almost the same for all the window durations. Compare this to Fig. 4.6(a), where the intelligibility of type A2 (Liu's method of constructing phase-only stimuli) is much worse at small window lengths. The crossover point in Fig. 4.6(b) is between 64 ms and 128 ms.

Table 4.7 compares the intelligibility scores at 32 ms of all 4 types of stimuli. There

Table 4.7: Experiment 2: Comparison of consonant intelligibility (or, identification accuracy) of magnitude-only and phase-only stimuli constructed with our parameter settings and those settings used in [80] at 32 ms window duration.

Type of stimuli	Intelligibility (in %) for	
	Our settings	[80] settings
Original	94.6	94.6
Magn. only	89.8 (D2)	92.2 (B2)
Phase only	85.2 (C2)	51.6 (A2)

is no significant difference between the two types of magnitude-only stimuli (B2 and D2). With the help of a rectangular window and the analysis frame shift of $T_w/8$, the human recognition results for short-time phase-only stimuli is significantly improved at small window lengths.

4.2.3 Experiment 3

As seen in Experiment 2, our intelligibility results for phase-only stimuli are better than previously reported by Liu et al. [80]. We have shown that phase-only stimuli, constructed with different parameter settings than those used by Liu et al., provide intelligibility comparable to that of magnitude-only stimuli at small analysis window durations. It will be interesting to see the reasons why we get this improvement in intelligibility. Recall the 4 analysis-modification-synthesis parameters discussed in Experiment 1: window type, window duration, window shift and zero-padding. In this experiment, we determine the contribution that each parameter setting provides toward improving the intelligibility of signals reconstructed from short-time phase spectra.

4.2.3.1 Stimuli

The previous experiment demonstrated that it is possible to attain a good intelligibility score for phase-only stimuli with a small analysis window duration of 32 ms. Therefore, in this experiment, the window duration is set constant at 32 ms. A number of combinations of the other parameters are tested in order to ascertain their respective contribution to the intelligibility of phase-only stimuli. Table 4.8 details the parameters used to construct each type of stimuli and the names we will use to refer to them

Table 4.8: Comparison of consonant intelligibility (or, identification accuracy) for the phase-only stimuli used in Experiment 3 (T_w = 32 ms).

Type of Stimuli	Parameter Settings	Phase-only Intelligibility
A3	Hamming window, $T_w/2$ overlap	45.3%
B3	Rectangular window, $T_w/2$ overlap	76.6%
C3	Rectangular window, $T_w/8$ overlap	82.8%
D3	Rectangular window, $T_w/8$ overlap, zero-padding	85.9%

in this experiment. Note that stimuli types A3, B3 and C3 are constructed without zero-padding. The original recordings and the stimuli provide a total of 320 audio files.

4.2.3.2 Procedure

The 320 audio files are presented to each subject in a single session. The details of the experimental setup are the same as those used in Experiment 1.

4.2.3.3 Results and Discussion

The intelligibility scores are provided in Table 4.8. The scores indicate that the major contribution to overall intelligibility comes from the use of the rectangular window (stimuli type B3). Decreasing the frame shift provides a smaller improvement in intelligibility (stimuli type C3). The zero-padding also seems to contribute a slight improvement in intelligibility (stimuli type D3).

Fig. 4.7 presents the spectrogram of a sentence of speech with its reconstructed phase-only stimuli A3, B3, C3 and D3. The increasing clarity of the formant tracks in these spectrograms, from A3 through D3, is indicative of the corresponding trend in the intelligibility of these stimuli.

4.2.4 Experiment 4

In the experiments described so far, we have replaced the short-time phase spectrum by random numbers in order to reconstruct our magnitude-only stimuli. It will be interesting to see how making the phase spectrum zero for all frequencies will affect the

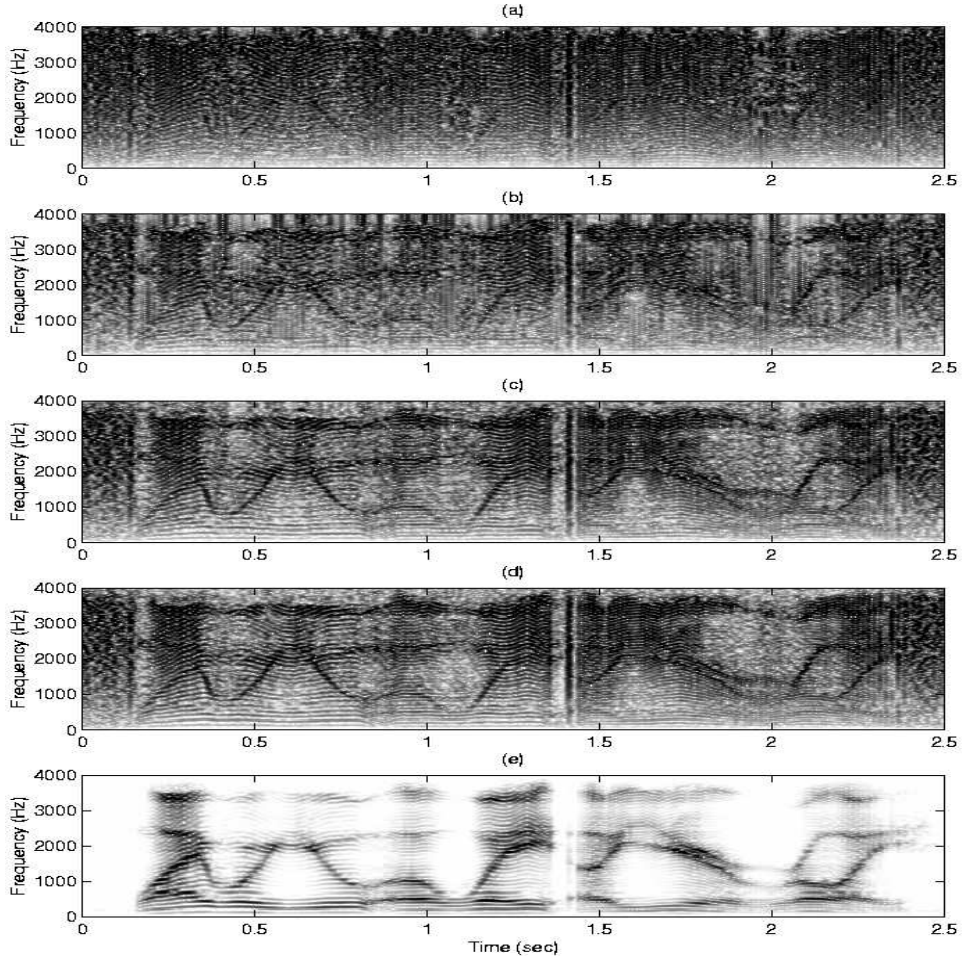


Fig. 4.7: The spectrograms of phase-only stimuli at an analysis window duration of 32 ms: (a) stimulus type A3, (b) stimulus type B3, (c) stimulus type C3, (d) stimulus type D3, and (e) spectrogram of the original speech sentence "Why were you away a year Roy?". Stimulus construction parameters are given in Table 4.8.

intelligibility of the magnitude-only stimuli. Therefore, in this experiment, we address the issue of replacing phase with random-phase and zero-phase in the construction of magnitude-only stimuli. In order to determine if there exists a significant intelligibility difference between these two choices, we conduct the following experiment.

4.2.4.1 Stimuli

Two sets of magnitude-only stimuli are constructed; one with zero-phase and the other with random-phase. A short window duration of 32 ms, frame shift of $T_w/8 = 4$ ms, and a Hamming analysis window are used. The 64 original utterances and the reconstructed

Table 4.9: Experiment 4: Consonant intelligibility (or, identification accuracy) of magnitude-only stimuli constructed with random-phase and zero-phase (with $T_w/8$ frame shift, and $T_w = 32$ ms).

Type of stimuli	Intelligibility (in %)
Original	91.1
Random phase	86.4
Zero phase	75.4

stimuli provide a total of 192 audio files.

4.2.4.2 Procedure

The 192 audio files are presented to each subject in a single session. The details of the experimental setup are the same as those used previously.

4.2.4.3 Results and Discussion

The results of this experiment are shown in Table 4.9. The intelligibility of the random-phase stimuli is significantly higher than those of the zero-phase stimuli.

We can explain this result by observing the spectrograms of Fig. 4.8. Setting the phase of each frame to zero introduces a periodicity (with a period equal to that of the frame shift) which manifests itself as horizontal lines on the spectrogram (Fig. 4.8(b)), producing a high pitched, unnatural sounding speech. The resulting stimuli is so unnatural that it seems to have an adverse effect on intelligibility. The subjects described the zero-phase stimuli as 'harsh' and 'robotic'.

To explain the periodicity in the zero-phase stimuli, we refer to Fig. 4.9. To create the waveform in Fig. 4.9(b), we perform the STFT analysis on the 32 ms frame of speech shown in Fig. 4.9(a), replace its phase spectrum with zero-phase, then apply an inverse Fourier transform. As expected, we get an autocorrelation-like sequence. Thus, zero-phase puts more energy toward the beginning of the frame. Note that the first sample in the frame, which is similar to the zeroth order autocorrelation coefficient, has the largest value. The waveform in Fig. 4.9(c) is constructed in a similar manner except that the phase spectrum is replaced by random-phase. The random-phase distributes the energy across the time-axis. Fig. 4.10 illustrates the results of using zero-phase

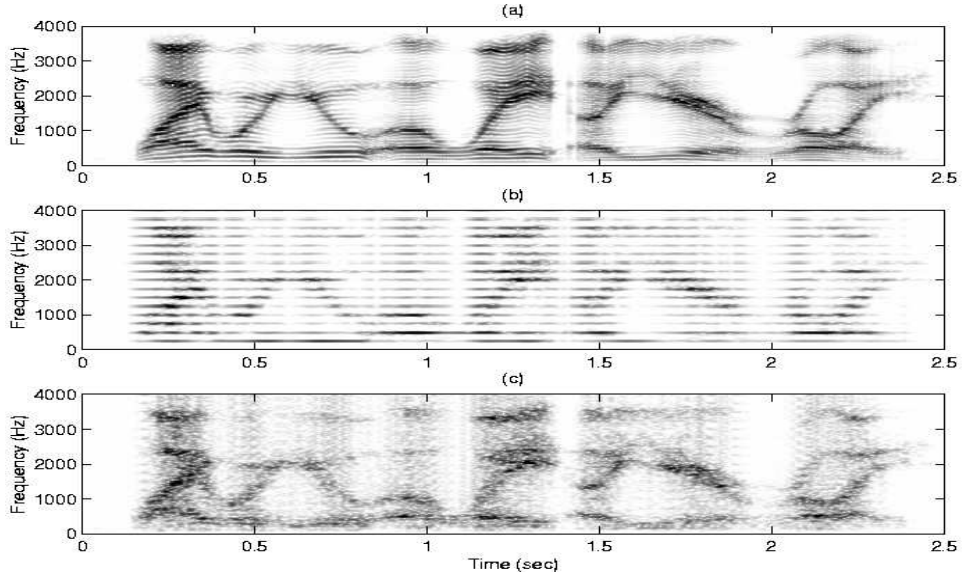


Fig. 4.8: Spectrograms of (a) the original speech sentence "Why were you away a year Roy?", (b) magnitude-only, zero-phase stimuli, and (c) magnitude-only, random-phase stimuli (with $T_w/8$ frame shift, and $T_w = 32$ ms).

and random-phase for the reconstruction of a magnitude-only stimuli. The large peak of the zero-phase signal in Fig. 4.9(b) now repeats itself every 4 ms (1/8 of 32 ms) in Fig. 4.10(b). This periodicity is not present in the random-phase signal of Fig. 4.10(c). Consequently, the random-phase stimuli are 'easier to listen to'. Subjects describe the random-phase stimuli as 'natural', 'mellow', and 'breathy'. This naturalness is most likely attributed to the lower peak factor of the random-phase signal [129].

4.2.5 Experiment 5

In this experiment, we explore the use of partial phase spectrum information for intelligible signal reconstruction.

4.2.5.1 Stimuli

In addition to the phase-only stimuli (type D1 from Experiment 1), we create the following types of stimuli from the original 64 utterances, using a 32 ms rectangular analysis window:

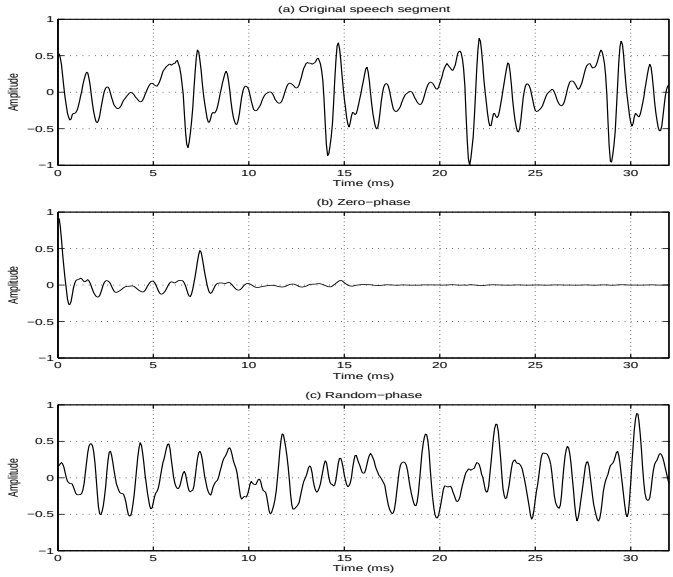


Fig. 4.9: The result of processing one frame of speech, retaining only its magnitude information.

1. **IFD-only stimuli**: We take the phase spectrum from each short-time section and randomise it across frequency, such that the IFD is preserved. In other words, add the same random sequence (across frequency) to the phase spectrum values of each frame. For example, consider a frame of length M and a DFT length of $N = 2M$. Add a random sequence to the phase values in the first $M+1$ DFT bins (i.e., bin numbers 0 to M). To determine the remaining $M-1$ phase values (i.e., bin numbers $M+1$ to $N-1$), take the new phase values from bins 1 to $M-1$ then reverse the sign and reverse the order of the numbers. That is, given the new phase values for the first $M+1$ bins, calculate the remaining bin phase values by $\psi(n) = -\psi(N-n)$, where $n = M+1, M+2, \ldots, N-1$ is the bin number. The resulting phase spectra are used in place of the original phase spectra in the reconstruction algorithm (and magnitude spectra are set to unity).

2. **GDF-only stimuli**: In a similar vein, we take the original phase spectra and randomise them across time, such that the GDF is preserved. That is, generate a random sequence whose length is equal to the number of frames in the utterance, then add this same sequence to the time-trajectory of the phase spectrum values for each DFT bin. Remember to do this for the phase values in the first $M+$

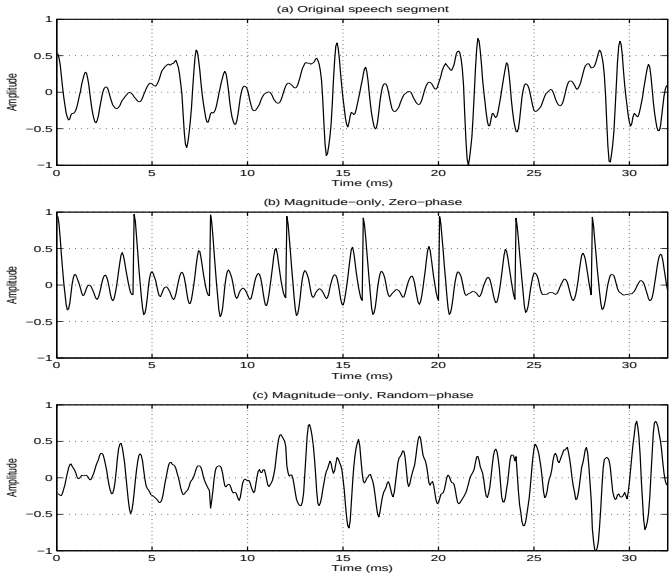


Fig. 4.10: (a) 32 ms segment of speech, and its magnitude-only reconstruction using (b) zero-phase and (c) random-phase.

1 DFT bins (for each frame), then calculate the remaining bin phase values as described above. Reconstruction is performed with the resulting phase spectra (and magnitude spectra are set to unity).

3. **IFD+GDF stimuli**: We reconstruct a signal from the knowledge of both the IFD and GDF. In order to do this, we must first reconstruct the phase spectra from these known quantities. Notice that the first-segment phase spectrum can only be reconstructed to within a time-shift of the original first-segment phase spectrum, since all we know about it is the GDF. The remaining segments are reconstructed in relation to this segment. Consequently, we cannot recover the original phase spectra[11]. To construct phase spectrum values from the GDF and IFD we do the following: The phase value for DFT bin number 0 is set to zero in every frame. The remaining phase spectrum values (for each frame) are calculated by cumulatively summing the GDF across DFT bins 1 to M. We then shift all of these values by a constant in each frame (dependent on the frame), so that the

[11]The phase spectrum values are only meaningful in the context of a fixed-time reference. All that we have lost in this reconstructed signal is the original fixed-time reference. Time referencing is now in relation to the phase spectrum values of the first frame (i.e., we still have a time reference, but it is different to that of the original phase spectra values).

phase spectrum changes over time for one particular DFT bin (this can be any bin, the decision is arbitrary) are the same as in the original signal (i.e., we use the IFD values for only one bin). The values for bins $M+1$ to $2M-1$ (i.e., $N-1$) are calculated as previously described[12]. Reconstruction is performed with the altered phase spectra (and magnitude spectra are set to unity).

4.2.5.2 Procedure

We use a subset of 5 listeners from the 12 used in Experiment 1. The reconstructed signals and the original signals are played in random order to each listener. The details of the experimental setup are the same as those used in Experiment 1.

4.2.5.3 Results and Discussion

The average consonant identification scores[13] are given in Table 4.10. Reconstructing stimuli from knowledge of only the IFD or the GDF results in poor intelligibility; the intelligibility of the stimuli reconstructed from knowledge of only the IFD is 50.94% and the intelligibility of the stimuli reconstructed from knowldege of only the GDF is 53.75%. However, when we create stimuli using knowledge of both the IFD and the GDF, intelligibility on par with the stimuli reconstructed from the original phase spectra is achieved; the intelligbility of the stimuli reconstructed from knowledge of both the IFD and GDF is 85.63% and the intelligibility of the stimuli reconstructed from knowldege of the original phase spectra is 86.88%. The results imply that both IFD and GDF are required for good intelligibility from the phase spectrum. Furthermore, the intelligibility score of the original signals is by far the best (95.31%). That is, all of the phase spectrum and the magnitude spectrum information must be retained for superior intelligibility. This will be addressed further in Experiment 6.

[12]Note that this is only one way of reconstructing the phase spectrum values. It is also possible to reconstruct by using the GDF values for only one frame then to extrapolate the phase values for the other frames by using the IFD values for all DFT bins.

[13]The intelligibility of the original signals and the phase-only stimuli are both higher than that reported in the Experiment 1. This can most likely be attributed to the following two reasons: 1) these results are based on a subset of listeners used in Experiment 1, and 2) this experiment was conducted at a different time and location than Experiment 1. Regardless, the absolute intelligibility scores are not that important; it is the relative intelligibility that is more interesting.

Table 4.10: Results from Experiment 5. Average consonant intelligibility of stimuli constructed from partial phase spectrum information (rectangular analysis window of duration 32 ms used in the STFT analysis).

Type of stimuli	Intelligibility score
Original	95.31%
IFD–only	50.94%
GDF–only	53.75%
IFD+GDF–only	85.63%
Phase-only	86.88%

4.2.6 Experiment 6

This experiment serves to quantify the intelligibility provided by the phase spectrum and the magnitude spectrum components of the STFT under noisy conditions.

4.2.6.1 Stimuli

In accordance with the results of Experiment 1, we use a rectangular analysis window to construct the phase-only stimuli and a Hamming analysis window to construct the magnitude-only stimuli. Once again, the duration of the analysis window is 32 ms. This time, however, the 64 original utterances are contaminated with white noise over several signal-to-noise ratios (SNRs) of -10 dB, 0 dB, 10 dB, 20 dB and ∞ dB (i.e., no noise added).

4.2.6.2 Procedure

We employ a subset of 3 listeners from the 12 used in Experiment 1. The reconstructed signals and the noisy original signals are played in random order to each listener. The details of the experimental setup are the same as those used previously.

4.2.6.3 Results and Discussion

The average consonant identification scores are plotted in Fig. 4.11. The results indicate that the intelligibility of both the phase-only stimuli and the magnitude-only stimuli degrade at a similar rate under decreasing SNR value. While the intelligibility provided

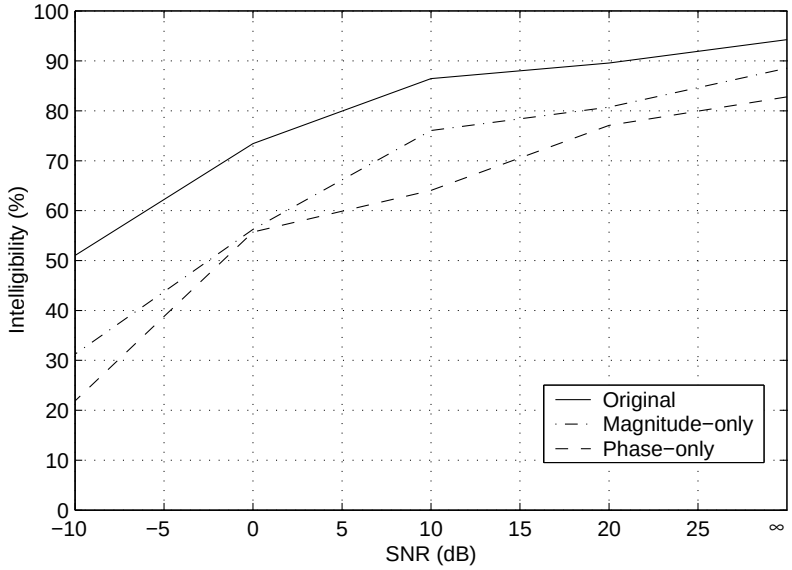


Fig. 4.11: Results from Experiment 6. Average consonant intelligibility of phase-only and magnitude-only stimuli constructed from white-noise contaminated speech over several SNRs (phase-only and magnitude-only stimuli are constructed with a rectangular and Hamming analysis window respectively, of duration 32 ms). Average intelligibility scores for the original (noisy) speech are also provided.

by the original signals also degrades at a similar rate, the intelligibility is consistently better than that provided by the phase-only stimuli and the magnitude-only stimuli. It is particularly interesting to see that the intelligibility provided by the original signals is far better than that provided by the magnitude-only stimuli. This result seems to be at odds with the common practice in ASR; which is to discard the phase spectrum in favour of features that are derived only from the magnitude spectrum. Should ASR features also encapsulate information about the phase spectrum? According to these perception results, robustness in human speech recognition requires that both the magnitude spectrum and the phase spectrum be retained (where a frame duration of 32 ms is used in the STFT analysis). Thus, a feature set that represents information from both the magnitude spectrum and the phase spectrum may result in improved ASR performance.

