ABSTRACT

Title of dissertation: SEGREGATION OF SPEECH SIGNALS IN NOISY ENVIRONMENTS

Srikanth Vishnubhotla

Dissertation directed by: Professor Carol Espy-Wilson
Department of Electrical and Computer Engineering

Automatic segregation of overlapping speech signals from single-channel recordings is a challenging problem in speech processing. Similarly, the problem of extracting speech signals from noisy speech is a problem that has attracted a variety of research for several years but is still unsolved. Speech extraction from noisy speech mixtures where the background interference could be either speech or noise is especially difficult when the task is to preserve perceptually salient properties of the recovered acoustic signals for use in human communication. In this work, we propose a speech segregation algorithm that can simultaneously deal with both background noise as well as interfering speech. We propose a feature-based, bottom-up algorithm which makes no assumptions about the nature of the interference or does not rely on any prior trained source models for speech extraction. As such, the algorithm should be applicable for a wide variety of problems, and also be useful for human communication since an aim of the system is to recover the target speech signals in the acoustic domain. The proposed algorithm can be compartmentalized into (1) a multi-pitch detection stage which extracts the pitch of the participating speakers, (2) a segregation stage which teases apart the harmonics of the participating sources, (3) a reliability and add-back stage which scales the estimates based on their reliability and adds back appropriate amounts of aperiodic energy for the unvoiced regions of speech and (4) a speaker assignment stage which assigns the extracted speech signals to their appropriate respective sources. The pitch of two overlapping speakers is extracted using a novel feature, the 2-D Average Magnitude Difference Function, which is also capable of giving a single pitch estimate when the input contains only one speaker. The segregation algorithm is based on a least squares framework relying on the

estimated pitch values to give estimates of each speaker's contributions to the mixture. The reliability block is based on a non-linear function of the energy of the estimates, this non-linear function having been learnt from a variety of speech and noise data but being very generic in nature and applicability to different databases. With both single- and multiple- pitch extraction and segregation capabilities, the proposed algorithm is amenable to both speech-in-speech and speech-in-noise conditions. The algorithm is evaluated on several objective and subjective tests using both speech and noise interference from different databases. The proposed speech segregation system demonstrates performance comparable to or better than the state-of-the-art on most of the objective tasks. Subjective tests on the speech signals reconstructed by the algorithm, on normal hearing as well as users of hearing aids, indicate a significant improvement in the perceptual quality of the speech signal after being processed by our proposed algorithm, and suggest that the proposed segregation algorithm can be used as a pre-processing block within the signal processing of communication devices. The utility of the algorithm for both perceptual and automatic tasks, based on a single-channel solution, makes it a unique speech extraction tool and a first of its kind in contemporary technology.

SEGREGATION OF SPEECH SIGNALS IN NOISY ENVIRONMENTS

by

Srikanth Vishnubhotla

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Carol Espy-Wilson, Chair/Advisor
Professor Shihab Shamma
Professor Rama Chellappa
Professor K J Ray Liu
Professor William J Idsardi

Dedicated to

My Family, for their constant love, support and understanding.

Acknowledgments

"Mathru Devo Bhava, Pithru Devo Bhava, Acharya Devo Bhava"

Mother is equivalent to God, Father is equivalent to God, and Teacher is equivalent to God.

I would like to express my undying and infinite gratitude towards my parents, who have given me everything I wished for, and have made me what I am. I owe everything to them, and this thesis is a dedication to them.

This is the best opportunity to express my sincere thanks and immense gratitude to my teacher and guide - Prof. Carol Espy-Wilson. She has been my inspiration and Guru during my work in the Speech Lab. She has taught me the philosophy and method of research, and has helped me shape my research career. I would especially like to thank her for the freedom she gave me in exploring various ideas... and for sportively tolerating my work schedule!

I would like to express my thanks to the members of my thesis committee, Prof. Rama Chellappa, Prof. Shihab Shamma, Prof. Ray Liu and Prof. William Idsardi for sparing their invaluable time in reviewing the manuscript of the thesis, and for providing valuable comment and suggestions. I have also had the pleasure and honor of learning from them on various matters related to and outside of this thesis, and am grateful for those opportunities.