4.3 Conclusion

In this chapter, the relative importance of the (short-time) magnitude spectrum and phase spectrum on speech perception was investigated. Human perception experiments were conducted to measure intelligibility of speech stimuli reconstructed either from the original magnitude spectra or the original phase spectra. The experiments reported here demonstrate that even for small window durations of 32 ms, the phase spectrum can contribute to speech intelligibility as much as the magnitude spectrum if the analysis-modification-synthesis parameters are properly selected. These results confirm the findings of Cox and Robinson [24], who informally demonstrated that many of the important features of speech are preserved in the short-time phase spectrum at small window durations (Cox and Robinson specifically used a window duration of 25.6 ms).

Since the speech processing in ASR applications is done frame-wise over small analysis window durations (20-40 ms), it now seems plausible to investigate the use of the phase spectrum to extract features for these applications (see Chapter 7). Our experiments have shown that, in the absence of all other spectral information, an intelligible signal can be reconstructed with knowledge of both the time-derivative and frequency-derivative (i.e., IFD and GDF) components of the phase spectrum. However, this is not the case if only one of these components is known. This result suggests that a possible avenue of future research could be to derive a feature representation from both the IFD and GDF information. Also, according to our perception experiments, robustness in human speech recognition requires that both the magnitude spectrum and the phase spectrum be retained. Thus, a feature set that represents information from both the magnitude spectrum and the phase spectrum may result in improved ASR performance.

Note that there is not enough consonant data to carry out an ASR test. Thus, in the following chapter, we do ASR tests on the ISOLET database.

Chapter 5

ASR with Speech Reconstructed from Short-time Phase Spectra

In the previous chapter, we measured human intelligibility of speech stimuli reconstructed either from the short-time magnitude spectra (magnitude-only stimuli) or the short-time phase spectra (phase-only stimuli) of a speech stimulus. We demonstrated that, even for small analysis window durations of 20–40 ms (of relevance to ASR), the short-time phase spectrum can contribute to speech intelligibility almost as much as the short-time magnitude spectrum. Following on from that, in this section, we perform ASR on magnitude-only and phase-only stimuli. The purpose of doing so is to determine if the ASR recognition scores are consistent with the human intelligibility scores.

Publications from this research: [10].

5.1 Experiments

The ASR experiments are performed with magnitude-only and phase-only stimuli created from speech in the ISOLET and Aurora II databases. The magnitude-only and phase-only stimuli are constructed using a Hamming and rectangular analysis window respectively, with a frame shift of $T_w/8$. This is done at both a small analysis duration of 32 ms and a large analysis duration of 1024 ms.

We use an MFCC-based front-end with the following settings:

- Frame duration: 20 ms
- Frame shift: 10 ms
- Window type: Hamming
- Pre-emphasis coefficient: 0.97
- Frequency range: 0–4 kHz
- Number of filter-bank energies: 24
- Number of cepstral coefficients: 12 (excluding the zeroth coefficient).

Using the Cambridge Hidden Markov model (HMM) Toolkit (HTK) [151], we do both training and testing with the original speech, magnitude-only speech and phase-only speech (i.e., three sets of training and testing). Systems are trained with SNR=∞ and tested over a range of SNRs. Note that in the case of magnitude-only and phase-only stimuli, the noise must be added to the original test set speech before modification.

5.1.1 Isolated Word Task

The ISOLET database is an isolated-word, speaker-independent task, sampled at 8 kHz. The vocabulary is 26 English alphabet letters. Two repetitions of each letter are recorded for each speaker. Speakers are divided into 2 sets: 90 for training, 30 for testing. Each word is modeled by a HMM with 5 emitting states and 5 Gaussian mixtures per state. The grammar is such that the likelihood of each word is the same. There is no need to set the word insertion probability since only one word can occur per utterance. Although the vocabulary is relatively small, this is a difficult task as all words are short and highly confusable. Word recognition scores, over a range of test SNRs, are given in Fig. 5.1.

5.1.2 Connected Digit Task

Aurora II caters for speaker-independent experiments using several noise types and SNRs. Speech consists of digit sequences derived from the TI digit database down-sampled to 8 kHz and filtered with a G.712 characteristic [155]. Each digit (0-9) is

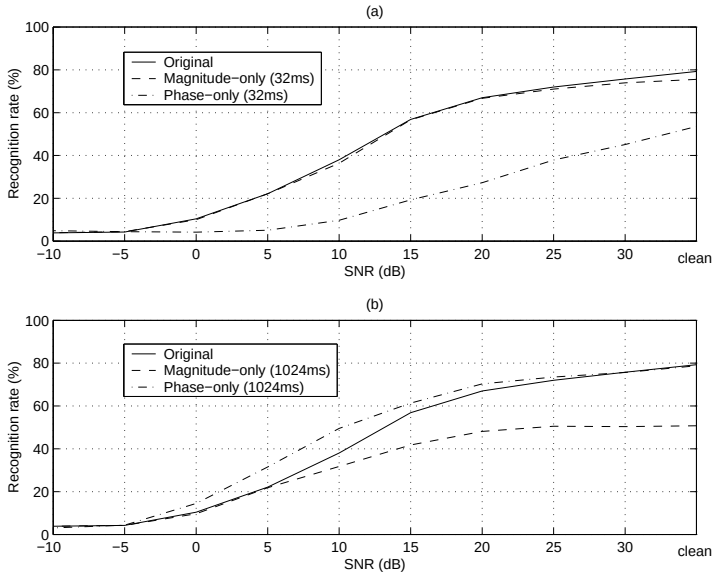


Fig. 5.1: Word accuracy versus SNR for ISOLET. White noise results provided. The magnitude-only and phase-only stimuli are constructed with an analysis window durations of (a) 32 ms and (b) 1024 ms.

modeled using a HMM with 16 emitting states and 3 Gaussian mixtures per state. We train with the clean training set (8440 utterances). The test set (28028 utterances) is divided evenly among 7 SNRs (∞, 20, 15, 10, 5, 0, -5 dB) and 4 noise types (subway, babble, car, exhibition). We use a unigram language model, where the probability of each word is equal (i.e., no grammar). The word insertion probability is set to 0. Word accuracy scores (which take into account insertions and deletions) for small and large analysis durations are given in Fig. 5.2 and Fig. 5.3.

5.1.3 Results and Discussion

For a reconstruction analysis duration of 1024 ms, magnitude-only and phase-only recognition scores on both databases agree with the trends observed in the human perception experiments; that is, for a long analysis window, phase-only stimuli are very intelligible and magnitude-only stimuli are unintelligible. Thus, we will not discuss these results further.

For an analysis duration of 32 ms, magnitude-only scores for both databases are also as expected. However, for the same analysis window size, phase-only results for both

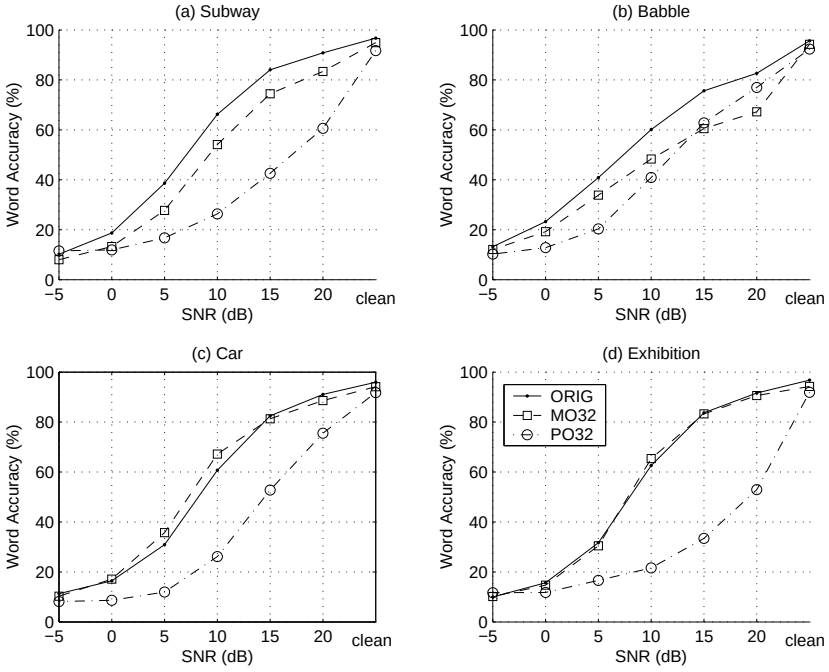


Fig. 5.2: Word accuracy versus SNR for Aurora II. Four noise types are investigated. The magnitude-only (MO32) and phase-only (PO32) stimuli are constructed with an analysis window duration of 32 ms. ORIG - original speech.

databases are worse than expected. While the phase-only stimuli sound quite intelligible (as proven in the intelligibility tests), ASR tests on both ISOLET and Aurora II result in poor word accuracy.

The poor ASR performance for phase-only stimuli constructed with a small analysis window duration can be attributed to a small dynamic range of the phase-only stimuli magnitude spectra[1]. This translates to less discriminating power for the MFCC features, because they are derived from the magnitude spectra. To demonstrate, we take an MFCC vector from original speech as well as the vectors at the corresponding locations in the magnitude-only and phase-only stimuli (32 ms). A zero is appended to the beginning of each of these vectors to account for the absence of the zeroth cepstral coefficient. An inverse discrete cosine transform (DCT) is then calculated for each vector

[1]When constructing a phase-only signal, the magnitude spectral values for each segment are set to unity. The phase spectra for these segments are identical to the original signal. However, when the segments are overlapped and added during synthesis, their magnitude spectra are changed because the samples in the overlapping regions between the segments are no longer consistent. Thus, upon re-analysis, each frame has a non-unity magnitude spectrum, the shape of which is determined by the original phase spectra of the frame and its surrounding frames. Refer to [107] for a detailed explanation of why we can see formant structure in the phase-only stimuli.

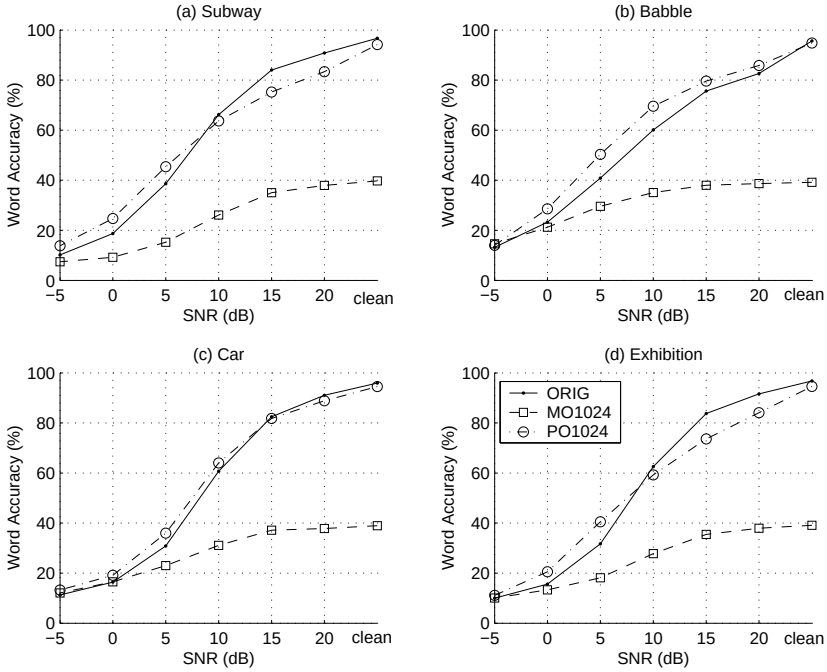


Fig. 5.3: Word accuracy versus SNR for Aurora II. Four noise types are investigated. The magnitude-only (MO1024) and phase-only (PO1024) stimuli are constructed with an analysis window duration of 1024 ms. ORIG - original speech.

(Figs. 5.4(a,c,e)). The inverse DCT allows us to view the reduced spectral information with which the models are effectively trained; that is, a smoothed magnitude spectrum. As can be seen, the dynamic range of the phase-only stimuli magnitude spectra is much smaller than the dynamic range in the magnitude spectra of the magnitude-only stimuli and the original speech. For the Aurora II task, this small amount of discriminating power (Fig. 5.4(e)) is sufficient to obtain good word accuracy in clean conditions, but insufficient once noise is added[2]. For the highly confusable ISOLET task[3], the small amount of discriminating power is not even sufficient in clean conditions. The poorer performance in lower SNRs is best explained by viewing Figs. 5.4(b,d,f), which were generated in a similar manner to Figs. 5.4(a,c,e); however, this time we have added 20dB of white noise to the original speech before stimuli construction. This was done at 20 different seed values for the random noise generator. Note that the phase-only

[2]The vocabulary of the Aurora II task consists of only 10 digits (0-9) which do not consist of many phonetic similarities. Thus, in clean conditions, it is quite possible that a feature set with poor discrimination capabilities can still provide good word accuracy.

[3]The ISOLET task is more difficult because the vocabulary of 26 letters consists of many phonetic similarities.

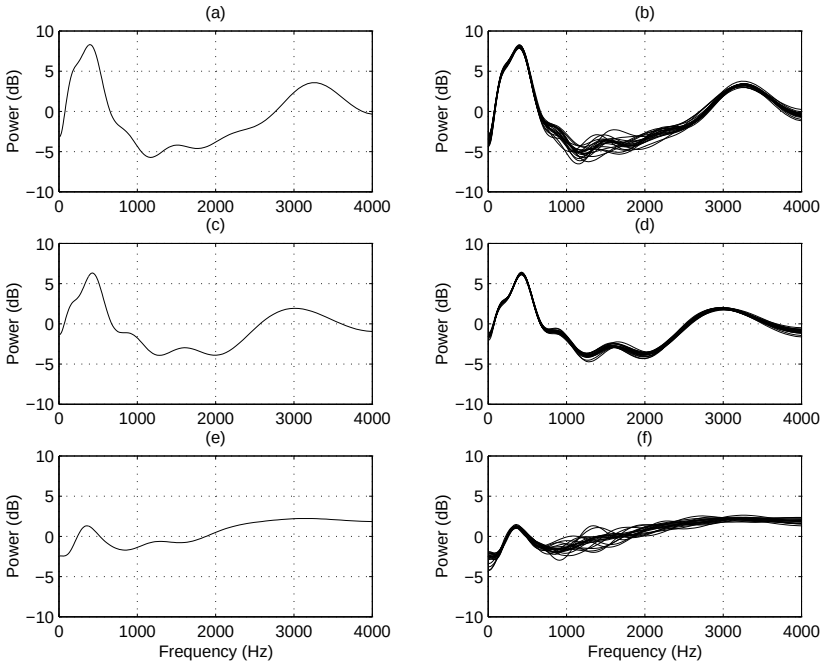


Fig. 5.4: Analysis used to determine why phase-only (32 ms) ASR results are worse than expected. The figures in the left column show magnitude spectra produced by inverse DCT of an MFCC vector from (a) original speech and its (c) magnitude-only and (e) phase-only stimuli. The figures in the right column show the magnitude spectra for 20 observations of white noise at 20dB SNR for (b) original speech and its (d) magnitude-only and (f) phase-only stimuli. The magnitude-only and phase-only stimuli are constructed with an analysis window duration of 32 ms.

stimuli magnitude spectrum exhibits much more variability over its dynamic range than the magnitude-only stimuli magnitude spectrum exhibits over its dynamic range; which explains the vulnerability of the phase-only stimuli to noise.

Since both magnitude-only and phase-only stimuli are almost equally intelligible to humans, and ASR recognises one much better than the other, it could be possible that the MFCC feature set is inadequate. That is, there seems to be some discriminating information in the phase spectrum part of the speech signal that is not being captured by the MFCC representation. These results demonstrate that there is a need for further research in the field of feature extraction.

Chapter 6

Iterative Reconstruction of Speech

Although the phase spectrum[1] has yet to be proven useful for ASR[2], it has successfully been used for many other tasks, such as formant extraction [31,38,86,116], fundamental frequency extraction [1, 22, 87, 89, 133], and iterative signal reconstruction [47, 50, 83, 90, 100, 119, 137, 139, 147, 150]. In this chapter, we concern ourselves with iterative signal reconstruction (in the previous chapter we used a non-iterative technique). Formant extraction and fundamental frequency extraction are addressed in Chapter 3. Any researcher with an interest in the phase spectrum should be aware of the flurry of activity that occurred in the 1980's in the area of signal reconstruction from magnitude spectrum and phase spectrum. In fact, what we find most interesting is that the phase spectrum alone can be used for perfect signal reconstruction (to within a scale factor), yet it has not been used successfully for ASR.

First and foremost, this chapter serves as a tutorial on the topic of iterative, one dimensional[3], signal reconstruction (specifically speech signals). Secondly, we provide the results of some further experimentation which may be interesting from an ASR view-

[1] As in previous chapters, the modifier 'short-time' is implied when mentioning the magnitude spectrum and the phase spectrum.

[2] There have been some attempts at using the phase spectrum as a representation for ASR feature extraction [29, 53, 54, 88, 108, 117, 142].

[3] The theory of one dimensional signal reconstruction can be extended to multidimensional signal reconstruction [50, 51, 100].

point. While iterative signal reconstruction has been extensively researched and documented [47, 50, 83, 90, 100, 119, 137, 139, 147, 150], we wish to recast some well-established results for the benefit of new researchers and those who desire a short, yet comprehensive, review of the subject. We believe that an appreciation for how the phase spectrum has proven useful in iterative signal reconstruction will motivate readers to investigate the potential for its use in ASR.

In general, the magnitude and phase spectra are both required in order to uniquely specify a signal. Under certain conditions, however, one can establish relationships between the magnitude and phase spectrum components. A well known result is the relationship of log magnitude spectrum and phase spectrum through the Hilbert transform for minimum and maximum-phase signals [99, 119, 147] (see Section **??**). However, finite duration speech signals are mixed-phase, all-zero signals. Hayes et al. [50] have determined the conditions under which such signals can be uniquely specified to within a scale factor by the phase spectrum, while Van Hove et al. [139] have determined that such signals can be uniquely specified by the signed-magnitude spectrum (magnitude spectrum with one bit of phase spectrum information). Given the phase spectrum, or signed-magnitude spectrum, the iterative framework in Fig. 6.1 can be used to reconstruct the signal (where the known spectral information is determined over the entire duration of the signal). This algorithm is equally valid for reconstruction of a signal from short-time segments (Fig. 6.3).

In this chapter, we provide several examples which demonstrate the application of these established iterative signal reconstruction algorithms. We also wish to draw attention to the results of some additional experimentation – since our interest lies in the phase spectrum, we look further into signal reconstruction from the phase spectrum, specifically partial phase spectrum information[4]. The train of thought is that if a signal can be reconstructed from knowledge of only the phase spectrum, why then is the phase spectrum useless for extracting ASR features? If so much information is contained in the phase spectrum, then it may be possible to capture and use it to improve the per-

[4]This work is different to the partial phase spectrum experiments conducted by Yegnanarayana and his colleagues [150]. See Section 6.2 for more details.

formance of ASR systems. However, using the phase spectrum directly for ASR has proven difficult due to phase-wrapping and other problems [31, 86, 145]. Here we consider some alternative representations of the phase spectrum. The phase spectrum has two independent variables: frequency and time. Thus, while there may be many ways to represent the information present in the phase spectrum, two representations that first come to mind are those that can be obtained either by taking its frequency-derivative (group delay function, GDF) or its time-derivative (instantaneous frequency distribution, IFD). We want to determine if an intelligible signal can be reconstructed given that we only know either the GDF or the IFD information. We also want to determine if we can reconstruct an intelligible signal given that we only know the phase spectrum sign information. The justification for this further experimentation is as follows: if the use of either the phase spectrum sign, GDF, or IFD information results in intelligible signal reconstruction, this would advocate the possible use of the partial information as a basis for an ASR feature set.

The chapter outline is as follows: In Section 6.1, we review some established iterative algorithms that attempt to reconstruct a signal from phase spectrum, magnitude spectrum or signed-magnitude spectrum information (where the spectrum is determined over the entire duration of the signal or on a short-time basis). We highlight the fact that knowledge of the phase spectra is enough for unique signal reconstruction (to within a scale factor), while the same is not true for magnitude spectra (the magnitude spectra must be accompanied by phase spectrum sign information for unique reconstruction). In Section 6.2, we explore the use of partial phase spectrum information, in the absence of all magnitude spectrum information, for intelligible signal reconstruction.

Publications from this research: [12].

6.1 An Overview of Iterative Reconstruction Algorithms

In this section, we review some well-established signal reconstruction algorithms. We see that under mild conditions, a finite duration signal can be reconstructed to within a scale factor by its phase spectrum (where the phase spectrum is determined over

the duration of the signal or on a short-time basis). This is not true for the magnitude spectrum. However, if the magnitude spectrum is accompanied by some phase spectrum information, then unique reconstruction is possible.

6.1.1 Reconstruction from Partial Fourier Transform Information

6.1.1.1 Reconstruction from Phase Spectrum

In practical terms, the theorem proposed by Hayes et al. [50] (1-d case) is stated as follows:

Theorem 1 *A sequence which is known to be zero outside the interval* $0 \leq n \leq (M-1)$ *is uniquely specified to within a scale factor by* $(M-1)$ *distinct samples of its phase spectrum in the interval* $0 < w < \pi$ *if it has a z-transform with no zeros on the unit circle or in conjugate reciprocal pairs [50].*

The reconstruction procedure is based on the iterative framework in Fig. 6.1. In the time domain, all samples outside of the interval $0 \leq n \leq (M-1)$ are set to zero (i.e., finite-time constraint). In the frequency domain, the known phase spectrum samples are imposed. In order to obtain $M-1$ distinct phase spectrum samples in the interval $0 < w < \pi$, a discrete Fourier transform (DFT) of length $N \geq 2M$ is required. In our experiments, we use a DFT length of $N = 2M$ (where M is a power of 2). Repeated transformations between the time and frequency domains, with the continued enforcement of the above constraints, provides a signal that converges to a scaled version of the original signal [137].

This algorithm has been used to reconstruct the signal in Fig. 6.2(a) from its phase spectrum. The magnitude spectrum is initially set to unity for all ω. The reconstructed signal after 200 iterations is shown in Fig. 6.2(b). The mean squared error (MSE) between the original and reconstructed signals[5] is non-increasing with each iteration (Fig. 6.2(c)).

[5]In all experiments, the signals reconstructed from phase spectrum are rescaled to vary over the same range as the original signal. MSE measurements are taken after rescaling.

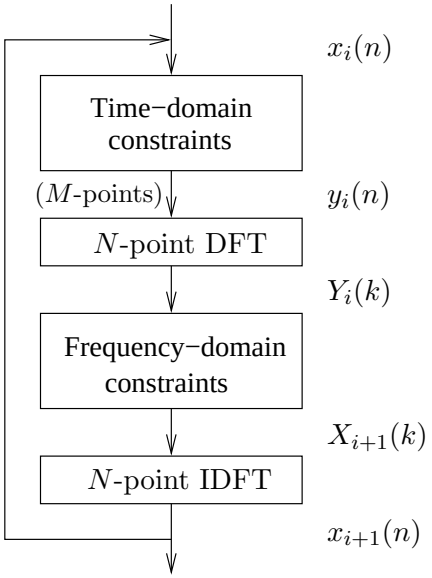


Fig. 6.1: Iterative framework used for reconstruction of an M-point sequence from phase spectrum, magnitude spectrum or signed-magnitude spectrum (where $N \geq 2M$).

6.1.1.2 Reconstruction from Magnitude Spectrum

Unlike the phase spectrum, the magnitude spectrum can not uniquely specify a 1-d sequence. The reason is as follows. If we express the system function as:

$$X(z) = G \prod_{k=1}^{N} (1 - b_k z^{-1}), \tag{6.1}$$

where G is real, then the square of the magnitude function is expressed as [99]:

$$\begin{aligned} P(z) &= X(z)X^*(1/z^*) \\ &= \prod_{k=1}^{N} (1 - b_k z^{-1})(1 - b_k^* z). \end{aligned} \tag{6.2}$$

The zeros occur in conjugate reciprocal pairs. Thus the zeros of $S(z)$ can not be determined by the magnitude spectrum alone. Therefore, the phase spectrum can not be determined by the magnitude spectrum. Consequently, if the known magnitude spectrum (instead of the phase spectrum) is imposed in the iterative reconstruction algorithm, it will not converge to the original signal.

If the signal is assumed to be minimum or maximum phase, then there is no am-

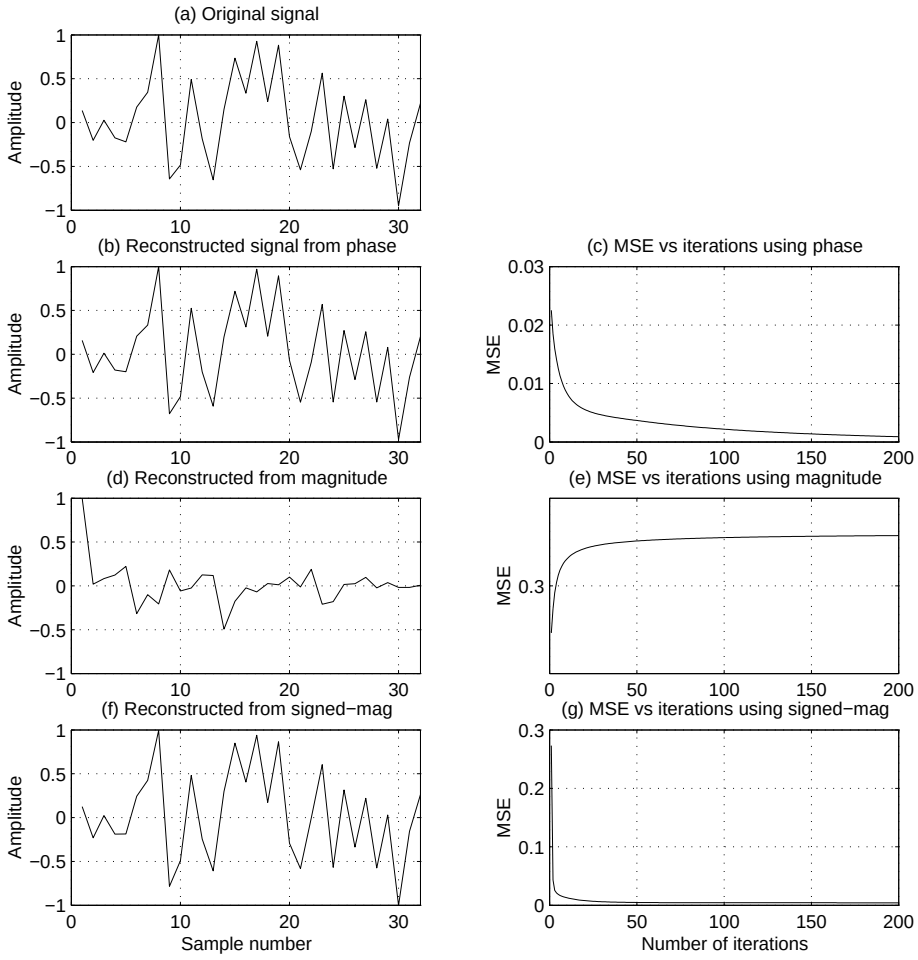


Fig. 6.2: Results of experiments in Section 6.1.1. (a) is the original signal. (b), (d) and (f) show reconstructed signals after 200 iterations (these signals are scaled to vary over the same range as the original signal). (c), (e) and (g) show plots of the respective MSE values at every iteration.

biguity in determining the zeros from magnitude[6]. In the case of mixed-phase signals, Van Hove et al. [139] have shown that this ambiguity can be resolved by imposing some phase spectrum information (see Section 6.1.1.3).

We reconstruct the signal in Fig. 6.2(a) from its magnitude spectrum. The phase spectrum is initialised with random values. After 200 iterations, the reconstructed signal does not resemble the original signal (Fig. 6.2(d)&(e)).

[6]Reconstruction algorithms for minimum phase signals can be found in [119, 147].

6.1.1.3 Reconstruction from Signed-Magnitude Spectrum

The principal phase spectrum values are calculated by:

$$\mathrm{ARG}[X(\omega)] = \arctan(X_I(\omega)/X_R(\omega)), \tag{6.3}$$

where the arctangent provides values in the range $[-\pi, \pi)$. Therefore, included in the knowledge of the principal phase spectrum values, are the signs of the real and imaginary components. Van Hove et al. [139] show that the magnitude spectrum, along with this sign information, provides a unique specification of a finite duration causal sequence. The 'signed-magnitude' is defined as,

$$A(\omega : \omega_o) = \begin{cases} |X(\omega)| & \text{if } -\omega_\mathrm{o} \leq \mathrm{ARG[X}(\omega)] < \omega_\mathrm{o} + \pi, \\ -|X(\omega)| & \text{otherwise} \end{cases} \tag{6.4}$$

where ω_o is an arbitrary number within the interval $[-\pi, \omega_o + \pi)$. Thus, $A(\omega : \omega_o)$ contains information about both the magnitude spectrum and the sign of the real and imaginary parts of the Fourier transform. Their theorem is stated as follows:

Theorem 2 *Let $x(n)$ and $y(n)$ be two real, causal, and finite extent sequences with z-transforms which have no zeros on the unit circle. If $A_x(\omega : \omega_o) = A_y(\omega : \omega_o)$ for all ω then $x(n) = y(n)$ [139].*

In terms of the iterative reconstruction algorithm, $A(\omega : \omega_o)$ imposes both a magnitude spectrum and phase spectrum constraint. When both of these constraints are enforced, the algorithm converges to the original signal (Fig. 6.2(f)&(g)). In our experiments, we use $\omega_o = \pi/2$.

The phase spectrum constraint amounts to splitting the phase spectrum in half (at an arbitrary point), then taking note in which half the phase spectrum values lie for each frequency. In every iteration of the reconstruction algorithm, each phase spectrum value is constrained to vary only within the half from which the original phase spectrum value came. This is enforced by adding π to any phase values that are not in the correct half.

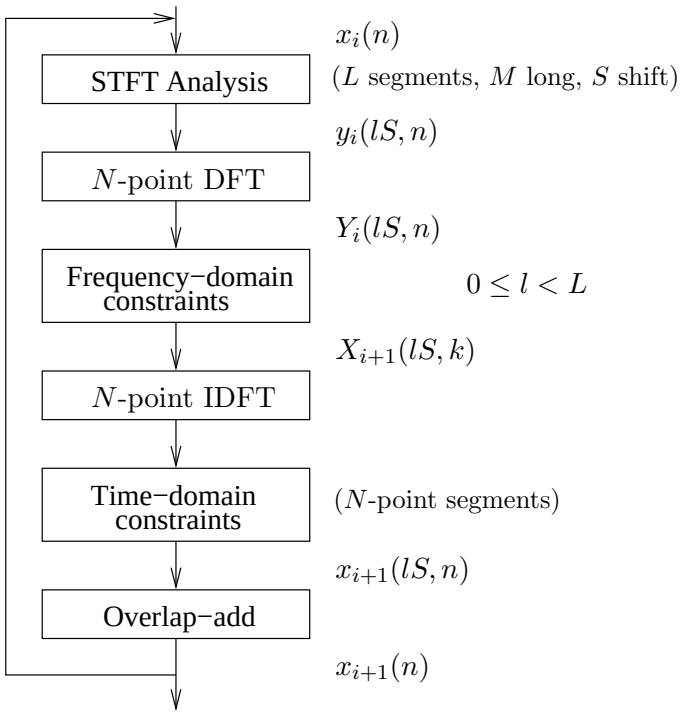


Fig. 6.3: STFT-based iterative reconstruction framework.

6.1.2 Reconstruction within the STFT Framework

Theorems 1 and 2 are also applicable in the context of the STFT [150]. The STFT overlap-add analysis imposes additional restrictions on magnitude-phase pairings. Specifically, adjacent short-time sections must be consistent in their region of overlap. Thus, when reconstructing from partial information, extra information is present in the overlapping sections.

There are two ways to reconstruct via the STFT. One method is referred to as 'sequential extrapolation', where the short-time sections are reconstructed in the order determined by their positions on the time axis. Each section is determined by its known spectral information as well as the known samples in the region of overlap with previous sections. This method is investigated by Nawab et al. [90]. The framework for the method we use is illustrated in Fig. 6.3. This method was employed by Griffin and Lim [47] for time-scale modification of speech. It is referred to as 'simultaneous extrapolation'. In this method, the known spectral information of all short-time sec-

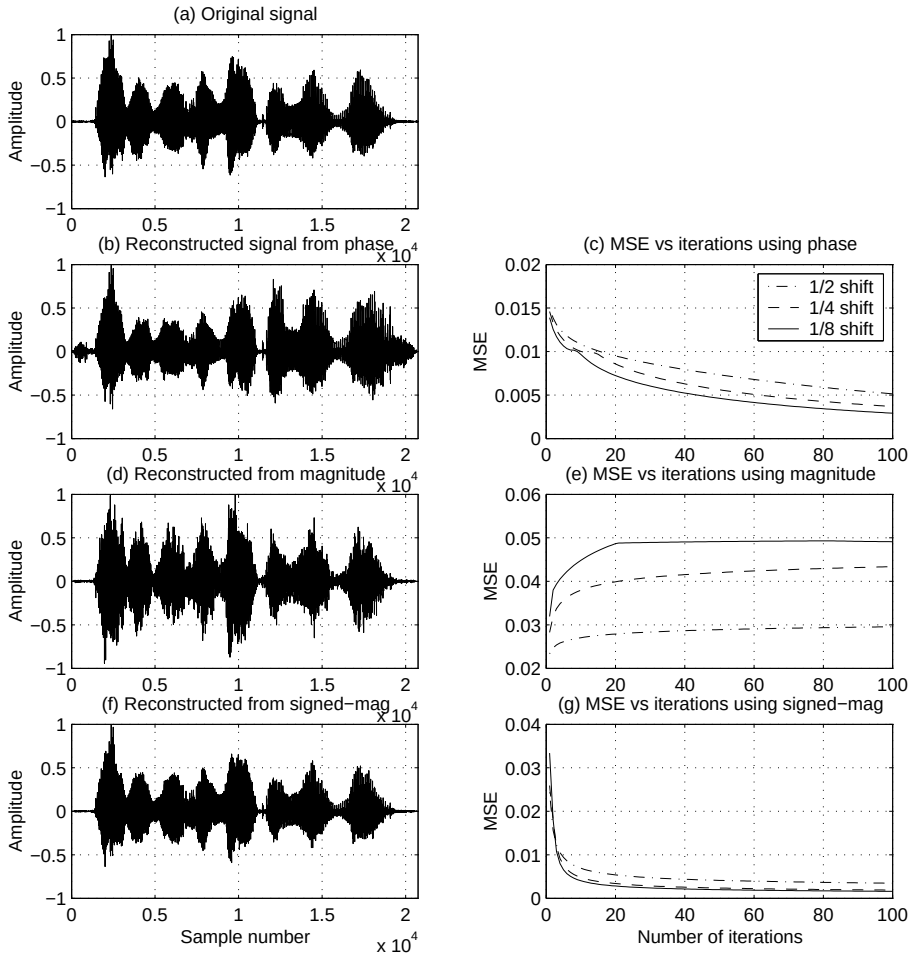


Fig. 6.4: Results of experiments in Section 6.1.2. (a) is the original signal used for experiments in Sections 6.1.2 and 6.2, "Why were you away a year Roy?". (b), (d) and (f) show reconstructed signals after 100 iterations, using a frame-shift of $\frac{1}{8}$ and a rectangular analysis window of duration 32 ms (these signals are scaled to vary over the same range as the original signal). (c), (e) and (g) show plots of the respective MSE values at every iteration.

tions are used simultaneously to determine the unknown signal (i.e., the whole signal is analysed and synthesised in every iteration). In the experiments that follow, we use a rectangular analysis window of duration 32 ms. Any comments made with respect to signal intelligibility are based on informal listening tests by the author.

6.1.2.1 Reconstruction from Short-time Phase Spectra

We analyse the signal in Fig. 6.4(a) at various segment shifts ($\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$), keeping only the phase spectra from each segment. The known phase spectra samples are enforced

for every iteration of the reconstruction algorithm. The magnitude spectrum is initially set to unity for all frequencies. For all segment shifts, the algorithm converges toward a scaled version of the original signal (Fig. 6.4(b)&(c)). The iterative application of overlap and addition ensures that adjacent short-time sections are consistent in their regions of overlap. The greater these regions of overlap, the faster the convergence due to the fact that more initial constraints are imposed on the final solution. This imposition of more constraints also leads to lower MSE. The associated spectrogram is given in Fig. 6.5(b). It looks identical to the spectrogram of the original signal in Fig. 6.5(a).

Note that, for the case of reconstruction from short-time phase spectra, there must be at least one sample of overlap between segments. The overlapping sample(s) serve to maintain the energy relationship between adjacent segments. So, even although no energy information is provided to seed the iterative algorithm, the energy contour (albeit scaled) is preserved.

6.1.2.2 Reconstruction from Short-time Magnitude Spectra

Griffin and Lim first used the STFT reconstruction framework of Fig. 6.3 to reconstruct time-scaled versions of a signal from short-time magnitude spectra [47]. Here, we analyse the algorithm for no time-scaling, imposing the known short-time magnitude spectra in each iteration. The short-time phase spectrum for each segment is initially randomised. The algorithm does not converge toward the original speech (Fig. 6.4(d)&(e)). Informal listening tests, however, indicate that more overlap between frames (i.e., less shift) leads to the reconstructed speech sounding more like the original. This is expected, since more overlap imposes more restrictions on the form of the final solution. The spectrogram of the reconstructed signal is given in Fig. 6.5(c).

6.1.2.3 Reconstruction from Short-time Signed-Magnitude Spectra

Here, we enforce the known short-time magnitude spectra in addition to the phase spectra sign information (see Section 6.1.1.3). Once again, we observe the signal converging, with the rate of convergence increasing, and the error reducing, with more overlap (Fig. 6.4(f)&(g)). The spectrogram of the reconstructed signal in Fig. 6.5(d) looks identical

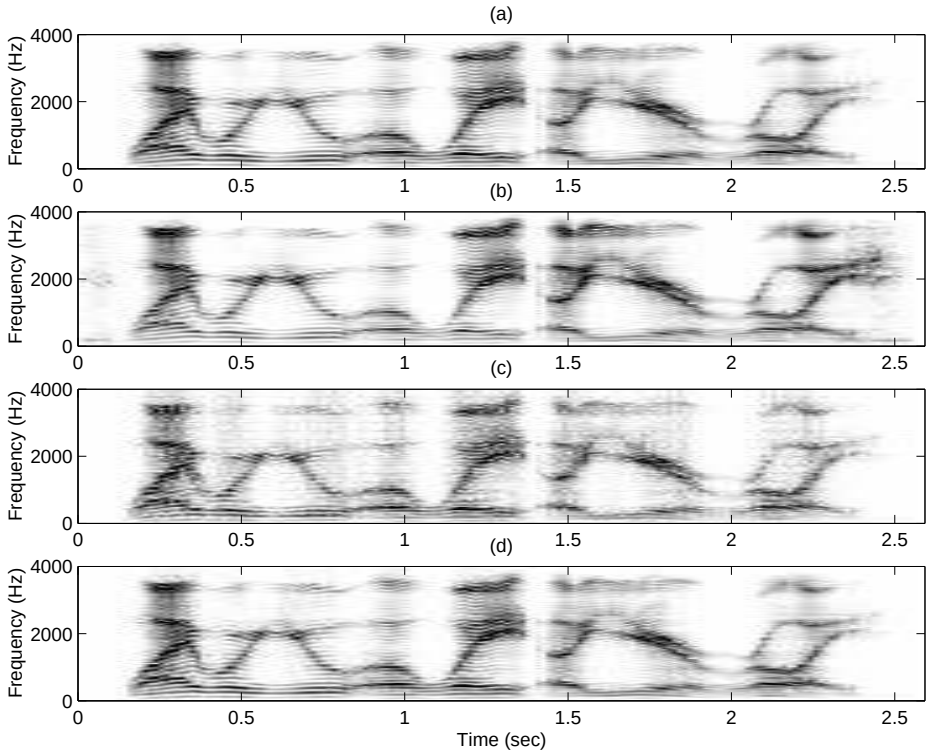


Fig. 6.5: Spectrograms of (a) original signal "Why were you away a year Roy?", (b) iterative reconstruction from short-time phase spectra, (c) iterative reconstruction from short-time magnitude spectra (d) iterative reconstruction from short-time signed-magnitude spectra. The signals in (b), (c) and (d) are reconstructed using 100 iterations, a frame-shift of $\frac{1}{8}$ and a rectangular analysis window of duration 32 ms.

to that of the original.

6.2 Reconstruction from Partial STFT Phase Spectra

In light of results from the previous section (which are well-established), we note that knowledge of the phase spectrum is enough for unique signal reconstruction (to within a scale factor), while the same is not true for magnitude spectrum — the magnitude spectrum must be accompanied by phase spectrum sign information for unique reconstruction. We find it interesting that the phase spectrum can be used to reconstruct a signal, while it is useless for extracting ASR features. If so much information is contained in the phase spectrum, then it may be possible to capture and use it to improve the performance of ASR systems. However, using the phase spectrum directly for ASR

has proven difficult due to phase-wrapping and other problems [31, 86, 145]. If there is to be any chance of using the phase spectrum for ASR, then it will have to be via an alternative representation.

To this end, we further analyse the use of the phase spectrum for signal reconstruction. Specifically, we explore the use of partial phase spectrum information (in the absence of all magnitude spectrum information) for intelligible signal reconstruction[7].We employ the STFT-based reconstruction framework in Fig. 6.3. The partial phase spectrum representations that we investigate are the phase spectrum sign information, GDF and the IFD. The justification for this is as follows: if the use of either the phase spectrum sign, GDF, or IFD information results in intelligible signal reconstruction, this would advocate the possible use of the partial information as a basis for an ASR feature set.

6.2.1 Reconstruction from Short-time Phase Spectra Sign

A similar experiment to that in Section 6.1.2.3 is performed. Rather than enforcing sign and magnitude spectra values, we only enforce the sign constraint in every iteration. All magnitude spectrum values are initially set to unity and phase spectrum values are randomised (such that the initial values are in the correct region, as determined by the sign information). At first glance, it appears that the MSE does not increase in each iteration (Fig. 6.6(a)&(b)). However, closer inspection reveals that the MSE increases slightly at some points along the curve. More overlap seems to result in a better estimation of the original signal. Informal listening tests indicate that more overlap also leads to better intelligibility. It is interesting that only a small amount of phase spectra information provides for an intelligible signal (although the reconstructed signal is noisy). The increased overlap accommodates, to some extent, for the sparse phase spectral information. The spectrogram of the reconstructed signal is given in Fig.

[7]This work is different to the partial phase spectrum experiments conducted by Yegnanarayana and his colleagues [150]. They use the word 'partial' to denote the situation where the required number of phase spectrum samples for unique signal reconstruction are not known. In order to compensate for the unknown phase spectrum samples, they enforce some known signal samples or magnitude spectrum samples during the iterative reconstruction procedure. In our work, the word 'partial' is used to mean something different. Continue to read Section 6.2 for an explanation.

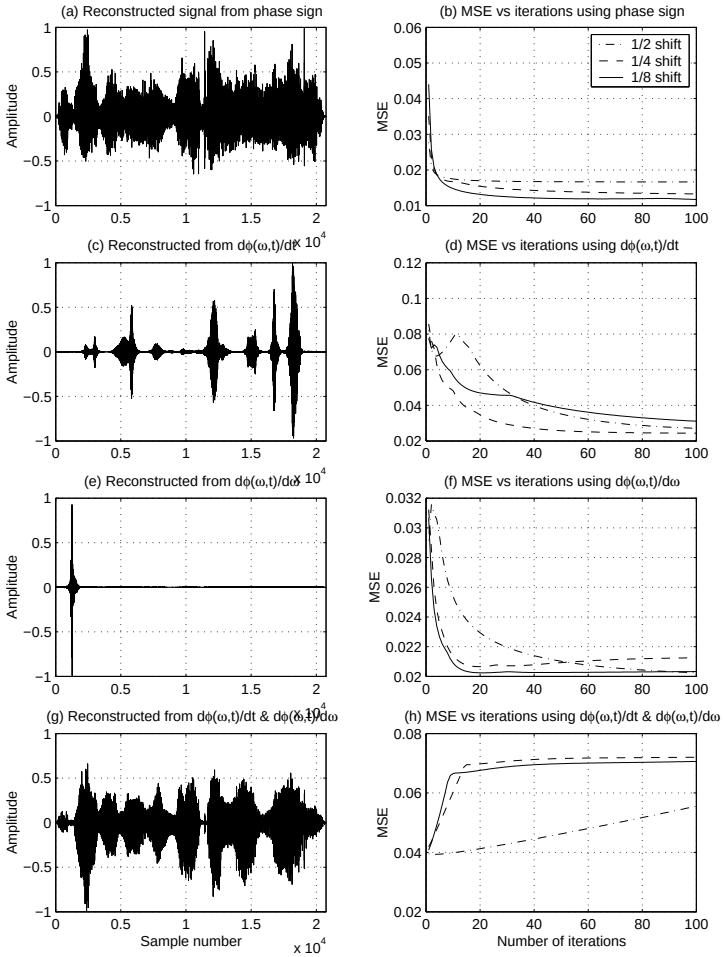


Fig. 6.6: Results of experiments in Section 6.2. (a), (c), (e) and (g) show reconstructed signals after 100 iterations, using a frame-shift of $\frac{1}{8}$ and a rectangular analysis window of duration 32 ms (these signals are scaled to vary over the same range as the original signal). (b), (d), (f) and (h) show plots of the respective MSE values at every iteration.