My thanks to all my colleagues of the Speech Communication lab. Discussions with Om, Daniel, Tarun, Xinhui and Vijay have always proved to be fruitful and insightful. Tarun has been a very good friend and colleague, and discussions with him in various stages of my thesis have been very influential. I thank Om for all his help and suggestions at various stages of my research. Vijay has been very helpful in running various experiments and bouncing ideas with him has been fun and informative.

I would like to acknowledge the timely help and support from the ECE and ISR IT staff.

I have been extremely lucky to have a very long list of friends here at UMD, whom I must thank for the memorable years full of discussions, help and support. Abhinav Gupta, Anuj Rawat, Himanshu Tyagi, Pavan Turaga, Rahul Ratan, Ramya Chari, Ravi Tandon and Swati Jarial will always find a place among the best of my friends, and they have my ever-lasting gratitude for all

the wonderful times shared together. Thanks Jishnu Keshavan, Satish Chintakunta and Shilpa Billa for all the great times, card games, movies and wonderful dinners! Thanks to Aparna Kotha, Anshu Sarje, Harita Tenneti, Kapil Anand, Raghu Gopalan for the movies, road trips, get-togethers and various forms of craziness. Bargava Subramanian, Jyothi Patri, Sravya Kosaraju and Vinay Pulusu - thanks for being there, and for a lot more. It is impossible to remember all and I may have inadvertently left some of you, my dear friends - but please know that I am grateful and love you.

To my mom, dad and sister - thanks again for being so patient and loving. I love you and miss you all.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACF	Auto-Correlation Function
AMDF	Average Magnitude Difference Function
ANN	Artificial Neural Networks
ASR	Automatic Speech Recognition
BSS	Blind Source Separation
GMM	Gaussian Mixture Model
HMM	Hidden Markov-Model
ICA	Independent Component Analysis
MCMC	Markov Chain Monte Carlo
MFCC	Mel-Frequency Cepstral Coefficients
MMSE	Minimum Mean-Square Error
MSE	Mean-Square Error
PDF	Probability Distribution Function
PESQ	Perceptual Evaluation of Speech Quality
SACF	Summary Auto-Correlation Function
SID	Speaker Identification
SNR	Signal-to-Noise Ratio
STSA	Short-Time Spectral Amplitude
SVR	Support Vector Regression
TFU	Time-Frequency Unit
TMR	Target-to-Masker Ratio

Chapter 1

INTRODUCTION

1.1 Speaker Separation and Speech Enhancement from a Mixture of Speakers

Speech processing applications are usually focused on the problem of analyzing an electrical signal and understanding the information content from the speech of a particular speaker. For example, in a recording of a meeting consisting of ten speakers, it may be necessary to perform automatic speech recognition (ASR), and also label the speech as coming from a specific speaker among the ten present, i.e., speaker identification (SID). In such scenarios, where machines are required to perform the tasks of recognizing who spoke what, it becomes necessary to first separate the incoming mixture of various sources from each other, and then follow each individual speech stream with the relevant task. Furthermore, many real-world situations consist of speech signals which are usually interspersed with various forms of distortion, like noise from multiple sources as well as background speech from speakers other than the primary speaker. Thus, the task becomes more difficult, as there is also the need to enhance the quality of the speech by removing the background noise.

The task of separation of speech from noise or a background speaker has been looked at by researchers for several years, and most approaches focus on exploiting the mutual information arising from a set of sensors / microphones. The approach has been to capture the mixture speech (i.e, from multiple speakers) or noisy speech using more than one microphone, and then designing optimal weights to combine the outputs of these microphones, so as to maximize the quality of the reconstructed individual speaker's speech. Each microphone functions like an antenna with a beam in a certain direction, and these beams can be combined in certain ways. The weights of these various microphones are designed in such a way, that the collective beam of all the microphones is directed towards the primary speech of interest and the background signal escapes very weakly into this collective beam. Multi-microphone approaches of this kind are collectively called Optimal Antenna Beam-forming. An important drawback of this approach is that the direction of the target speech must be known beforehand, so that the antenna beam may be formed to point in the desired direction. Several other techniques fall under the category of Blind Source Separation (BSS), Independent Component Analysis (ICA), etc., and these methods work by exploiting the signals from multiple microphones to hypothesize signals which are statistically more independent from each other and presumably gave rise to the observed mixtures. For the BSS and ICA methods in practice, the number of sensors (microphones) must typically be equal to the number of speakers to ensure good separation of the speech signals. The most important drawback of all these multi-microphone approaches is that they rely on the presence of multiple sensors. However, information relevant to number of speakers (and thus, the number of microphones required for the task) is rarely known beforehand and even if so, it might be impractical to use more than one sensor for the specific application. In such cases, it becomes necessary to use a single speech mixture to separate the speech from multiple speakers.