6.7(a).

6.2.2 Reconstruction from Time and Frequency Derivatives of Short-time Phase Spectra

We take the phase spectrum from each short-time section and randomise it across frequency, such that the IFD of each segment is preserved. In other words, add the same random sequence (across frequency) to the phase spectrum values of each frame. For example, consider a frame of length M and a DFT length of $N = 2M$. Add a random sequence to the phase values in the first $M+1$ DFT bins (i.e., bin numbers 0 to M). To

determine the remaining $M-1$ phase values (i.e., bin numbers $M+1$ to $N-1$), take the new phase values from bins 1 to $M-1$ then reverse the sign and reverse the order of the numbers. That is, given the new phase values for the first $M+1$ bins, calculate the remaining bin phase values by $\psi(k) = -\psi(N-k)$, where $k = M+1, M+2, \ldots, N-1$ is the bin number. The resulting phase spectra are used in place of the original phase spectra in the reconstruction algorithm (and magnitude spectra are set to unity). The algorithm does not converge toward the original signal, nor does it provide an intelligible signal (Fig. 6.6(c) & 6.7(b)).

In a similar vein, we take the original phase spectra and randomise them across time, such that the GDF of each segment is preserved. That is, generate a random sequence whose length is equal to the number of frames in the utterance, then add this same sequence to the time-trajectory of the phase spectrum values for each DFT bin. Remember to do this for the phase values in the first $M+1$ DFT bins (for each frame), then calculate the remaining bin phase values as described above. Reconstruction is performed with the resulting phase spectra (and magnitude spectra are set to unity). Again, the reconstruction algorithm does not converge to an intelligible solution (Fig. 6.6(e)&6.7(c)).

Therefore, reconstruction of intelligible speech is not possible from either knowledge of only the IFD or the GDF. This holds true, regardless of the amount of overlap. Figures 6.6(d) and 6.6(f) seem to indicate convergence. This is deceiving. In fact, the MSE is converging toward the original signal mean squared amplitude (0.0194), since the algorithm (in both cases) provides a signal whose energy tends to diminish with each iteration.

We now attempt to reconstruct the signal from the knowledge of both IFD and GDF. In order to do this, we must first reconstruct the phase spectra from these known quantities. Notice that the first-segment phase spectrum can only be reconstructed to within a time-shift of the original first-segment phase spectrum, since all we know about it is the GDF. The remaining segments are reconstructed in relation to this segment. To reconstruct the phase spectrum values from the GDF and IFD we do the following: The phase value for DFT bin number 0 is set to zero in every frame. The

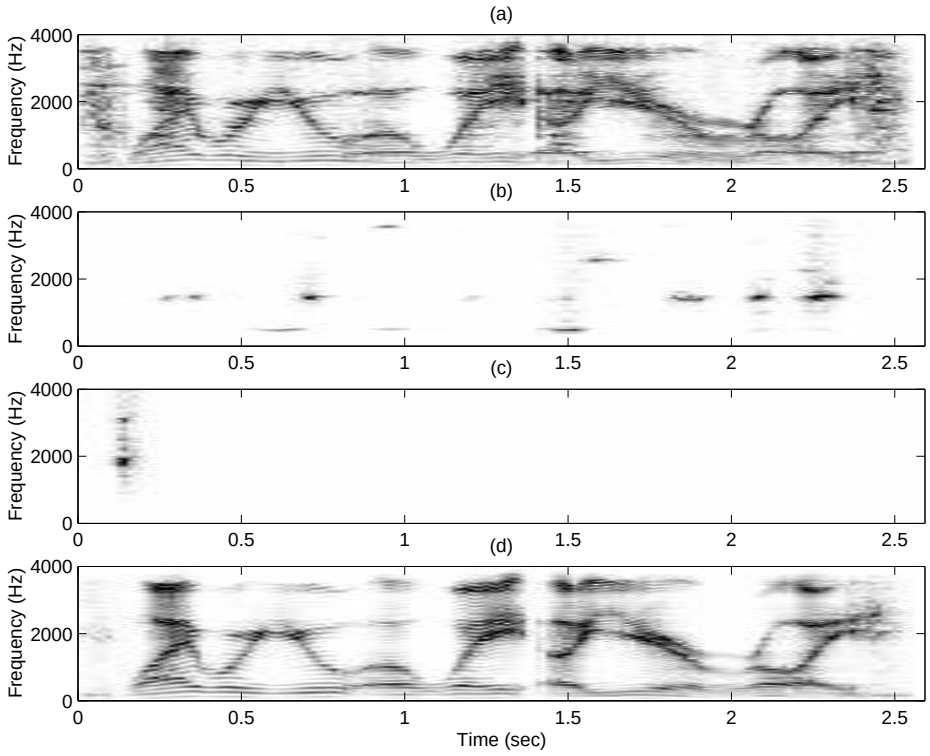


Fig. 6.7: Spectrograms of reconstructed signals. Signals are reconstructed from knowledge of (a) short-time phase spectrum sign information, (b) IFD, (c) GDF, and (d) both time and frequency derivatives of the short-time phase spectra. All signals are reconstructed using 100 iterations, a frame-shift of $\frac{1}{8}$ and a rectangular analysis window of duration 32 ms. The spectrogram of the original signal is given in Fig. 6.5(a).

remaining phase values (for each frame) are calculated by cumulatively summing the GDF across DFT bins 1 to M. We then shift all of these phase values by a constant in each frame (dependent on the frame), so that the phase changes over time for one particular DFT bin (this can be any bin, the decision is arbitrary) are the same as in the original signal (i.e., we use the IFD values for only one bin). The phase values for bins $M+1$ to $N-1$ are calculated as previously described[8]. Since the original phase spectra values cannot be recovered[9], the algorithm does not converge (Fig. 6.6(h)). Regardless of this, a solution that sounds almost exactly like the original speech is

[8]Note that this is only one way of reconstructing the phase spectrum values. It is also possible to reconstruct by using the GDF values for only one frame then to extrapolate the phase values for the other frames by using the IFD values for all DFT bins.

[9]The raw phase spectrum values are only meaningful in the context of a fixed-time reference. All that we have lost in this reconstructed signal is the original fixed-time reference. Time referencing is now in relation to the phase spectrum values of the first frame (i.e., we still have a time reference, but it is different to that of the original phase spectra values).

provided (Fig. 6.6(g)). The reconstructed signal is similar to the original signal (Fig. 6.4(a)) in many respects, apart from the fact that it looks upside-down (which has no effect on intelligibility). The spectrogram in Fig. 6.7(d) is almost identical to that of the original in Fig. 6.5(a). Therefore, in the context of the STFT reconstruction framework, when both the IFD and GDF are preserved, adequate information is available for intelligible signal reconstruction.

6.3 Conclusion

In this chapter, we provided a tutorial on the topic of iterative, one dimensional, signal reconstruction (specifically speech signals) from the magnitude spectrum and the phase spectrum. While this topic has been extensively researched and documented, our intention was to recast some well-established results for the benefit of new researchers and those who desire a short, yet comprehensive, review of the subject. The three main points of the tutorial are: (i) a signal can be reconstructed to within a scale factor from its phase spectrum, (ii) a signal cannot be reconstructed to within a scale factor from its magnitude spectrum, and (iii) a signal can be reconstructed to within a scale factor from its magnitude spectrum when the phase-sign (i.e., one bit of phase information) is known. Through a number of illustrate examples, we first demonstrated how the algorithms work when the spectral information is determined over the entire duration of the signal. We then demonstrated that the algorithms are equally valid for reconstruction of a signal from the spectra obtained from short-time segments. In addition, we presented the results of some further experimentation in which we have attempted to reconstruct a speech signal from only partial phase spectrum information (in the absence of all magnitude spectrum information). We make the following observations: (i) intelligible signal reconstruction (albeit noisy) is possible from knowledge of only the phase spectrum sign information, (ii) an intelligible signal cannot be reconstructed from knowledge of only the phase spectrum frequency-derivative or only the phase spectrum time-derivative, and (iii) an intelligible signal can be reconstructed from the combined knowledge of both the phase spectrum frequency-derivative and time-derivative.

Chapter 7

Evaluation of Modified Group Delay Features on Several ASR Tasks

The human perception experiments in Chapter 4 were performed with the aid of phase-only stimuli which were made by analysing a speech signal with an STFT, setting the magnitude spectra of each short-time segment to unity, then performing an inverse Fourier transform on each segment and reconstructing with the OLA method. Since an arbitrary change to the STFT is not necessarily a valid STFT (see Section 3.3), the resulting STFT of the phase-only stimuli does not actually have unity magnitude spectra. As previously explained, when the segments are overlapped and added during synthesis, their magnitude spectra deviate from unity because the samples in the overlapping regions between the segments are no longer consistent. With this in mind, it may seem that the intelligibility of the phase-only stimuli comes from the resulting magnitude spectra. If this is so, then an ASR system with an MFCC front-end should provide good word accuracy for the phase-only stimuli; however, we observed Chapter 5 that the results are rather poor. This leads us to believe that there is discriminating information in the phase spectrum part of the speech signal that is not being captured by the MFCC representation, providing motivation to investigate the use of the phase spectrum to

derive features for ASR.

In its raw form, the phase spectrum is not amiable to ASR processing. Unlike the magnitude spectrum, the phase spectrum does not explicitly exhibit the system resonances. A physical connection between the phase spectrum and the structure of the vocal apparatus is not apparent. It is therefore preferrable that the phase spectrum be transformed into a more physically meaningful representation. If such a representation can be found, we need to determine if it can be used to improve ASR recognition performance. The phase spectrum has two independent variables: frequency and time. Thus, two phase spectrum representations that can be explored are the frequency-derivative (group delay function, GDF) and the time-derivative (instantaneous frequency distribution, IFD).

The focus of this section is on the use of the GDF for ASR. Murthy and Gadde [88] have recently proposed a feature set, called MODGDF, that is derived from a modified GDF. These features are perhaps the most concerted effort into phase spectral features thus far. We conduct experiments (independent of the authors in [88]) to determine if their proposed features provide an improvement over the popular MFCC representation on several ASR tasks. Our results indicate that, in isolation, the MODGDF features provide no improvement over MFCCs. In some cases, the concatenation of the MODGDF features to the MFCCs provide for a performance improvement; however, there is no consistency in the results. MFCCs seem to provide the best overall performance.

The outline of this chapter is as follows: In Section 7.1, we review the GDF and highlight the problems when using it directly for ASR. We demonstrate, with some simple examples, the volatility of the GDF to noise, pitch epochs and windowing effects. In Section 7.2, we summarise the work by Yegnanarayana and Murthy [145] on the modified GDF (MGDF), which serves to remedy the problems of the GDF. In Section 7.3, we provide the implementation details of Murthy and Gadde's MGDF-based features [88] (MODGDF). In Section 7.4, we test the MODGDF features on several ASR tasks (ISOLET, Aurora II and Resource Management) in additive white and coloured noises.

Publications from this research: [11].

7.1 Group Delay Function

As discussed in Section 3.5.1, the GDF, $\tau(\omega)$, is defined as the negative derivative of the phase spectrum with respect to ω [99]:

$$\tau(\omega) = -\frac{d}{d\omega}\arg[X(\omega)] \quad = \quad -\frac{X_R(\omega)X'_I(\omega) - X_I(\omega)X'_R(\omega)}{|X(\omega)|^2}, \tag{7.1}$$

where the time dependency has been dropped (since in this analysis we only consider one short-time segment).

A theoretical analysis of the volatility of the GDF to the effects of noise, pitch epochs and windowing (or truncation) has been provided by other authors [88, 145]. Our intention in this section is not to repeat this theory, but rather to convey our own practical understanding of the GDF through a comprehensive set of simple examples.

For the following illustrations, we employ an autoregressive system:

$$X(z) = \frac{1}{1 + \sum_{i=1}^{4} a_i z^{-i}}, \tag{7.2}$$

with coefficient values: $a_1 = -2.760, a_2 = 3.809, a_3 = -2.654$ and $a_4 = 0.924$ (these values are the same as those used in [145]). We compute the system impulse response and retain a sufficient number of its initial samples such that the impulse response has fully decayed (Fig. 7.1(a)). This version of the truncated impulse response is, for all intents and purposes, representative of the complete impulse response. The power spectrum of this signal, shown in Fig. 7.1(f), exhibits two resonances. The GDF, shown in Fig. 7.1(k), also clearly conveys the two resonances[1]. The zeros of this signal are shown in Fig. 7.1(p).

Fig. 7.1(b) shows the impulse response with additive white noise, such that the signal-to-noise ratio (SNR) is 40 dB. The resonance peaks are still clearly discernible in the associated power spectrum of Fig. 7.1(g). The resonances, previously conveyed

[1]Note that the autoregressive system in Eq. 7.2 is a minimum-phase system. A property of a minimum-phase system is that the principal phase spectrum, ARG$[X(\omega)]$, is equal to the continuous phase spectrum, $\arg[X(\omega)]$. In other words, there are no discontinuities in the principal phase spectrum. This explains why the GDF in Fig. 7.1(k) is smooth.

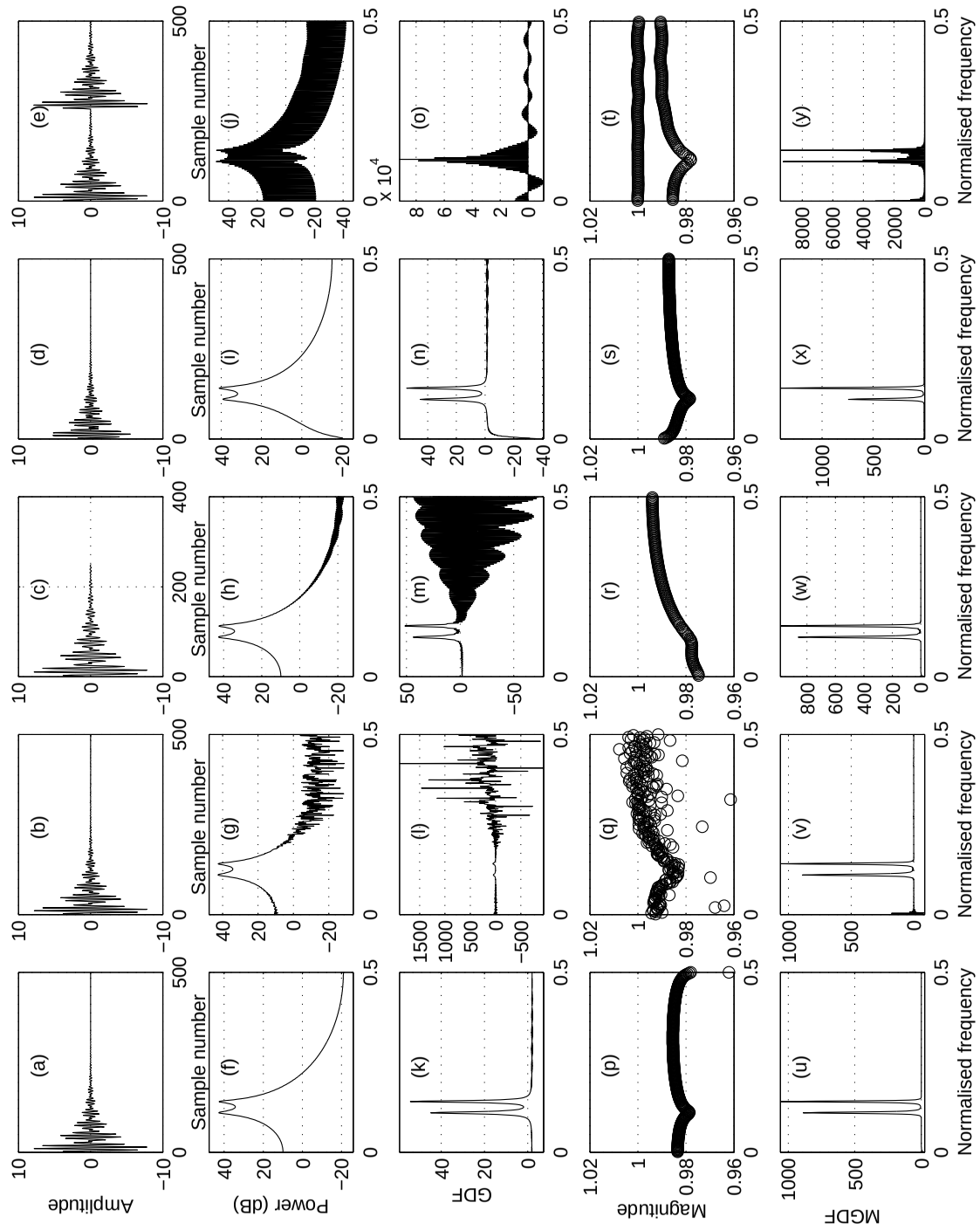


Fig. 7.1: The first row shows the (a) impulse response of an autoregressive process, (b) impulse response of the same autoregressive process in white noise (SNR=40dB), (c) truncated (or windowed) impulse response, (d) pre-emphasised impulse response (coeff. 0.97), and (e) the system response with an excitation of two impulses. The corresponding power spectra are shown in (f)–(j). The GDFs are shown in (k)–(o). The zero distributions are shown in (p)–(t). The MGDFs ($\alpha = 1$, $\gamma = 1$, and $s_w = 6$) are shown in (u)–(y). A rectangular analysis window is used in all cases.

by the GDF in Fig. 7.1(k), are non-existent in the GDF for the noisy signal (Fig. 7.1(l)). The additive white noise introduces zeros close to the unit circle (Fig. 7.1(q)) which results in very small power spectral values at the frequency locations of these zeros. These small values of the power spectrum, $|X(\omega)|^2$, in the denominator of Eq. 7.1, subsequently result in large GDF values. Another way to explain this is that the impulse response is no longer seen to be produced by a minimum-phase system, since the additive white noise results in a response that does not decay. The principal phase spectrum contains discontinuities which subsequently show up as spikes in the GDF.

Now consider the same impulse response, but this time we only have half the number of samples, such that the full decay of the impulse response is not captured (Fig. 7.1(c)). The windowing results in zeros being closer to the unit circle (Fig. 7.1(r)). Thus, a severe amount of distortion is introduced into the GDF (Fig. 7.1(m)). Note that windowing does not distort the power spectrum (Fig. 7.1(h)) as much as the GDF. In fact, by applying different window types (e.g., Hamming, Hanning, Gaussian, Blackman), the distortion can be reduced somewhat, in exchange for diminished resolving capability. The choice of window has a large impact on the resulting GDF [20, 21].

Fig. 7.1(d) presents a pre-emphasised impulse response (coeff. 0.97). The power spectrum for this signal is shown in Fig. 7.1(i). The effect of pre-emphasis on the GDF is shown in Fig. 7.1(n). Pre-emphasis introduces a zero near the unit circle causing a negative peak at $\omega = 0$. The associated zero distribution is shown in Fig. 7.1(s).

Considering that speech can be approximately modeled as the output of an autoregressive system excited by a periodic train of impulses, we now examine a signal obtained by exciting the autoregressive system with two impulses (Fig. 7.1(e)). Although harmonics of F0 locally dominate the power spectrum, the resonance peaks are still globally discernible (Fig. 7.1(j)). However, no such peaks are visible in the GDF (Fig. 7.1(o)). This is due to the fact that the excitation introduces zeros extremely close to, if not on, the unit circle (Fig. 7.1(t)).

These simple examples demonstrate the volatility of the GDF to noise, pitch epochs and windowing effects. In all cases, it is the presence of zeros close to the unit circle that corrupt the GDF. Therefore, the GDF (given by Eq. 7.1) needs modification for it

to be useful in ASR feature extraction.

7.2 Modified Group Delay Function

If we assume that speech is produced by a source-system model, the speech power spectrum, $|X(\omega)|^2$, can be expressed as the multiplication of the system component of the power spectrum, $S(\omega)^2$, with the source (or excitation) component of the power spectrum, $E(\omega)^2$:

$$|X(\omega)|^2 = S(\omega)^2 E(\omega)^2. \tag{7.3}$$

As demonstrated in the previous section, the excitation contributes zeros near the unit circle which cause meaningless peaks in the GDF. The modified group delay function (MGDF), $\tilde{\tau}(\omega)$, proposed by Yegnanarayana and Murthy [145], is formed by multiplying the GDF by the source component of the power spectrum:

$$\tilde{\tau}(\omega) \quad = \quad \tau(\omega) E(\omega)^2. \tag{7.4}$$

This operation gives less weight to peaks in the GDF which are the result of excitation-induced zeros near the unit circle. This is equivalent to replacing the denominator in Eq. 7.1 with the system component of the power spectrum, $S(\omega)^2$:

$$\tilde{\tau}(\omega) \quad = \quad \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{S(\omega)^2}. \tag{7.5}$$

$S(\omega)^2$ is obtained by cepstral smoothing of $|X(\omega)|^2$. In practice, the cepstral smoothing operation not only smooths out zeros introduced by excitation, but also those contributed by noise and windowing. In fact, the cepstral smoothing removes the effect of any zeros that are close to the unit circle.

Murthy and Gadde [88] recently expanded on this expression, proposing the addition of two variables, γ and α. The role of γ is to vary the contribution from the system

component of the power spectrum, as follows:

$$\tilde{\tau}_{\gamma}(\omega) \quad = \quad \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{S(\omega)^{2\gamma}}. \tag{7.6}$$

The second additional variable, α, is a compression factor, such that the final expression for the MGDF is:

$$\tilde{\tau}_{\alpha,\gamma}(\omega) \quad = \quad \frac{\tilde{\tau}_{\gamma}(\omega)}{|\tilde{\tau}_{\gamma}(\omega)|}|\tilde{\tau}_{\gamma}(\omega)|^{\alpha}. \tag{7.7}$$

This parameter does not add any information, but rather represents the information already present in a more favourable form for ASR (as does the application of logarithmic compression for filter-bank energy coefficients). Please refer to [88] for a detailed discussion of these additional variables.

The bottom row of Fig. 7.1 shows the MGDFs for each of the cases previously examined in Section 7.1 (a cepstral smoothing window of size $s_w = 6$ is used, with $\alpha = 1$ and $\gamma = 1$). In each case, the resonance peaks are now clearly discernible in the MGDF.

7.3 Computation of Features

This section details the computation of the features used for the ASR experiments described in Section 7.4. In all cases, speech is pre-emphasised before analysis (pre-emphasis coefficient of 0.97) and a Hamming analysis window of duration 25 ms is used, with a 10 ms frame-shift[2].

As a baseline for recognition performance, we test with MFCCs. These are derived from the magnitude spectrum. For each frame:

1. Compute the discrete Fourier transform (DFT) of $x(n)$, denoted by $X(k)$.
2. Compute the power spectrum $|X(k)|^2$.
3. Apply a Mel-warped filter-bank (0 – 4 kHz) to $|X(k)|^2$ to obtain 24 filter-bank energies (FBEs).

[2]Note that the MFCC analysis window duration was 20 ms for the experiments in Section 5.1. Therefore, there may be some slight differences in the word accuracy results.

4. Compute the discrete cosine transform (DCT) of the log FBEs.

5. Keep 12 cepstral coefficients, not including $c(0)$ (i.e., keep $c(n)$ for $n = 1, 2, \ldots, 12$).

The MODGDF features are computed as follows. For each frame:

1. Compute the DFT of $x(n)$ and $nx(n)$, denoted by $X(k)$ and $Y(k)$ respectively.

2. Compute the cepstrally smoothed spectrum of $|X(k)|$, denoted by $S(k)$. To do this: i) calculate the IDFT of $|X(k)|$, ii) keep the first s_w samples and set the remaining samples to zero, iii) calculate the DFT of this modified vector, iv) the magnitude spectrum of this DFT is the smoothed spectrum, $S(k)$.

3. Compute the modified GDF, $\tilde{\tau}_{\alpha,\gamma}(k)$, as:

$$\tilde{\tau}_{\alpha,\gamma}(k) \quad = \quad sign. \left| \frac{X_R(k)Y_R(k) + X_I(k)Y_I(k)}{S(k)^{2\gamma}} \right|^{\alpha} \tag{7.8}$$

   where $sign$ is the sign of $\frac{X_R(k)Y_R(k)+X_I(k)Y_I(k)}{S(k)^{2\gamma}}$.

4. Compute the DCT of $\tilde{\tau}_{\alpha,\gamma}(k)$.

5. Keep 12 cepstral coefficients, which includes $c(0)$ (i.e., keep $c(n)$ for $n = 0, 1, \ldots, 11$).

Results of experiments by Murthy and Gadde [88] indicate that a cepstral smoothing window of size $s_w = 6$ is best for smoothing. They also recommend that $\alpha = 0.3$ and $\gamma = 0.9$. Best recognition performance is obtained when keeping $c(0)$. In addition, their results show that CMS improves performance. Therefore, we employ all of these settings[3]. CMS is performed on both the MFCCs and the MODGDFs (except for the ISOLET task).

7.4 Experiments

We use a number of ASR tasks to compare the performance of the MFCC and MODGDF features. For consistency, all databases contain data sampled at the same rate of 8 kHz.

[3]Note that Murthy and Gadde [88] weight the MODGDF cepstral values, $c(n)$, by n for $n > 0$; this is liftering and makes no difference to recognition performance in a HMM framework [105]. This is because Gaussian mixture modeling (GMM) is invariant to scaling — scaling of data leads to a corresponding scaling in GMM parameters. Therefore, we do not perform liftering.

Table 7.1: ISOLET word recognition scores: white noise

Case	Feature type (E–energy, D–delta , A–acceleration.)	SNR (dB)				
		∞	30	20	15	10
I-W1	MFCC	78.27	74.87	67.44	56.67	37.50
I-W2	MODGDF	76.79	63.01	32.82	16.22	7.37
I-W3	MFCC+MODGDF	78.59	69.55	51.67	33.33	18.65
I-W4	MFCC+E	79.62	76.73	66.03	52.50	36.41
I-W5	MODGDF+E	78.65	65.51	37.82	20.58	8.65
I-W6	MFCC+MODGDF+E	79.42	70.45	51.86	33.59	19.74
I-W7	MFCC+E+D+A	90.83	**89.04**	**79.36**	**68.91**	**52.95**
I-W8	MODGDF+E+D+A	89.29	82.37	70.58	56.79	35.32
I-W9	MFCC+MODGDF+E+D+A	91.41	85.19	75.06	64.49	46.86
I-W10	ISOLET-tuned	**92.31**	85.64	76.79	66.41	46.22

Table 7.2: ISOLET word recognition scores: averaged over several types of coloured noise

Case	Feature type (E–energy, D–delta , A–acceleration.)	SNR (dB)				
		∞	30	20	15	10
I-C1	MFCC	78.27	76.57	68.26	57.84	42.02
I-C2	MODGDF	76.79	74.17	62.81	45.34	24.31
I-C3	MFCC+MODGDF	78.59	67.80	54.33	32.77	16.44
I-C4	MFCC+E	79.62	71.20	60.00	42.87	25.37
I-C5	MODGDF+E	78.65	65.48	48.46	25.39	11.78
I-C6	MFCC+MODGDF+E	79.42	69.30	54.52	32.73	15.04
I-C7	MFCC+E+D+A	90.83	**86.17**	**79.97**	**67.60**	**49.62**
I-C8	MODGDF+E+D+A	89.29	82.13	72.53	56.22	33.30
I-C9	MFCC+MODGDF+E+D+A	91.41	85.14	78.61	65.24	43.19
I-C10	ISOLET-tuned	**92.31**	85.96	79.91	66.89	43.86

We use HTK [151] to train and test the HMMs.

7.4.1 Isolated Word Task

ISOLET is an isolated-word, speaker-independent task, with speech sampled at 8 kHz. The vocabulary consists of 26 English letters. Two repetitions of each letter are recorded for each speaker. Speakers are divided into two sets: 90 for training, 30 for testing. Each word is modeled by a HMM with 5 emitting states and 5 Gaussian mixtures per state. HMMs are trained on clean data and tested on data with white noise or coloured noise added at several SNRs. The grammar is such that the likelihood of each word is the same. There is no need to set the word insertion probability since only one word can

Table 7.3: Combinations used for ISOLET tuning.

Parameter	Range	Step
α	0.1–0.5	0.1
γ	0.5–1.0	0.1
s_w	4–10	2

occur per utterance. We do not use CMS here because the ISOLET utterances are too short[4].

Word recognition scores for additive white noise are provided in Table 7.1. Bold font denotes the best word recognition score for each SNR. We observe that MODGDFs perform worse than MFCCs in all SNRs (cases I-W1 and I-W2). The same is true when energy and deltas are attached (cases I-W4, I-W5, I-W7, and I-W8). When the MODGDFs are concatenated with the MFCCs (case I-W3), a slight performance improvement over using MFCCs alone is observed in matched conditions (SNR=∞); however, this improvement in matched conditions is at the expense of a reduced performance in unmatched conditions. This is also the case when deltas are attached (compare case I-W7 to I-W9).

Murthy and Gadde performed a line search on the SPINE database to determine the best values for α, γ, and s_w [88]. Using the same feature size and configuration as for case I-W9, we perform a line search on ISOLET (see Table 7.3). Optimal values for the MODGDF feature, such that the matched condition score for MFCC+MODGDF+E+D+A is maximised, were found to be $\alpha = 0.3$, $\gamma = 0.9$, and $s_w = 8$. We refer to these features as the 'ISOLET-tuned' case. Note that we do not determine l_w (see [88] for definition) since we are ignoring channel effects. The matched recognition score improves slightly (case I-W10), but again at the expense of reduced performance in lower SNRs[5].

Word recognition scores, averaged over four types of coloured noise[6] (subway, babble,

[4]Empirical evidence suggests that, for increased recognition performance from CMS, utterances must be longer than 2-4 seconds [62].

[5]While tuning does provide slightly improved recognition scores, we found that over the variable ranges tested in Table 7.3, the recognition score did not change that much to warrant the large computational overhead required by the tuning. Therefore, we use the same parameter values as in [88] (i.e., $\alpha = 0.3$, $\gamma = 0.9$ and $s_w = 0.6$) for all other experiments. In fact, one could argue that if we must tune the MODGDF features for each database, then MFCCs should also be tuned (e.g., the number of FBEs and the type of warping).

[6]ISOLET word recognition scores for each noise type are provided in Appendix C.

Table 7.4: Aurora II word accuracy scores: averaged over several types of coloured noise

Case	Feature type (E–energy, D–delta , A–acceleration.)	SNR (dB)				
		∞	20	15	10	5
A-C1	MFCC	97.05	89.21	80.50	61.65	32.93
A-C2	MODGDF	97.15	89.53	78.50	57.55	32.75
A-C3	MFCC+MODGDF	97.34	91.79	83.75	64.83	35.00
A-C4	MFCC+E	98.07	81.63	70.32	51.80	28.09
A-C5	MODGDF+E	97.94	84.22	74.86	54.17	27.44
A-C6	MFCC+MODGDF+E	97.82	86.91	78.04	65.86	45.86
A-C7	MFCC+E+D+A	99.20	96.58	92.80	78.35	47.54
A-C8	MODGDF+E+D+A	98.92	95.20	88.15	72.31	47.40
A-C9	MFCC+MODGDF+E+D+A	**99.33**	**97.53**	**94.50**	**85.35**	**65.16**

car, and exhibition – sourced from the Aurora II database), are provided in Table 7.2. The trends are very similar to those that were observed for white noise. These results are by no means conclusive; they may be task specific. While the vocabulary of the ISOLET task is highly confusable, one thing that may artificially boost or reduce recognition scores is the use of the grammar, which enforces only one word per recognised utterance. This constraint eliminates the possibility of insertions and deletions. It would be interesting to see the results on a less-constrained task, such as Aurora II.

7.4.2 Connected Word Task

Aurora II caters for speaker-independent experiments using several coloured noise types and SNRs. Speech consists of digit sequences derived from the TI (Texas Instruments) digit database down-sampled to 8 kHz and filtered with a G.712 characteristic. Each digit (0-9) is modeled using a HMM with 16 emitting states and 3 Gaussian mixtures per state. We train with the clean training set (8440 utterances). The test set (28028 utterances) is divided evenly among 7 SNRs (∞ 20,15,10,5,0,-5 dB) and 4 noise types (subway, babble, car, exhibition). We use a unigram language model, where the probability of each word is equal (i.e., no grammar). The word insertion probability is set to 0. Word accuracy scores for test set A are provided in Table 7.4. Results are averaged over all noise types[7]. In an effort to be concise, we only show results for SNRs from 5 to ∞.

[7]Aurora word accuracy scores for each noise type are provided in Appendix C.

Table 7.5: RM word accuracy scores: white noise

Case	Feature type (E–energy, D–delta , A–acceleration.)	No grammar			Word pair		
		SNR (dB)			SNR (dB)		
		∞	30	20	∞	30	20
R-W1	MFCC	36.74	24.99	4.84	80.05	73.21	56.42
R-W2	MODGDF	32.68	19.21	2.62	79.62	69.54	50.88
R-W3	MFCC+MODGDF	31.78	20.03	-3.87	86.06	79.46	62.63
R-W4	MFCC+E	44.44	21.12	-4.49	84.73	74.78	51.62
R-W5	MODGDF+E	40.80	11.44	-12.73	83.87	68.33	44.87
R-W6	MFCC+MODGDF+E	42.09	13.74	-20.73	90.16	78.64	56.50
R-W7	MFCC+E+D+A	**70.13**	**52.13**	12.85	**95.67**	**92.54**	**80.05**
R-W8	MODGDF+E+D+A	62.79	42.25	-2.89	95.20	91.14	69.66
R-W9	MFCC+MODGDF+E+D+A	65.81	51.29	**14.96**	93.40	91.14	78.09

In this case, the combination of MFCCs and MODGDFs in coloured noise seems to provide more robustness than MFCCs alone[8] (compare case A-C3 to A-C1 and A-C2, compare case A-C6 to A-C4 and A-C5, and compare A-C9 to A-C7 and A-C8). This is contrary to the trends observed on the ISOLET task. However, we are dealing with a baseline performance close to 100%. It would be prudent to test on a more complex task, on which there is adequate room for improvement. Also, we should test with a more complex vocabularly (rather than a vocabulary that consists of only 10 numbers). In addition, the HMMs for this task are word-based. Do these results reflect performance for a more complex task for which the HMMs are phoneme-based?

7.4.3 Context-dependent, Phoneme-based, Continuous Recognition Task

For this experiment, we use the speaker-independent part of the Resource Management (RM) corpus, down-sampled from 16 kHz to 8 kHz. RM consists of oral readings of sentences taken from a 991-word language model of a naval resource management task. The training set consists of 3990 utterances spoken by 109 speakers. We use the February 1989 test set, which consists of 300 utterances spoken by 10 speakers.

A set of six-mixture, tied-state, cross-word triphone HMMs was trained in accordance with the RM recipe, which is supplied with the HTK distribution. Rather than use

[8]We do not perform white noise testing on Aurora II since the supplied clean data is already filtered with the G.712 characteristic. The white noise must be added before the characteristic is applied.

Table 7.6: RM word accuracy scores: averaged over several types of coloured noise

Case	Feature type (E–energy, D–delta , A–acceleration.)	No grammar			Word pair		
		SNR (dB)			SNR (dB)		
		∞	30	20	∞	30	20
R-W1	MFCC	36.74	24.83	5.48	80.05	74.53	60.86
R-W2	MODGDF	32.68	22.07	6.62	79.62	72.85	58.53
R-W3	MFCC+MODGDF	31.78	17.75	-4.52	86.06	82.03	69.58
R-W4	MFCC+E	44.44	31.95	4.85	84.73	80.15	63.48
R-W5	MODGDF+E	40.80	26.90	0.03	83.87	77.75	60.01
R-W6	MFCC+MODGDF+E	42.09	26.02	-10.55	90.16	84.37	68.01
R-W7	MFCC+E+D+A	**70.13**	**60.45**	32.51	**95.67**	**94.81**	**87.84**
R-W8	MODGDF+E+D+A	62.79	52.36	22.62	95.20	94.00	83.85
R-W9	MFCC+MODGDF+E+D+A	65.81	57.44	**32.71**	93.40	92.10	85.82

the pre-trained monophone models (accompanying the RM recipe scripts) to initiate training, we begin from a flat start. All states of all models are initialised with the global mean and variance. State transition probabilities of all states and models are initialised to common values. In the first instance, we test with a unigram language model, where the probability of each word is equal (i.e., no grammar). In addition, there is no word insertion penalty[9]. As a point of reference, we repeat the tests with a word-pair grammar (this is the standard grammar for RM). We use a grammar scale factor of 7.0 and a word insertion probability of -40.0.

We first test over several SNRs with white noise, the results of which are provided in Table 7.5. We only show three SNR conditions because below 20 dB the no grammar word accuracy scores for all cases become significantly worse. We also test over several SNRs with the same coloured noise types as used previously, the results are averaged over all noise types[10] and provided in Table 7.6.

[9]We understand that using a unigram language model and no word insertion penalty will result in reduced recognition performance. The aim here, however, is not to get the best possible performance, but to compare each feature type under 'fair' conditions. For example, if we use a bigram language model and use those recognition rates to compare features, the results will be questionable because of the high dependency on the grammar scale factor. By eliminating the need to tune this parameter (and other heuristic parameters), the recognition scores are more indicative of the raw classification ability of the feature sets.

[10]RM word accuracy scores for each noise type are provided in Appendix C.

7.5 Discussion

We have implemented Murthy and Gadde's MODGDF features and compared their recognition performance to standard MFCCs on the ISOLET, Aurora II, and Resource Management tasks with additive white and coloured noises. Our results indicate that, in isolation, the MODGDF features provide no improvement over MFCCs. In some cases, the concatenation of the MODGDF features to the MFCCs provide for a performance improvement; however, there is no consistency in the results. MFCCs seem to provide the best overall performance.

Chapter 8

Summary, Conclusions and Future Work

8.1 Chapter Summary

8.1.1 Chapter 2: Automatic Speech Recognition

In this chapter, we provided a general review of the ASR literature. We discussed the state-of-the-art in ASR and explained the statistical framework on which ASR is based. Each component of a typical ASR system was then described, from the front-end through to the back-end.

8.1.2 Chapter 3: Short-time Phase Spectrum

This chapter provided the theory of the short-time Fourier transform and discussed how it is used to analyse, synthesise and modify a speech signal. We mentioned two common representations derived from the short-time phase spectrum, namely the time-derivative (i.e., the IFD) and the frequency-derivative (i.e., the GDF). We then briefly described a number of speech applications in which these representations have successfully been used.

8.1.3 Chapter 4: Human Listening Experiments

In this chapter, the relative importance of the short-time magnitude spectrum and short-time phase spectrum on speech perception was investigated. Human perception experiments were conducted to measure intelligibility of speech stimuli reconstructed either from the original phase spectra or the original magnitude spectra. The experiments demonstrated that even for small analysis window durations of 32 ms, the phase spectrum can contribute to speech intelligibility as much as the magnitude spectrum if the analysis-modification-synthesis parameters are properly selected.

Also in this chapter, we explored the use of partial phase spectrum information, in the absence of all the magnitude spectrum information, for intelligible signal reconstruction. We created two types of stimuli; one in which the phase spectrum frequency-derivative (i.e., GDF) was preserved and another in which the phase spectrum time-derivative (i.e., IFD) was preserved. We did this to determine the contribution that each component of the phase spectrum provides toward intelligibility. Our experiments have shown that, in the absence of all other spectral information, an intelligible signal can be reconstructed with knowledge of both the time-derivative and frequency-derivative (i.e., IFD and GDF) components of the phase spectrum. However, this is not the case if only one of these components is known.

In addition, we attempted to quantify the intelligibility of stimuli reconstructed from the phase spectrum and the magnitude spectrum of noisy speech. The results indicate that the intelligibility of both the phase-only stimuli and the magnitude-only stimuli degrade at a similar rate under decreasing SNR value. While the intelligibility provided by the original signals also degrades at a similar rate, the intelligibility is consistently better than that provided by the phase-only stimuli and the magnitude-only stimuli.

8.1.4 Chapter 5: ASR on Speech Reconstructed from Short-time Phase Spectra

In this chapter, we conducted ASR experiments on phase-only stimuli and magnitude-only stimuli constructed from speech stimuli in the ISOLET and Aurora II databases.

The ASR performance for phase-only stimuli is much worse than magnitude-only stimuli at the small analysis window duration of 32 ms. This result is not consistent with the high phase-only intelligibility measured from human perception experiments. Since both phase-only and magnitude-only stimuli (reconstructed using a small analysis window duration of 32 ms) are almost equally intelligible to humans, and ASR recognises one much better than the other, it could be possible that the MFCC feature set is inadequate. That is, there seems to be some discriminating information in the phase spectrum part of the speech signal that is not being captured by the MFCC representation. These results demonstrate that there is a need for further research in the field of feature extraction.

8.1.5 Chapter 6: Iterative Reconstruction of Speech from Short-time Phase Spectra

In the first part of this chapter, we provided a tutorial-like review of iterative, one dimensional, signal reconstruction (specifically speech signals). This review was included because we believe that an appreciation for how the phase spectrum has proven useful in iterative signal reconstruction will motivate readers to investigate the potential for its use in ASR. The three main points of the review were: (i) a signal can be reconstructed to within a scale factor from its phase spectrum, (ii) a signal cannot be reconstructed to within a scale factor from its magnitude spectrum, and (iii) a signal can be reconstructed to within a scale factor from its magnitude spectrum when the phase-sign (i.e., one bit of phase information) is known. Through a number of illustrate examples, we first demonstrated how the algorithms work when the spectral information is determined over the entire duration of the signal. We then demonstrated that the algorithms are equally valid for reconstruction of a signal from the spectra obtained from short-time segments.

In addition, we presented the results of some further experimentation in which we have attempted to iteratively reconstruct a speech signal from only partial phase spectrum information (in the absence of all magnitude spectrum information). We made the following observations: (i) intelligible signal reconstruction (albeit noisy) is possible from knowledge of only the phase spectrum sign information, (ii) an intelligible signal cannot

be reconstructed from knowledge of only the phase spectrum frequency-derivative or only the phase spectrum time-derivative, and (iii) an intelligible signal can be reconstructed from the combined knowledge of both the phase spectrum frequency-derivative and time-derivative.

8.1.6 Chapter 7: Evaluation of Modified Group Delay Features on Several ASR Tasks

In Chapter 7, we first reviewed the frequency-derivative of the short-time phase spectrum (i.e., the group delay function, GDF), highlighting the problems when using it directly for ASR. We summarised the work by Yegnanarayana and Murthy [145] on the modified GDF (MGDF), which serves to remedy the problems of the GDF. We provided the implementation details of Murthy and Gadde's MGDF-based features [88] (MODGDF), then tested the MODGDF features on several ASR tasks in additive white and coloured noises.

We have implemented Murthy and Gadde's MODGDF features and compared their recognition performance to standard MFCCs on several ASR tasks (ISOLET, Aurora II and Resource Management) with additive white and coloured noises. Our results indicate that, in isolation, the MODGDF features provide no improvement over MFCCs. In some cases, the concatenation of the MODGDF features to the MFCCs provide for a performance improvement; however, there is no consistency in the results. MFCCs seem to provide the best overall performance.

8.2 Conclusions and Future Work

The human intelligibility tests demonstrate that when short-time magnitude spectra of a speech signal are set to unity, the short-time phase spectra can be used to reconstruct an intelligible signal. This is true for both small and long analysis window durations. This dispels the belief that there is no intelligibility in the short-time phase spectrum at small analysis window lengths of 20–40 ms [80, 100, 127].

An obvious question that one may ask is, "If the short-time phase spectrum provides

intelligibility, does it provide anything additional to that already conveyed by the short-time magnitude spectra?". These human intelligibility tests were performed with the aid of phase-only stimuli which were made by analysing a speech signal with an STFT, setting the magnitude spectra of each short-time segment to unity, then performing an inverse Fourier transform on each segment and reconstructing with the OLA method. Since an arbitrary change to the STFT is not necessarily a valid STFT, the resulting STFT of the phase-only stimuli does not actually have unity magnitude spectra. When the segments are overlapped and added during synthesis, their magnitude spectra deviate from unity because the samples in the overlapping regions between the segments are no longer consistent. One may suggest that it is the non-unity magnitude spectra that results in intelligibility. If this is the case (i.e., that the intelligibility is coming from the resulting short-time magnitude spectra), then ASR tests using a spectral magnitude feature should agree with the good human intelligibility scores for the phase-only stimuli. However, when using an MFCC-based front-end, we found that ASR recognition scores for phase-only stimuli are not consistent with the human intelligibility results. This leads us to believe that there is discriminating information in the phase spectrum part of the signal that is not being captured by the MFCC representation (i.e., there is information not being captured by the magnitude spectrum), providing motivation to investigate the use of the phase spectrum to derive features for ASR.

We have also attempted to answer the above question by comparing the human intelligibility of original speech, phase-only stimuli and magnitude-only stimuli under several SNRs of additive white noise. The results indicate that the intelligibility of both the phase-only stimuli and the magnitude-only stimuli degrade at a similar rate under decreasing SNR value. While the intelligibility provided by the original signals also degrades at a similar rate, the intelligibility is consistently better than that provided by both the phase-only stimuli and the magnitude-only stimuli. It is particularly interesting to see that the intelligibility provided by the original signals is far better than that provided by the magnitude-only stimuli. This result seems to be at odds with the common practice in ASR; which is to discard the phase spectrum in favour of features that are derived only from the magnitude spectrum. Should ASR features also encap-

sulate information about the phase spectrum? According to these perception results, a feature set that represents information from both the magnitude spectrum and the phase spectrum may result in improved ASR performance.

The next obvious question one may ask is, "It seems that the short-time phase spectrum does provide intelligibility, but how can we capture that for ASR?". Unlike the magnitude spectrum, the phase spectrum does not explicitly exhibit the system resonances. A physical connection between the phase spectrum and the structure of the vocal apparatus is not apparent. It is therefore necessary that the phase spectrum be transformed into a more physically meaningful representation. In this thesis we discussed the GDF and the IFD representations. These representations are much more 'human readable' than the principal phase spectrum. We investigated their possible use for ASR indirectly. That is, rather than come up with features based on the GDF or IFD, we simply attempted to reconstruct speech from either component to determine if intelligibility resulted. Since the resulting reconstructed signals were only intelligible when both components were retained, we infer from these results that we may need a feature set that consists of both GDF and IFD information. Further still, for best performance (as discussed above) the feature set may also need to be concatenated with magnitude spectral data.

The most concerted effort into phase spectral features, so far, has been that by Murthy and her colleagues. For interest, we have thoroughly tested their proposed MODGDF features on several ASR tasks. We have found that the MFCC feature set seems to provide the best overall performance. The MODGDF features do, however, provide reasonable recognition scores. Perhaps some continued effort into researching these features and a better understanding of the required parameter tuning may result in improved performance.

In addition to a hit-and-miss approach of creating phase-spectral features, one could consider employing a brute-force method such as non-linear discriminant analysis (NLDA) [32, 57, 125]. In this technique, a multi-layer perception (MLP) is trained using back-propagation with a minimum-cross-entropy criterion. The number of input nodes to the MLP is determined by the representation of choice (e.g., short-time segment GDF or

IFD) and the number of output nodes is equal to the number of classes (the MLP is trained for 'one-hot' targets). The output from this MLP is a vector of posterior probabilities which are subsequently log-compressed so that their distributions are Gaussian-like. If desired, further transformations to this vector can be applied at this point. The trained MLP (and any subsequent transformations on its output) is attached to the front-end of a HMM-based ASR system, where the output vectors are treated as the feature vectors for training and testing the HMMs.

It may also be interesting to investigate the use of the phase spectrum for speaker identification and verification. Do humans identify a person better from the original speech than from the magnitude-only reconstruction? If so, then it could be that the short-time phase spectra is useful for the task. Some initial experiments in this area have been conducted by Yegnanarayana et al. [146].

Appendix A

An Explanation of the Formant Structure in Phase-only Stimuli

In Chapter 4, we explained intelligibility results through the comparison of spectrograms. There seems to be a direct correlation between intelligibility and the presence of formant-like structure in these spectrograms. One may ask how do we get this formant structure in the spectrograms of our "phase-only" stimuli. This may happen either due to the overlap-add procedure used in the reconstruction of phase-only stimuli, or it may come as an artifact of spectrogram computation. These issues are addressed in the following discussion.

We construct a phase-only signal using a rectangular window with duration of $T_w = 32\ ms$ and frame shift of T_w (ie., no overlap). We compute the spectrogram[1] for this signal with a window duration of 32 ms and a frame shift of 32 ms. This is shown in Fig. A.1(b). As expected, we attain a flat spectrogram. Although one can hear speech in the signal (albeit with little intelligibility), it is an interesting observation that the spectrogram provides no information whatsoever. If we change the spectrogram frame shift to 1 ms, we obtain Fig. A.1(c). The reason we can now see some formant structure is because the magnitude spectrum is not unity for all frames used in the spectrogram

[1]This spectrogram is created with a rectangular analysis window and no pre-emphasis in order to visualise the effect of the unity-magnitude constraint. For consistency, all other spectrograms for this discussion are created in the same manner.

computation. The unity-magnitude spectrum constraint only exists if the spectrogram frame duration is 32 ms (the same as the reconstruction frame duration) and its ends coincide with the ends of a reconstruction frame (eg., Fig. A.2(a)). Thus, wherever the spectrogram frame does not line up with a reconstruction frame, the unity-magnitude spectrum constraint is not enforced (eg., Fig. A.2(b)). In addition, no unity-magnitude spectrum constraint exists at spectrogram frame durations less than and greater than 32 ms (eg., Fig. A.2(c) and (d)).

Next, we construct another phase-only signal using a rectangular window with the same duration ($T_w = 32ms$), but change the frame shift to $T_w/8$. Again, we create a spectrogram for this signal with a window duration of 32 ms and frame shift of 32 ms (Fig. A.1(d)). Unlike Fig. A.1(b), we see formant structure, which comes from overlapping and adding of the reconstructed frames. By using a spectrogram frame shift of 1 ms, we obtain a better view of the formant structure (Fig. A.1(e)).

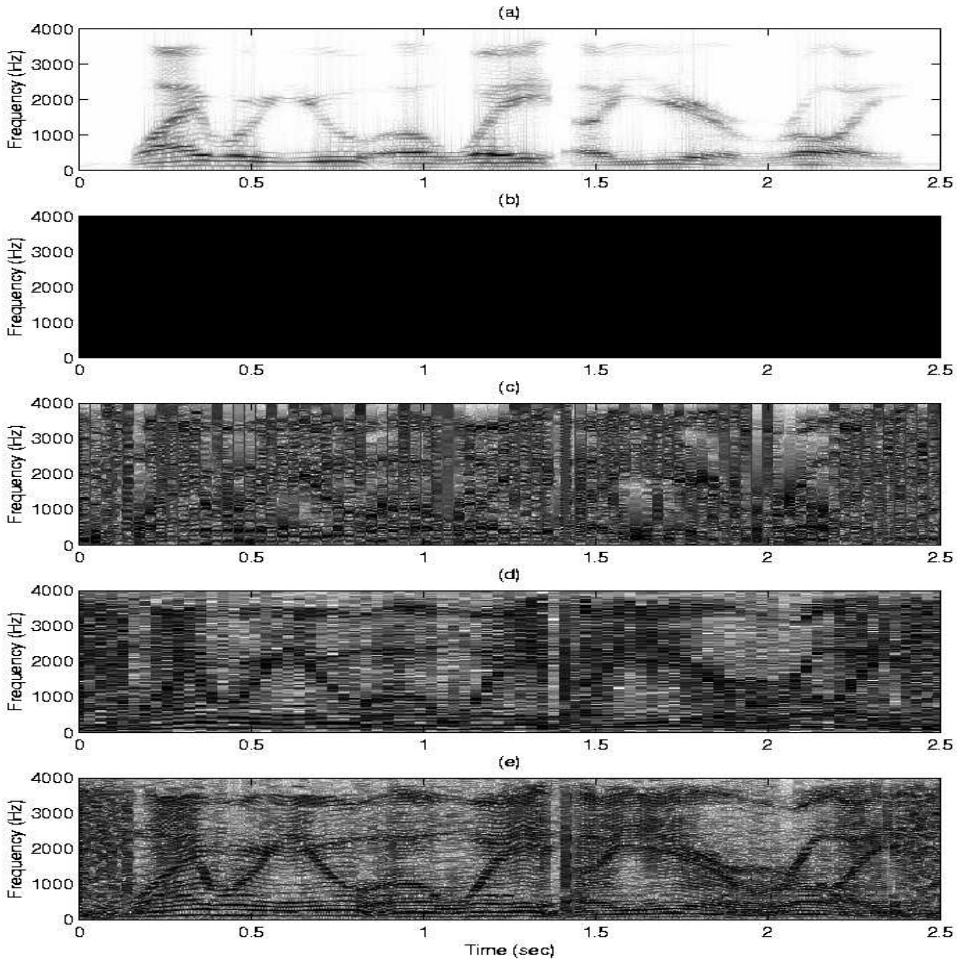


Fig. A.1: Various figures to explain the formant structure present in the phase-only stimuli. Spectrograms of (a) the original speech sentence "Why were you away a year Roy?", and the phase-only stimuli, where (b) the stimulus is constructed with frame duration of T_w and frame shift of T_w and the spectrogram is created with the same frame duration and shift, (c) as in b, but the spectrogram uses a frame shift of 1 ms, (d) the stimulus is constructed with frame duration of T_w and frame shift of $T_w/8$ and the spectrogram is created with frame duration of T_w and frame shift of T_w, and (e) as in d, but the spectrogram uses a frame shift of 1 ms.

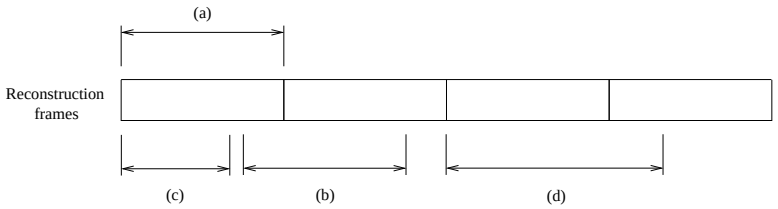


Fig. A.2: Illustration of the various ways to analyse a reconstructed signal; all of which result in different spectrograms. Reconstructed frames placed end-to-end (ie., no overlap). For the spectrogram computation, the reconstructed signal can be analysed (a) in synchrony, (b) out of synchrony, (c) at a smaller frame duration, or (d) at a longer frame duration.

Appendix B

Confusion Matrices from Human Listening Experiments

This appendix provides all of the data collected for Experiment 1 in Section 4.2.1. In each table, the labels in the top row denote the consonant that the subjects thought they heard and the leftmost column denotes the actual identity of the consonant. The numbers in each cell are the accumulation of the number of responses from each of the 12 subjects. All stimuli were created with the OLA method of reconstruction, using a 1/8 frameshift and $N = 2M$ FFT, where M is the length of the analysis window (a power of 2). Three window types have been investigated – Hamming, rectangular, and triangular – and two window durations have been investigated – small window of 32 ms and a long window of 1024 ms.

Note that for each type of stimulus, the 12 subjects were played each consonant 4 times (i.e., 2 males voices and 2 female voices); therefore, the maximum number of correct responses is 48. Also note that there are two confusion matrices for the original speech signals (Tables B.1 and B.8). The data in Table B.1 was obtained from the first sitting, where the subjects listened to the 32 ms stimuli. Table B.8 was obtained from the second sitting, where the subjects listened to the 1024 ms stimuli. Tables B.1 through B.7 comprise the data from sitting one and Tables B.8 through B.14 comprise the data from sitting two.

Table B.1: Confusion matrix: Intelligibility of original signals, sitting one.

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	1	0	0	23	23	0	0	0	0	0	0	1	0	0	0	0	0
th	0	0	0	4	43	0	0	0	0	0	0	1	0	0	0	0	0
s	0	0	0	0	5	42	0	0	0	0	0	0	1	0	0	0	0
sh	0	0	0	0	0	1	39	0	0	0	0	0	0	8	0	0	0
b	0	0	0	0	0	0	0	46	2	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	47	0	0	1	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	47	0	0	0	0	0	0
dh	3	1	0	0	2	0	0	3	16	0	0	23	0	0	0	0	0
z	0	0	0	0	0	4	0	0	0	0	0	0	42	2	0	0	0
zh	0	0	0	0	0	0	3	0	0	0	0	0	2	43	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	1	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Table B.2: Confusion matrix: Intelligibility of magnitude-only stimuli (type A1), constructed with a Hamming window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	46	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	5	0	0	12	27	0	0	0	0	0	1	3	0	0	0	0	0
th	2	1	0	5	36	0	0	0	0	0	0	3	0	0	0	0	1
s	0	0	0	0	4	44	0	0	0	0	0	0	0	0	0	0	0
sh	0	0	0	0	0	1	39	0	0	0	0	0	0	8	0	0	0
b	0	0	0	0	0	0	0	46	1	0	1	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	44	1	0	2	0	0	0	0	1
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	1	0	0	1	0	0	0	25	0	0	21	0	0	0	0	0	0
dh	6	0	0	1	9	0	0	0	1	0	0	30	0	0	0	0	1
z	0	0	0	0	0	2	0	0	0	0	1	0	44	1	0	0	0
zh	0	0	0	0	0	0	2	0	0	0	0	0	0	46	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	47	0

Table B.3: Confusion matrix: Intelligibility of magnitude-only stimuli (type B1), constructed with a rectangular window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	46	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
t	0	42	0	0	1	1	1	0	0	0	0	0	1	2	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	13	0	0	15	20	0	0	0	0	0	0	0	0	0	0	0	0
th	4	0	0	5	37	0	0	0	0	0	0	2	0	0	0	0	0
s	0	1	0	1	12	34	0	0	0	0	0	0	0	0	0	0	0
sh	0	0	0	0	0	1	38	0	0	0	0	0	0	9	0	0	0
b	1	0	0	0	0	0	0	44	1	0	1	0	0	0	0	0	1
d	0	2	0	0	2	0	0	0	30	10	0	4	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	1	0	0	0	0	0	0	31	1	0	15	0	0	0	0	0	0
dh	10	0	0	0	5	0	0	1	1	1	3	27	0	0	0	0	0
z	0	0	0	0	1	4	0	0	0	0	1	0	41	1	0	0	0
zh	0	0	0	0	0	0	4	0	0	0	0	0	1	43	0	0	0
m	0	0	0	0	0	0	0	0	0	0	1	0	0	0	46	1	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	46	0

Table B.4: Confusion matrix: Intelligibility of magnitude-only stimuli, constructed with a triangular window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	43	0	0	3	0	0	0	0	0	0	1	0	1	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	3	1	0	19	24	0	0	0	0	0	0	1	0	0	0	0	0
th	2	0	2	3	40	0	0	0	0	0	0	1	0	0	0	0	0
s	0	1	0	0	3	44	0	0	0	0	0	0	0	0	0	0	0
sh	0	0	0	0	0	1	33	0	0	0	0	0	0	14	0	0	0
b	0	0	0	0	0	0	0	43	2	0	3	0	0	0	0	0	0
d	0	2	0	0	0	0	0	0	38	0	0	8	0	0	0	0	0
g	0	0	1	0	0	0	0	0	0	47	0	0	0	0	0	0	0
v	0	0	0	0	0	0	0	15	1	0	31	1	0	0	0	0	0
dh	10	0	0	0	5	0	0	0	2	0	1	30	0	0	0	0	0
z	0	0	0	0	0	3	0	0	0	0	0	0	45	0	0	0	0
zh	0	0	0	0	0	0	7	0	0	0	0	0	1	40	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Table B.5: Confusion matrix: Intelligibility of phase-only stimuli (type C1), constructed with a Hamming window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	37	0	0	0	4	1	0	3	2	0	0	0	0	0	0	0	1
t	0	43	2	1	1	0	0	0	0	0	0	1	0	0	0	0	0
k	5	3	33	1	2	0	0	1	1	2	0	0	0	0	0	0	0
f	1	2	0	16	25	0	0	0	0	0	0	2	0	1	0	0	1
th	1	3	2	10	24	1	0	0	1	0	1	5	0	0	0	0	0
s	2	1	2	7	29	2	0	0	0	0	0	3	0	0	0	0	2
sh	0	0	6	4	24	2	8	0	0	0	0	2	0	2	0	0	0
b	0	0	0	0	0	0	0	43	1	1	2	0	0	0	1	0	0
d	0	0	0	0	0	0	0	0	44	0	0	4	0	0	0	0	0
g	0	0	1	0	1	0	0	0	11	35	0	0	0	0	0	0	0
v	0	0	0	0	2	0	0	6	1	0	37	1	0	0	0	0	1
dh	3	3	0	0	6	0	0	3	15	1	0	17	0	0	0	0	0
z	0	0	0	0	1	1	0	1	1	1	23	6	14	0	0	0	0
zh	0	3	1	2	7	0	1	0	1	2	6	4	3	18	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	4	0
n	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	44	0

Table B.6: Confusion matrix: Intelligibility of phase-only stimuli (type D1), constructed with a rectangular window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	43	0	0	0	2	0	0	3	0	0	0	0	0	0	0	0	0
t	0	47	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
k	0	2	45	0	1	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	23	23	0	1	0	0	0	0	1	0	0	0	0	0
th	0	0	1	6	35	3	1	0	0	0	0	2	0	0	0	0	0
s	0	0	0	5	17	25	1	0	0	0	0	0	0	0	0	0	0
sh	1	0	1	8	6	7	21	0	0	0	0	1	1	2	0	0	0
b	0	0	0	1	0	0	0	46	1	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	45	0	0	3	0	0	0	0	0
g	0	0	0	0	0	0	0	0	4	42	0	2	0	0	0	0	0
v	0	1	0	0	0	0	0	6	0	0	41	0	0	0	0	0	0
dh	2	0	0	0	2	0	0	0	19	0	0	25	0	0	0	0	0
z	0	0	0	0	1	2	0	0	0	0	1	0	40	4	0	0	0
zh	0	0	0	1	0	1	1	0	0	0	1	0	2	42	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	47	0	0
n	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	47	0

Table B.7: Confusion matrix: Intelligibility of phase-only stimuli, constructed with a triangular window of duration 32 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	37	0	0	0	5	0	0	6	0	0	0	0	0	0	0	0	0
t	1	35	5	1	2	0	0	0	1	0	0	3	0	0	0	0	0
k	6	4	30	0	2	0	0	0	0	4	0	1	0	0	0	0	1
f	3	1	2	18	22	0	0	0	0	0	0	2	0	0	0	0	0
th	9	7	5	1	19	0	0	0	1	0	0	6	0	0	0	0	0
s	3	1	1	11	28	0	0	0	0	0	0	3	0	0	0	0	1
sh	1	0	11	3	23	0	6	0	0	0	0	3	0	1	0	0	0
b	0	0	0	0	0	0	0	44	1	0	1	2	0	0	0	0	0
d	0	0	0	0	0	0	0	1	43	0	0	4	0	0	0	0	0
g	0	0	0	0	1	0	0	3	8	36	0	0	0	0	0	0	0
v	0	0	0	0	0	0	0	20	0	3	25	0	0	0	0	0	0
dh	3	0	1	0	1	0	0	3	17	1	0	19	0	0	0	0	3
z	0	1	0	1	4	0	0	7	4	2	17	8	2	1	0	0	1
zh	0	1	1	2	6	0	0	7	2	3	4	7	1	14	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0
n	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	37	3

Table B.8: Confusion matrix: Intelligibility of original signals, sitting two.

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	1	0	28	18	0	0	0	0	0	0	1	0	0	0	0	0
th	0	0	0	2	44	0	0	0	0	0	0	2	0	0	0	0	0
s	0	0	0	1	10	36	0	0	0	0	0	0	1	0	0	0	0
sh	0	0	0	0	0	0	47	0	0	0	0	0	0	1	0	0	0
b	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	45	0	0	3	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0
dh	4	0	0	0	4	0	0	3	10	0	0	27	0	0	0	0	0
z	0	0	0	0	0	1	0	0	0	0	0	0	46	1	0	0	0
zh	0	0	0	0	0	0	5	0	0	0	0	0	1	42	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	1	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Table B.9: Confusion matrix: Intelligibility of magnitude-only stimuli (type E1), constructed with a Hamming window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	5	0	1	2	1	0	0	1	1	6	3	0	0	1	3	7	17
t	1	5	1	2	2	5	2	1	2	1	1	2	1	5	0	3	14
k	2	1	2	1	3	0	1	4	2	6	3	0	0	2	2	6	13
f	1	0	2	4	1	2	1	4	3	4	3	1	1	1	3	3	14
th	8	1	3	2	0	0	0	5	0	5	0	2	0	1	2	4	15
s	2	0	1	0	1	17	6	1	1	1	0	0	2	4	0	1	11
sh	0	1	0	1	2	0	26	0	1	3	0	0	0	5	0	4	5
b	3	0	2	1	1	0	0	3	1	3	0	0	0	0	7	6	21
d	3	0	2	1	1	1	0	5	3	5	1	0	1	0	3	5	17
g	2	0	0	1	1	0	0	7	1	5	2	2	0	1	1	8	17
v	4	1	4	0	1	0	0	2	2	5	2	2	1	0	4	5	15
dh	1	4	3	0	0	0	0	4	4	8	1	3	2	0	0	3	15
z	0	0	1	1	1	12	3	3	1	2	2	2	6	1	1	2	10
zh	1	0	1	0	0	1	12	2	1	1	1	0	0	9	0	1	18
m	2	0	2	1	1	3	0	1	0	1	1	0	0	0	12	10	14
n	0	4	2	0	1	1	0	0	4	4	1	0	0	0	11	6	14

Table B.10: Confusion matrix: Intelligibility of magnitude-only stimuli (type F1), constructed with a rectangular window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	4	3	1	2	4	0	0	2	1	4	0	1	0	0	3	2	21
t	1	2	1	0	0	2	8	1	1	2	2	1	0	7	1	2	17
k	5	1	6	0	0	0	0	2	2	1	1	1	0	1	4	0	24
f	4	2	1	0	2	0	0	1	1	1	3	0	1	0	3	7	22
th	3	0	3	2	1	0	0	0	3	2	2	2	0	1	2	1	26
s	2	3	0	0	3	17	3	2	0	0	0	0	2	1	1	0	14
sh	0	0	0	0	0	0	28	0	0	0	0	1	0	4	1	2	12
b	2	0	0	2	1	0	1	5	2	1	6	0	2	0	1	2	23
d	5	1	2	2	0	2	2	7	1	3	2	0	0	0	1	4	16
g	2	1	0	0	0	0	0	1	0	6	0	4	1	0	2	6	25
v	6	0	0	2	2	0	0	4	2	2	2	0	0	2	4	4	18
dh	2	0	4	0	0	0	1	4	3	1	1	2	2	2	2	0	24
z	2	2	0	1	3	4	0	1	0	4	1	0	3	5	2	2	18
zh	0	0	0	0	0	1	13	1	2	1	1	0	1	7	0	3	18
m	3	1	1	1	1	0	0	1	2	0	0	1	0	0	7	10	20
n	2	0	0	1	0	0	1	1	0	4	0	1	0	2	8	10	18

Table B.11: Confusion matrix: Intelligibility of magnitude-only stimuli, constructed with a triangular window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	6	1	3	1	0	0	0	1	1	3	2	0	2	0	1	7	20
t	1	3	5	0	2	3	8	0	2	2	2	0	1	5	0	0	14
k	2	0	2	1	2	1	0	2	3	5	1	2	1	0	3	7	16
f	6	0	2	2	1	0	1	4	0	5	2	1	1	2	1	2	18
th	6	0	2	1	1	0	1	4	3	2	2	3	1	1	1	4	16
s	1	4	0	0	2	16	2	3	1	1	0	0	3	1	3	1	10
sh	0	0	0	0	1	0	28	0	1	1	0	0	2	6	0	0	9
b	7	0	0	0	0	0	1	5	4	5	1	1	0	1	5	5	13
d	1	2	1	0	1	1	1	5	4	6	1	0	1	2	0	4	18
g	2	1	6	1	1	0	1	3	3	4	2	1	0	0	2	8	13
v	5	0	2	0	1	0	0	9	1	3	3	0	0	0	6	0	18
dh	1	0	3	2	1	1	0	3	2	8	0	0	0	0	5	2	20
z	1	2	0	2	1	6	4	2	3	4	1	0	9	4	2	0	7
zh	0	0	1	0	0	0	16	0	1	2	1	0	0	8	0	4	15
m	2	0	0	0	0	0	1	4	0	0	1	1	0	0	9	11	19
n	2	1	1	0	0	1	0	2	1	5	0	0	1	1	9	7	17

Table B.12: Confusion matrix: Intelligibility of phase-only stimuli (type G1), constructed with a Hamming window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	47	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
t	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	21	20	5	0	0	0	0	0	1	1	0	0	0	0
th	0	1	0	5	41	0	0	0	0	0	0	1	0	0	0	0	0
s	0	1	0	2	6	35	0	0	0	0	0	0	3	1	0	0	0
sh	0	0	0	0	0	0	46	0	0	0	0	0	0	2	0	0	0
b	0	0	0	0	0	0	0	47	1	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	44	0	0	4	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	0	0	0	0	0	0	0	3	0	0	45	0	0	0	0	0	0
dh	9	0	0	0	0	0	0	1	15	0	1	22	0	0	0	0	0
z	0	0	0	0	0	2	0	0	0	0	0	0	45	1	0	0	0
zh	0	0	0	0	0	0	4	0	0	0	0	0	0	44	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	1	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Table B.13: Confusion matrix: Intelligibility of phase-only stimuli (type H1), constructed with a rectangular window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	47	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
t	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	46	0	0	0	0	0	0	0	0	0	0	0	2	0	0
f	0	0	0	17	22	9	0	0	0	0	0	0	0	0	0	0	0
th	0	1	0	3	43	1	0	0	0	0	0	0	0	0	0	0	0
s	0	1	0	0	1	46	0	0	0	0	0	0	0	0	0	0	0
sh	0	0	0	0	0	0	47	0	0	0	0	0	0	1	0	0	0
b	0	0	0	0	0	0	0	47	0	0	1	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	44	0	0	4	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	47	0	0	0	0	0	0
dh	7	0	0	0	1	0	0	1	16	0	0	23	0	0	0	0	0
z	0	0	0	0	0	3	0	0	0	0	0	0	44	1	0	0	0
zh	0	0	0	0	0	0	3	0	0	0	0	0	0	45	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	2	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Table B.14: Confusion matrix: Intelligibility of phase-only stimuli, constructed with a triangular window of duration 1024 ms

	p	t	k	f	th	s	sh	b	d	g	v	dh	z	zh	m	n	?
p	47	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	0	0	22	23	3	0	0	0	0	0	0	0	0	0	0	0
th	0	0	0	3	44	0	0	0	0	0	0	1	0	0	0	0	0
s	0	1	0	2	5	37	0	0	0	0	0	0	3	0	0	0	0
sh	0	0	0	0	0	0	46	0	0	0	0	0	0	2	0	0	0
b	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0
d	0	0	0	1	0	0	0	0	44	0	0	3	0	0	0	0	0
g	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0
v	0	0	0	0	1	0	0	1	0	0	46	0	0	0	0	0	0
dh	8	0	0	0	1	0	0	0	12	0	0	26	0	0	0	0	1
z	0	0	0	0	0	1	0	0	0	0	0	0	47	0	0	0	0
zh	0	0	0	0	0	0	2	0	0	0	0	0	0	46	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	0

Appendix C

Detailed ASR Results: Modified Group Delay Feature Experiments

In Section 7.4, we conducted a number of ASR experiments on several tasks to compare the performance of MFCC and MODGDF features. When comparing feature performance in additive coloured noise, we only provided recognition scores that were averaged over four types; subway, babble, car, and exhibition. In this appendix, we provide the recognition scores for each noise type. We also provide the detailed results of the line-search that we performed on ISOLET in order to determine the best values for α, γ, and s_w (see Section 7.4.1).

Table C.1: ISOLET word recognition scores: detailed coloured noise results

Feature type (E–energy, D–delta , A–acceleration.)	Noise type	SNR (dB)				
		∞	30	20	15	10
MFCC	Subway	78.27	77.44	68.72	56.99	40.71
	Babble	78.27	75.96	65.32	52.76	35.71
	Car	78.27	75.51	69.04	60.38	48.40
	Exhibition	78.27	77.37	69.94	61.22	43.27
	Average	78.27	76.57	68.26	57.84	42.02
MODGDF	Subway	76.79	73.27	60.77	40.19	20.32
	Babble	76.79	75.38	66.86	51.47	29.94
	Car	76.79	75.32	66.99	54.55	31.47
	Exhibition	76.79	72.69	56.60	35.13	15.51
	Average	76.79	74.17	62.81	45.34	24.31
MFCC+E	Subway	79.62	71.15	59.68	40.00	25.38
	Babble	79.62	68.59	55.77	41.15	21.73
	Car	79.62	71.99	63.01	50.58	32.24
	Exhibition	79.62	73.08	61.54	39.74	22.12
	Average	79.62	71.20	60.00	42.87	25.37
MODGDF+E	Subway	78.65	63.91	44.55	21.41	10.00
	Babble	78.65	69.94	58.01	33.40	14.55
	Car	78.65	69.36	56.73	31.47	13.78
	Exhibition	78.65	58.72	34.55	15.26	8.78
	Average	78.65	65.48	48.46	25.39	11.78
MFCC+E+D+A	Subway	90.83	86.35	79.55	66.99	47.82
	Babble	90.83	86.09	79.49	67.18	52.18
	Car	90.83	86.09	81.79	70.38	54.81
	Exhibition	90.83	86.15	79.04	65.83	43.65
	Average	90.83	86.17	79.97	67.60	49.62
MODGDF+E+D+A	Subway	89.29	80.71	70.19	51.15	26.86
	Babble	89.29	84.36	76.67	65.06	48.21
	Car	89.29	84.10	76.47	63.46	40.26
	Exhibition	89.29	79.36	66.79	45.19	17.88
	Average	89.29	82.13	72.53	56.22	33.30
MFCC+MODGDF+E+D+A	Subway	91.41	83.46	76.86	61.67	38.91
	Babble	91.41	85.96	81.15	70.83	52.24
	Car	91.41	86.86	80.71	70.32	52.44
	Exhibition	91.41	84.29	75.71	58.14	29.17
	Average	91.41	85.14	78.61	65.24	43.19
MFCC+MODGDF	Subway	78.59	67.88	52.31	30.83	17.63
	Babble	78.59	68.65	54.49	30.00	12.82
	Car	78.59	70.45	61.86	44.87	20.06
	Exhibition	78.59	64.23	48.65	25.38	15.26
	Average	78.59	67.80	54.33	32.77	16.44
MFCC+MODGDF+E	Subway	79.42	68.72	51.35	29.62	15.26
	Babble	79.42	70.83	58.65	33.53	12.31
	Car	79.42	73.08	63.27	44.17	19.36
	Exhibition	79.42	64.55	44.81	23.59	13.21
	Average	79.42	69.30	54.52	32.73	15.04
MFCC	Subway	92.31	84.81	78.59	64.55	41.35
	Babble	92.31	86.79	81.22	71.22	52.12
	Car	92.31	87.69	82.69	71.86	52.95
	Exhibition	92.31	84.55	77.12	59.94	29.04
	Average	92.31	85.96	79.91	66.89	43.87

Table C.2: ISOLET word recognition scores: detailed MODGDF-tuning results

s_w	α	γ					
		1.0	0.9	0.8	0.7	0.6	0.5
4	0.1	91.22	90.26	91.22	90.71	91.28	91.54
	0.2	91.03	91.22	91.73	91.60	91.54	90.93
	0.3	90.64	91.09	92.05	90.90	90.64	90.71
	0.4	91.15	89.94	90.77	90.32	89.62	88.53
	0.5	89.55	90.19	90.96	89.42	88.59	87.95
6	0.1	90.77	90.58	91.41	90.77	91.41	91.60
	0.2	91.22	90.58	91.28	91.28	92.12	90.77
	0.3	91.15	91.41	91.86	91.22	90.83	90.19
	0.4	91.03	90.83	91.41	90.58	89.74	89.10
	0.5	90.32	91.09	91.15	90.32	89.81	88.65
8	0.1	91.35	91.35	91.54	90.90	91.60	91.22
	0.2	91.60	91.15	91.60	91.54	91.67	91.09
	0.3	91.35	**92.31**	92.18	91.35	91.41	90.64
	0.4	90.90	92.24	91.22	90.19	90.51	88.72
	0.5	91.03	90.77	91.41	89.81	88.91	88.33
10	0.1	91.09	91.28	91.35	91.15	91.54	91.79
	0.2	91.54	91.03	91.09	91.79	91.99	91.09
	0.3	91.22	91.79	91.03	91.54	90.83	90.77
	0.4	90.51	91.41	91.73	90.90	90.71	88.78
	0.5	90.45	91.03	91.47	90.77	89.10	88.91

Table C.3: Aurora II word accuracy scores: detailed results

Feature type (E–energy, D–delta , A–acceleration.)	Noise type	SNR (dB)				
		∞	20	15	10	5
MFCC	Subway	97.33	89.50	80.87	62.11	34.60
	Babble	96.89	90.36	82.71	66.29	37.09
	Car	97.14	90.61	82.05	61.29	29.65
	Exhibition	96.82	86.36	76.37	56.90	30.36
	Average	97.05	89.21	80.50	61.65	32.93
MODGDF	Subway	97.08	89.96	80.01	59.10	36.26
	Babble	96.77	90.69	82.19	63.91	38.21
	Car	97.26	89.29	76.77	54.07	27.26
	Exhibition	97.50	88.18	75.04	53.10	29.25
	Average	97.15	89.53	78.50	57.55	32.75
MFCC+E	Subway	98.40	75.31	64.69	45.56	27.05
	Babble	97.61	88.30	78.30	59.58	31.23
	Car	97.97	85.42	77.87	60.13	30.21
	Exhibition	98.30	77.48	60.41	41.93	23.85
	Average	98.07	81.63	70.32	51.80	28.09
MODGDF+E	Subway	98.25	81.21	71.42	48.11	24.93
	Babble	97.46	88.18	79.38	59.70	29.35
	Car	97.76	86.10	77.87	58.43	30.00
	Exhibition	98.27	81.39	70.75	50.45	25.46
	Average	97.94	84.22	74.86	54.17	27.44
MFCC+E+D+A	Subway	99.11	96.13	91.53	73.63	46.15
	Babble	99.21	97.97	94.95	79.96	45.19
	Car	99.14	97.35	95.26	85.80	56.37
	Exhibition	99.32	94.88	89.45	73.99	42.46
	Average	99.20	96.58	92.80	78.35	47.54
MODGDF+E+D+A	Subway	98.89	93.21	83.36	66.20	46.45
	Babble	98.91	97.49	93.68	78.69	49.33
	Car	98.69	96.66	91.65	76.92	51.06
	Exhibition	99.20	93.43	83.89	67.42	42.76
	Average	98.92	95.20	88.15	72.31	47.40
MFCC+MODGDF+E+D+A	Subway	99.32	97.64	93.92	84.22	65.27
	Babble	99.18	97.91	95.77	87.09	66.17
	Car	99.28	97.91	95.41	87.41	68.42
	Exhibition	99.54	96.67	92.90	82.69	60.78
	Average	99.33	97.53	94.50	85.35	65.16
MFCC+MODGDF	Subway	97.54	91.93	83.97	65.86	37.76
	Babble	97.10	92.74	86.28	69.35	38.60
	Car	97.17	92.19	83.84	61.77	29.62
	Exhibition	97.53	90.28	80.90	62.33	34.00
	Average	97.34	91.79	83.75	64.83	35.00
MFCC+MODGDF+E	Subway	98.13	82.65	74.79	63.80	46.27
	Babble	97.49	91.14	82.50	69.62	45.92
	Car	97.55	88.82	80.35	67.85	47.93
	Exhibition	98.09	85.04	74.51	62.17	43.32
	Average	97.82	86.91	78.04	65.86	45.86

Table C.4: RM word accuracy scores: detailed coloured noise results

Feature type (E–energy, D–delta , A–acceleration.)	Noise type	Word pair grammar			No grammar		
		SNR					
		∞	30	20	∞	30	20
MFCC	Subway	36.74	24.21	3.83	80.05	73.21	56.93
	Babble	36.74	16.17	-2.19	80.05	74.23	61.73
	Car	36.74	30.14	11.87	80.05	76.02	65.79
	Exhibition	36.74	28.78	8.40	80.05	74.66	59.00
	Average	36.74	24.83	5.48	80.05	74.53	60.86
MODGDF	Subway	32.68	22.30	5.74	79.62	72.55	55.72
	Babble	32.68	17.14	2.46	79.62	73.21	62.01
	Car	32.68	27.53	13.86	79.62	75.13	64.00
	Exhibition	32.68	21.32	4.41	79.62	70.52	52.40
	Average	32.68	22.07	6.62	79.62	72.85	58.53
MFCC+E	Subway	44.44	28.43	-0.94	84.73	77.94	57.24
	Babble	44.44	34.01	5.51	84.73	80.98	68.37
	Car	44.44	33.89	7.54	84.73	82.08	67.98
	Exhibition	44.44	31.47	7.30	84.73	79.58	60.33
	Average	44.44	31.95	4.85	84.73	80.15	63.48
MODGDF+E	Subway	40.80	23.04	-6.05	83.87	75.56	54.59
	Babble	40.80	31.08	3.16	83.87	80.79	66.46
	Car	40.80	31.63	7.65	83.87	79.27	64.82
	Exhibition	40.80	21.91	-4.65	83.87	75.36	54.16
	Average	40.80	26.92	0.03	83.87	77.75	60.01
MFCC+E+D+A	Subway	70.13	57.52	24.8	95.67	93.95	83.48
	Babble	70.13	62.09	38.23	95.67	95.16	91.41
	Car	70.13	63.80	43.38	95.67	95.51	91.29
	Exhibition	70.13	58.38	23.62	95.67	94.61	85.16
	Average	70.13	60.45	32.51	95.67	94.81	87.84
MODGDF+E+D+A	Subway	62.79	48.18	13.04	95.20	93.64	78.72
	Babble	62.79	54.98	32.60	95.20	94.65	89.81
	Car	62.79	57.59	33.89	95.20	94.65	88.56
	Exhibition	62.79	48.69	10.93	95.20	93.05	78.29
	Average	62.79	52.36	22.62	95.20	94.00	83.85
MFCC+MODGDF+E+D+A	Subway	65.81	55.35	26.27	93.40	91.41	82.66
	Babble	65.81	56.57	34.71	93.40	92.70	88.64
	Car	65.81	61.62	43.23	93.40	93.40	88.91
	Exhibition	65.81	56.23	26.63	93.40	90.90	83.05
	Average	65.81	57.44	32.71	93.40	92.10	85.82
MFCC+MODGDF	Subway	31.78	20.03	-3.87	86.06	81.96	65.68
	Babble	31.78	19.45	-2.19	86.06	80.36	68.96
	Car	31.78	7.81	-18.63	86.06	84.11	76.53
	Exhibition	31.78	23.70	6.60	86.06	81.69	67.16
	Average	31.78	17.75	-4.52	86.06	82.03	69.58
MFCC+MODGDF+E	Subway	12.09	23.86	-14.96	90.16	82.74	63.41
	Babble	12.09	23.82	-13.12	90.16	84.38	69.43
	Car	12.09	29.87	-5.78	90.16	86.26	73.41
	Exhibition	12.09	26.51	-8.32	90.16	84.11	65.79
	Average	12.09	26.02	-10.55	90.16	84.37	68.01

Appendix D

Matlab code

Here we provide the Matlab algorithm used for phase-only and magnitude-only stimuli construction. Example audio files are available at http://maxwell.me.gu.edu.au/spl/research/phase/project.htm.

```
% Algorithm for phase-only and magnitude-only stimuli construction
% Leigh Alsteris

clear;

% Variables
fs = 16000;                 % Sampling Frequency in Hertz
T = 32;                     % Window length in milliseconds
Ns = 0.125 ;                % Window shift is between 0 and 1
F = 1 - Ns ;                % Fractional overlap
N = round(fs * T * 0.001) ; % Number of samples in a segment
wnd = (hamming(N))' ;       % Window type
M = round(N*F) ;            % The number of overlapping samples in each segment
inc = N-M ;

speech_in = wavread('speech16.wav') ; % Speech is sampled at fs
```

```
% Append zeros to start and end of signal to allow for integer number of segments
D = mod(length(speech_in), inc) ;
G = (ceil(N/inc)-1)*inc ;
speech = [zeros(1,G) speech_in' zeros(1,N-D)];

% Enframe the speech into a matrix of frames, were each row is a frame
segments = ((length(speech)-N)/inc)+1 ;               % Number of overlapping segments
indf = inc*(0:(segments-1))';                         % Frame index
inds = (1:N) ;                                        % Sample index
srefs = indf(:,ones(1,N)) + inds(ones(segments,1),:) ; % Sample refs
speech_segment = speech(srefs);
speech_segment = speech_segment .* wnd(ones(segments,1),:) ; % Apply window to frames

speech_segment = [speech_segment zeros(segments,N)]; % Zero padding

SPEECH_SEGMENT = fft(speech_segment').' ;

PO_SEGMENT=exp(j*angle(SPEECH_SEGMENT));    % Phase only - unity magnitude

[R,C] = size(SPEECH_SEGMENT) ;
r=(rand(R,C)-0.5)*2*pi;
MO_SEGMENT=abs(SPEECH_SEGMENT) .*exp(j*r);  % Magnitude only - randomised phase

po_segment = (real(ifft(PO_SEGMENT.')))' ;
mo_segment = (real(ifft(MO_SEGMENT.')))' ;

po_segment = po_segment(1:segments,1:N) ;   % Remove the extra samples from the convolution
mo_segment = mo_segment(1:segments,1:N) ;

% Apply window again, if using Griffin & Lim's method
% po_segment = po_segment .* wnd(ones(segments,1),:);
% mo_segment = mo_segment .* wnd(ones(segments,1),:);
```

```
po = zeros(1, length(speech)) ;
mo = zeros(1, length(speech)) ;
wsum = zeros(1, length(speech)) ;

% Add overlapping segments to construct output signal
for i = 1:segments
  po(srefs(i,:)) = po(srefs(i,:)) + po_segment(i,:) ;
  mo(srefs(i,:)) = mo(srefs(i,:)) + mo_segment(i,:) ;
  wsum(srefs(i,:)) = wsum(srefs(i,:)) + wnd ;      % Allen & Rabiner's method
  % wsum(srefs(i,:)) = wsum(srefs(i,:)) + wnd.^2 ; % Griffin & Lim's method
end

po = po./wsum ;                 % Divide out the analysis window
mo = mo./wsum ;

po = po(G+1:length(po)-(N-D)) ; % Remove extra samples from ends
mo = mo(G+1:length(mo)-(N-D)) ;

po = (po/max(abs(po)))*0.99 ;   % Values outside the range [-1,+1] will be clipped
mo = (mo/max(abs(mo)))*0.99 ;

wavwrite(po,16000,16,'phsonly.wav')  % Write data to wave files
wavwrite(mo,16000,16,'magonly.wav')
```

Bibliography

[1] T.Abe, T. Kobayashi, and S. Imai, "Harmonics tracking and pitch extraction based on instantaneous frequency", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 756–759, May 1995.

[2] A. Acero and R.M. Stern, "Environmental robustness in automatic speech recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 849–852, Apr. 1990.

[3] P. Alexandre, J. Boudy, and P. Lockwood, "Root homomorphic deconvolution schemes for speech recognition in car noise environments", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 99–102, Apr. 1993.

[4] J.B. Allen and L.R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis" *Proceedings of the IEEE*, Vol. 65, No. 11, pp. 1558–1564, Nov. 1977.

[5] J.B. Allen, "Short-term spectral analysis, synthesis, and modification by discrete Fourier transform", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-25, No. 3, pp. 235–238, June 1977.

[6] L.D. Alsteris and K.K. Paliwal, "On the importance of phase spectrum in speech perception", *Proc. International Conf. Perception and Action*, July 2003.

[7] L.D. Alsteris and K.K. Paliwal, "Intelligibility of speech from phase spectrum", *Proc. Microelectronic Engineering Research Conf.*, Nov. 2003.

[8] L.D. Alsteris and K.K. Paliwal, "Importance of window shape for phase-only reconstruction of speech", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 573–576, May 2004.

[9] L.D. Alsteris and K.K. Paliwal, "Further intelligibility results from human listening tests using the short-time phase spectrum", *Speech Communication*, in press.

[10] L.D. Alsteris and K.K. Paliwal, "ASR on speech reconstruction from short-time Fourier phase specta", *Proc. International Conf. Spoken Language Processing*, 2004.

[11] L.D. Alsteris and K.K. Paliwal, "Evaluation of the modified group delay feature for isolated word recognition", *Int. Symposium on Signal Processing and its Applications*, pp. 715–718, Aug. 2005.

[12] L.D. Alsteris and K.K. Paliwal, "Some experiments on iterative reconstruction of speech from STFT phase and magnitude spectra", *Proc. European Conf. Speech Communication and Technology*, pp.337–340, 2005.

[13] B.S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification", *J. Acoust. Soc. Am.*, Vol. 55, No. 6, pp. 1304–1312, June 1974.

[14] B.S. Atal and S.L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave", *J. Acoust. Soc. Am.*, Vol. 50, No. 2, pp. 637–655, Aug. 1971.

[15] H. Banno, K. Takeda and F. Itakura, "The effect of group delay spectrum on timbre", *Acoust. Sci. and Tech.*, Vol. 23, pp. 1–9, 2002.

[16] L. Barbier and G. Chollet, " Robust speech parameters extraction for word recognition in noise using neural networks", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 145–148, May 1991.

[17] J.A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models", Technical Report TR-97-021, International Computer Science Institute, Berkeley, California, Apr. 1998.

[18] E. de Boer, "A note on phase distortion in hearing", *Acoustica*, Vol. 11, pp. 182–184, 1961.

[19] S.F. Boll, "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-27, No. 2, pp. 113–120, Apr. 1979.

[20] B. Bozkurt, B. Doval, C. D'Alessandro, and T. Dutoit, "Zeros of z-transform (ZZT) decomposition of speech for source-tract separation", *Proc. International Conf. Spoken Language Processing*, Oct. 2004.

[21] B. Bozkurt, B. Doval, C. D'Alessandro, and T. Dutoit, "Appropriate windowing for group delay analysis and roots of z-transform of speech signals", *European Signal Processing Conf.*, Sept. 2004.

[22] F. J. Charpentier, "Pitch detection using the short-term phase spectrum", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 113–116, Apr. 1986.

[23] R.V. Cox, C.A. Kamm, L.R. Rabiner, J. Schroeter, and J.G. Wilpom, "Speech and language processing for the next-millennium communication services", *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1314–1337, Aug. 2000.

[24] R.C. Cox and D.M. Robinson, "Some notes on phase in speech signals", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 150–153, Apr. 1980.

[25] R.E. Crochiere, "A weighted overlap-add method of short-time Fourier analysis/synthesis", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 1, pp. 99–102, Feb. 1980.

[26] B.A. Dautrich, L.R. Rabiner, and T.B. Martin "On the use of filter bank features for isolated word recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 1061–1064, Apr. 1983.

[27] S.B. Davis and P. Mermelstein, "Comparison of parametric representations for mono-syllabic word recognition in continuously spoken utterances", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 28, No. 4, pp. 357–366, Aug. 1980.

[28] M. Dendrinos, S. Bakamidis, and G. Carayannis, "Speech enhancement from noise: a regenerative approach", *Speech Communication*, Vol. 10, No. 2, pp. 45–47, Feb. 1991.

[29] D. Dimitriadis and P. Maragos, "Robust energy demodulation based on continuous models with application to speech recognition", *Proc. European Conf. Speech Communication and Technology*, pp. 2853–2856, Sept. 2003.

[30] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification*, 2nd ed., John Wiley and Sons, 2001.

[31] G. Duncan, B. Yegnanarayana, and Hema A. Murthy, "A nonparametric method of formant estimation using group delay spectra", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 572–575, May 1989.

[32] D.P.W. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 517–520, May 2001.

[33] W.M. Fisher, G.R. Doddington, and K.M. Goudie-Marshall, "The DARPA speech recognition research database: specifications and status", *Proc. DARPA Workshop on Speech Recognition*, pp. 93–99, Feb. 1986.

[34] J.L. Flanagan and R.M. Golden, "Phase Vocoder", *Bell System Technical Journal*, Vol. 45, pp. 1493–1509, Nov. 1966.

[35] J.L. Flanagan, *Speech analysis, synthesis, and perception*, 2nd ed., Springer-Verlag, New York, 1972.

[36] J.A.N. Flores and S.J. Young, "Adapting a hmm-based recogniser for noisy speech enhanced by spectral subtraction", *Proc. European Conf. Speech Communication and Technology*, pp. 829–832, Sept. 1993.

[37] R.H. Frazier, S. Samsam, L.D. Braida, and A.V. Oppenheim, "Enhancement of speech by adaptive filtering", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 251–253, Apr. 1976.

[38] David H. Friedman, "Instantaneous-frequency distribution vs. time: an interpretation of the phase structure of speech", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 1121–1124, Mar. 1985.

[39] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 34, No. 1, pp. 52–59, Feb. 1986.

[40] B. Gajic and K.K. Paliwal, "Robust feature extraction using subband spectral centroid histograms", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 85–88, May 2001.

[41] M.J. Gales, "Model based techniques for noise robust speech recognition", Phd Thesis in Engineering Department, Cambridge University, 1995.

[42] Y. Gao, T. Huang, and J. Haton, "Central auditory model for spectral processing", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 704–707, Apr. 1993.

[43] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", *IEEE Trans. Speech Audio Processing*, Vol. 2, pp. 291–298, Apr. 1994.

[44] D.C. Ghiglia and M.D. Pritt, *Two-dimensional phase unwrapping. Theory, algorithms and software*, Wiley, New York, 1998.

[45] O. Ghitza, "Auditory models and human performance in tasks related to speech coding and speech recognition", *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 1, pp. 115–132, Jan. 1994.

[46] J.J. Godfrey, E.C. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 517–520, Mar. 1992.

[47] D.W. Griffin and J.S. Lim, "Signal estimation from modified short-time Fourier transform", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-32, No. 2, pp. 236–243, Apr. 1984.

[48] L. Gu and K. Rose, "Perceptual harmonic cepstral coefficients for speech recognition in noisy environment", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 125–128, May 2001.

[49] F.J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform", *Proceedings of the IEEE*, Vol. 66, No. 1, Jan. 1978.

[50] M.H. Hayes, J.S. Lim, and A.V. Oppenheim, "Signal reconstruction from phase or magnitude", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 6, pp. 672–680, Dec. 1980.

[51] M.H. Hayes, "The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-30, No. 2, pp. 140–154, Apr. 1982.

[52] R.M. Hegde, H.A. Murthy, and G.V.R. Rao, "Application of the modified group delay function to speaker identification and discrimination", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 517–520, May 2004.

[53] R.M. Hegde, H.A. Murthy, and G.V.R. Rao, "Continuous speech recognition using joint features derived from the modified group delay function and MFCC", *Proc. International Conf. Spoken Language Processing*, Oct. 2004.

[54] R.M. Hegde, H.A. Murthy, and G.V.R. Rao, "The modified group delay feature: a new spectral representation of speech", *Proc. International Conf. Spoken Language Processing*, Oct. 2004.

[55] H.L.F. von Helmholtz, *On the Sensations of Tone*, 1875, (English Translation by A.J. Ellis, Longmans, Green and Co., London, 1912).

[56] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", *J. Acoust. Soc. Am.*, Vol. 87, No. 4, pp. 1738–1752, Apr. 1990.

[57] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 1635–1638, June 2000.

[58] H. Hermansky and N. Morgan, "RASTA processing of speech", *IEEE Trans. Speech and Audio Processing*, Vol. 2, pp. 578–589, Oct. 1994.

[59] J. Hernando and C. Nadeu, "Speech representation in noisy car environment based on OSALPC representation and robust similarity measuring techniques", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 69–72, Apr. 1994.

[60] W. Hess, *Pitch determination of speech signals*, Springer-Verlag, Berlin, 1983.

[61] X. Huang, "Speaker normalization for speech recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 465–468, Mar. 1992.

[62] X. Huang, A. Acero, and H. Hon, *Spoken language processing*, Prentice Hall, New Jersey, 2001.

[63] X.D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov models for speech recognition*, Edinburgh University Press, 1990.

[64] M. Hwang and X. Hwang, "Shared-distribution hidden Markov models for speech recognition", *IEEE Trans. Speech and Audio Processing*, Vol. 1, No. 4, pp. 414–420, Apr. 1993.

[65] F. Itakura, "Line spectrum representation of linear predictive coefficients", *J. Acoust. Soc. Am.*, Vol. 57, No. 4, p. 535, Apr. 1975.

[66] S.H. Jensen, P.C. Hansen, S.D. Hansen, and J.A. Srensen, "Reduction of broadband noise in speech by truncated QSVD", *IEEE Trans. Speech and Audio Processing*, Vol. 3, pp. 439–448, Nov. 1995.

[67] J.C. Junqua and J.P. Haton, *Robustness in automatic speech recognition: fundamentals and applications*, Kluwer Academic Publishers, USA, 1996.

[68] S. Kajita and F. Itakura, "Robust speech feature extraction using SBCOR analysis", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 421–424, May 1995.

[69] H. Kawahara, I.M. Katsuse and A.D. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous frequency based F0 extraction: Possible role of a repetitive structure in sounds", *Speech Communication*, Vol. 27, pp. 187–207, 1999.

[70] D. Kim, S. Lee, and R.M. Kil, "Auditory processing of speech signals for robust speech recognition in real-world noisy environments", *IEEE Trans. Speech and Audio Processing*, Vol. 7, No. 1, pp. 55–69, Jan. 1999.

[71] D. H. Klatt, "A digital filter bank for spectral matching", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 573–576, Apr. 1976.

[72] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 37, pp. 1641–1648, Nov. 1989.

[73] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", *Computer Speech and Language*, Vol. 9, pp. 171–185, Apr. 1995.

[74] J.S. Lim and A.V. Oppenheim, *Advanced topics in signal processing*, Prentice-Hall, 1988.

[75] J.S. Lim, "Spectral root homomorphic deconvolution system", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 27, No. 3, pp. 223–233, June 1979.

[76] J.S. Lim and A.V. Oppenheim, "Enhancement and bandwidth compression of noisy speech", *Proceedings of the IEEE*, Vol. 67, No. 12, pp. 1586–1604, Dec. 1979.

[77] P.H. Lindsay and D.A. Norman, *Human information processing*, Academic Press, New York and London, 1972.

[78] R.P. Lippmann, "An introduction to computing with neural nets", *IEEE ASSP Magazine*, Vol. 4, pp. 4–22, Apr. 1987.

[79] R.P. Lippmann, "Speech recognition by machines and humans", *Speech Communication*, Vol. 22, pp. 1–15, 1997.

[80] L. Liu, J. He, and G. Palm, "Effects of phase on the perception of intervocalic stop consonants", *Speech Communication*, Vol. 22, No. 4, pp. 403–417, Sept. 1997.

[81] P. Lockwood, C. Baillargeat, J.M. Gillot, J. Boudy, and G. Faucon, "Noise reduction of speech enhancement in cars: non-linear spectral subtraction - Kalman filtering", *Proc. European Conf. Speech Communication and Technology*, pp. 83–86, Sept. 1991.

[82] D. Mansour and B.H. Juang, "The short-time modified coherence representation and noisy speech recognition", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 37, No. 6, pp. 795–804, June 1989.

[83] G.A. Merchant and T.W. Parks, "Reconstruction of signals from phase: efficient algorithms, segmentation, and generalisations", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-31, No. 5, pp. 1135–1147, Oct. 1983.

[84] C. Mokbel and G. Chollet, "Word recognition in the car speech enhancement/spectral transformations", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 925–928, May 1991.

[85] Hema A. Murthy, K. V. Madhu Murthy, and B. Yegnanarayana, "Formant extraction from phase using weighted group delay function", *Electronic Letters*, Vol. 25, No. 23, pp. 1609–1611, Nov. 1989.

[86] Hema A. Murthy, K. V. Madhu Murthy, and B. Yegnanarayana, "Formant extraction from Fourier transform phase", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 484–487, May 1989.

[87] P. Satyanarayana Murthy and B. Yegnanarayana, "Robustness of group-delay based method for extraction of significant instants of excitation from speech signals", *IEEE Trans. Speech and Audio Processing*, Vol. 7, No. 6, pp. 609–619, Nov. 1999.

[88] H.A. Murthy and V. Gadde, "The modified group delay function and its application to phoneme recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 68–71, Apr. 2003.

[89] T. Nakatani, T. Irino, and P. Zolfaghari, "Dominance spectrum based v/uv classification and F_o estimation", *Proc. European Conf. Speech Communication and Technology*, pp. 2313–2316, Sept. 2003.

[90] S.H. Nawab, T.F. Quatieri, and J.S. Lim, "Signal reconstruction from short-time Fourier transform magnitude", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-31, No. 4, pp. 986–998, Aug. 1983.

[91] D.J. Nelson, "Cross-spectral methods for processing speech", *J. Acoust. Soc. Am.*, Vol. 110, No. 5, pp. 2575–2592, Nov. 2001.

[92] D.J. Nelson, "Cross-spectral based formant estimation and alignment", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 621–624, May 2004.

[93] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 417–420, Apr. 1994.

[94] Y. Neuvo, "Mobile future", *Proc. European Conf. Speech Communication and Technology*, pp. K7–K9, 2001.

[95] G. Nico and J. Fortuny, "Using the matrix pencil method to solve phase unwrapping", *IEEE Trans. Signal Processing*, Vol. 51, No. 3, Mar. 2003.

[96] A.H. Nuttall, "Some windows with very good sidelobe behaviour", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 1, pp. 84–91, Feb. 1981.

[97] G.S. Ohm, "Uber die Definition des Tones, nebst daran geknupfter Theorie der Sirene und ahnlicher tonbildender Vorrichtungen", *Ann. Phys. Chem.*, Vol. 59, pp. 513–565, 1843.

[98] M. Omologo, M. Matassoni, P. Svaizer, and D. Giuliani, "Microphone array based speech recognition with different talker-array positions", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 227–230, Apr. 1997.

[99] A.V. Oppenheim and R.W. Schafer, *Digitial signal processing*, Prentice-Hall, 1975.

[100] A.V. Oppenheim and J.S. Lim, "The importance of phase in signals", *Proceedings of the IEEE*, Vol. 69, pp. 529–541, May 1981.

[101] A.V. Oppenheim and R.W. Schafer, *Discrete-time signal processing*, 2nd ed., Prentice-Hall, 1999.

[102] J.P. Openshaw and J.S. Mason, "On the limitations of cepstral features in noise", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 49–52, Apr. 1994.

[103] D. O'Shaughnessy, *Speech communication: human and machine*, Addison-Wesley, 1987.

[104] F.J. Owens, *Signal processing of speech*, Macmillan publishers, 1993.

[105] K.K. Paliwal, "Decorrelated and liftered filter-bank energies for robust speech recognition", *Proc. European Conf. Speech Communication and Technology*, pp. 85–88, Sept. 1999.

[106] K.K. Paliwal and L. Alsteris, "Usefulness of phase spectrum in human speech perception", *Proc. European Conf. Speech Communication and Technology*, pp. 2117–2120, Sept. 2003.

[107] K.K. Paliwal and L.D. Alsteris, "On the usefulness of STFT phase spectrum in human listening tests", *Speech Communication*, Vol. 45, No. 2, pp. 153–170, Feb. 2005.

[108] K.K. Paliwal and B.S. Atal, "Frequency-related representation of speech", *Proc. European Conf. Speech Communication and Technology*, pp. 65–68, Sept. 2003.

[109] J.W. Picone, "Signal modeling techniques in speech recognition", *Proceedings of the IEEE*, Vol. 81, No. 9, pp. 1215–1247, Sept. 1993.

[110] H. Pobloth and W.B. Kleijn, "On phase perception in speech", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 29–32, 1999.

[111] M.R. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-24, No. 3, pp. 243–248, June 1976.

[112] M.R. Portnoff, "Magnitude-phase relationships for short-time Fourier transforms based on Gaussian analysis windows", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 186–189, Apr. 1979.

[113] M.R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 1, pp. 55–69, Feb. 1980.

[114] M.R. Portnoff, "Short-time Fourier analysis of sampled speech" *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No.3, pp. 364–373, June 1981.

[115] M.R. Portnoff, "Time-scale modification of speech based on short-time Fourier analysis" *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 3, pp. 374–390, June 1981.

[116] A. Potamianos and P. Maragos, "Speech formant frequency and bandwidth tracking using multiband energy demodulation", *J. Acoust. Soc. Am.*, Vol. 99, No. 6, pp. 3795–3806, Jun. 1996.

[117] A. Potamianos and P. Maragos, "Time-frequency distributions for automatic speech recognition", *IEEE Trans. Speech and Audio Processing*, Vol. 9, pp. 196–200, Mar. 2001.

[118] P. Price, W.M. Fisher, J. Bernstein, and D.S. Pallett, "The DARPA 1000-word resource management database for continuous speech recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 651–654, Apr. 1988.

[119] T.F. Quatieri and A.V. Oppenheim, "Iterative techniques for minimum phase signal reconstruction from phase or magnitude", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 6, pp. 1187–1193, Dec. 1981.

[120] T.F. Quatieri, *Discrete-time speech signal processing*, Prentice Hall, Upper Saddle River, NJ, 2002.

[121] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, Feb. 1989.

[122] L.R. Rabiner, M.J. Cheng, A.E. Rosenberg, and C.A. McGonegal, "A comparative performance study of several pitch detection algorithms", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 24, pp. 399–418, Oct. 1976.

[123] L.R. Rabiner and R.W. Schafer, *Discrete-time speech signal processing, principles and practice*, Prentice Hall, Englewood Cliffs, 1978.

[124] N.S. Reddy and M.N.S. Swamy, "Derivative of phase spectrum of truncated autoregressive signals", *IEEE Trans. Circuits and Systems*, Vol. CAS-32, No. 6, June 1985.

[125] G. Rigoll and D. Willett, "A NN/HMM hybrid for continuous speech recognition with a discriminant nonlinear feature extraction", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 9–12, May 1998.

[126] R.W. Schafer and L.R. Rabiner, "Design and simulation of a speech analysis-synthesis system based on short-time Fourier analysis", *IEEE Trans. Audio Electroacoustics*, Vol. AU-21, pp. 165–174, June 1973.

[127] M.R. Schroeder, "Models of hearing", *Proceedings of the IEEE*, Vol. 63, pp. 1332–1350, 1975.

[128] M.R. Schroeder, B.S. Atal, and J.L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear", *J. Acoust. Soc. Am.*, Vol. 66, No. 6, pp. 1647–1652, Dec. 1979.

[129] M.R. Schroeder and H.W. Strube, "Flat-spectrum speech", *J. Acoust. Soc. Am.*, Vol. 79, No. 5, pp. 1580–1583, May 1986.

[130] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing", *J. Phonetics*, Vol. 16, pp. 55–76, Jan. 1988.

[131] B. Shannon and K.K. Paliwal, "A comparative study of filter bank spacing for speech recognition", *Proc. Microelectronic Engineering Research Conf.*, Nov. 2003.

[132] V.C. Shields, Jr., "Separation of added speech signals by digital comb filtering", Department of Electrical Engineering, Massachusetts Institute of Technology, Sept. 1970.

[133] R. Smits and B. Yegnanarayana, "Determination of instants of significant excitation in speech using group delay function", *IEEE Trans. Speech and Audio Processing*, Vol. 3, No. 5, pp. 325–333, Sept. 1995.

[134] S.S. Stevens and J. Volkman, "The relation of pitch to frequency", *J. Psychology*, Vol. 53, pp. 329–353, 1940.

[135] S. Tamura and A. Waibel, "Noise reduction using connectionist models", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 553–556, Apr. 1988.

[136] J. Tchorz and B. Kollmeier, "A model of auditory perception as front end for automatic speech recognition", *J. Acoust. Soc. Am.*, Vol. 106, No. 4, pp. 2040–2050, Oct. 1999.

[137] V.T. Tom, T.F. Quatieri, M.H. Hayes and J.H. McClellan, "Convergence of iterative nonexpansive signal reconstruction algorithms", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-29, No. 5, pp. 1052–1058, Oct. 1981.

[138] J.M. Tribolet, "A new phase unwrapping algorithm", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-25, No. 2, pp. 170–177, Apr. 1977.

[139] P.L. Van Hove, M.H. Hayes, J.S. Lim, and A.V. Oppenheim, "Signal reconstruction from signed Fourier transform magnitude", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-31, No. 5, pp. 1286–1293, Oct. 1983.

[140] A.P. Varga and R.K. Moore, "Hidden Markov model decomposition of speech and noise", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 845–848, Apr. 1990.

[141] S.V. Vaseghi, B.P. Milner, and J.J. Humphries, "Noisy speech recognition using cepstral-time features and spectral-time filters", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 65–68, Apr. 1994.

[142] Y. Wang, J. Hansen, G.K. Allu, and R. Kumaresan, "Average instantaneous frequency and average log envelopes for ASR with the aurora 2 database", *Proc. European Conf. Speech Communication and Technology*, pp. 25–28, Sept. 2003.

[143] D.L. Wang and J.S. Lim, "The unimportance of phase in speech enhancement", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-30, No. 4, pp. 679–681, Aug. 1982.

[144] B. Widrow, "Adaptive noise canceling: principles and applications", *Proceedings of the IEEE*, Vol. 63, No. 12, pp. 1692–1716, Dec. 1975.

[145] B. Yegnanarayana and H.A. Murthy, "Significance of group delay functions in spectrum estimation", *IEEE Trans. Signal Processing*, Vol. 40, No. 9, pp. 2281–2289, Sept. 1992.

[146] B. Yegnanarayana, K.S. Reddy, and S.P. Kishore, "Source and system features for speaker recognition using AANN models", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 409–412, May 2001.

[147] B. Yegnanarayana, D.K. Saikia, and T.R. Krishnan, "Significance of group delay functions in signal reconstruction from spectral magnitude or phase", *IEEE Trans.*

Acoustics, Speech and Signal Processing, Vol. ASSP-32, No. 3, pp. 610–623, June 1984.

[148] B. Yegnanarayana and R. Smits, "A robust method for determining instants of major excitations in voiced speech", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 776–779, May 1995.

[149] B. Yegnanarayana, J. Sreekanth, and A. Rangarajan, "Waveform estimation using group delay processing", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-33, No. 4, pp. 832–836, Aug. 1985.

[150] B. Yegnanarayana, S. Tanveer Fathima, and H.A. Murthy, "Reconstruction from Fourier transform phase with applications to speech analysis", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 301–304, Apr. 1987.

[151] S. Young, *The HTK Book*, Cambridge University Engineering Department, Cambridge, England, 2001.

[152] D. Yuk, C. Che, L. Jin, and Q. Lin, "Environment-independent continuous speech recognition using neural networks and hidden Markov models", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 3359–3362, May 1996.

[153] D. Zhu and K.K. Paliwal, "Product of power spectrum and group delay function for speech recognition", *Proc. International Conf. Acoustics, Speech, Signal Processing*, pp. 125–128, May 2004.

[154] E. Zwicker and E. Terhardt, "Analytical expressions for critical band rate and critical bandwidth as a function of frequency", *J. Acoust. Soc. Am.*, Vol. 68, No. 5, pp. 1523–1525, Nov. 1980.

[155] "Transmission performance characteristics of pulse code modulation", *ITU-T Recommendation G712*, Sept. 1992.