In this thesis, we attempt to perform the tasks of separating speech emanating from different sources, as well as speech enhancement, i.e, the removal of background noise in the speech signal. Furthermore, we intend to perform these tasks using a single speech mixture (i.e., assuming only one microphone). The problem description is as follows: there is a single speech mixture available to the algorithm. There is one single "main" speaker whose speech we want to extract and separate

from a mixture of speech and background. The background could be either noise (wide-band or narrow-band) or competing speech (where there is another single "masker" speaker talking in the background). We limit ourselves to extracting the target speech from a mixture containing a maximum of *two* simultaneous speakers, and solve the case of more than two speakers by treating the situation as target speech in the presence of babble noise. In any case, we restrict ourselves to the Single-Channel problem.

While single-channel speech enhancement and separation of speech from speech mixtures (collectively refered to as "speech extraction" in the rest of this thesis) are also both well-studied problems, with various approaches that are devoted to certain salient properties of either the speech signal or the setting of the environment (locations of the microphones, etc.), either problem is still not yet fully solved. In particular, the goals of these various algorithms which "clean" the speech signals are often disparate and as such, it is very difficult to combine the useful aspects of these algorithms into one generic system. For example, there exist algorithms which are very well-adapted to perform automatic speech recognition or speaker identification in adverse environments (noisy speech or in the presence of competing speech), c.f. [21]. However, these algorithms do not yield cleaned speech signals in the acoustic domain. Instead, they rely heavily on exploiting the training process of the recognizer to deal with such adverse conditions by incorporating higher-level speech information like grammar and language models. As such are restricted in their application to the recognition domain - they do not yield a cleaned version of the speech signal in the acoustic domain. On the other hand, several algorithms exist which attempt to achieve a cleaner version of the noisy signal in the acoustic domain; however, such algorithms yield versions which are perceptually not of very good quality and furthermore, their performance for recognition tasks is far from the levels desired. Both these observations may be due to severe artifacts introduced by the cleaning algorithms while they remove the interfering signal. The problem is compounded by the fact that some of these algorithms are specifically designed to exploit the statistical nature of background noise, and hence are applicable only to the problem of speech enhancement (i.e., these enhancement approaches are unsuitable for segregation). Similarly, on the other hand, most of the algorithms that are designed to separate competing speakers are not very useful for the scenario of speech enhancement (i.e., these segregation approaches are unsuitable for enhancement).

This thesis aims to extend the state-of-the-art in both the enhancement and segregation domains. The main goal of this thesis is to design a system that separates speech from non-speech, as well as from other masker speakers. That is, we aim at designing an algorithm that can perform both speech separation (separating competing masker speech) and enhancement (separating background noise). Furthermore, we aim at restricting ourselves to the acoustic domain, i.e., to process the incoming speech signal and generate a cleaned (acoustic) version of the speech signal which can be played back over speakers. The motivation here is to generate the cleaned versions in the *acoustic* domain, with the reasoning that generating good acoustic outputs will increase the scope of the proposed algorithm for various applications. In addition to using the processed speech for automatic tasks like ASR and SID, the speech extraction algorithm will also be useful for remove interference over communication channels, and to improve the performance of devices which currently do not function well in the presence of noise (e.g., hearing aids and cochlear implants).

In the rest of this thesis, we will describe the algorithm as it is applied to either the problem of Speech Segregation or Speech Enhancement. Specifically, in certain sections, some of the content is discussed in reference to only one of the two problems, but general comments are also made to how the content is applicable to the other problem as well.

1.2 Previous Work on Speech Separation

The problem of separating the speech of speakers talking simultaneously in a single-channel speech mixture has been addressed since the late 1980s and is still an unsolved problem, especially for the requirement of separating the speech signals in the acoustic domain. Almost all of the prior

work reported in literature has focused on speech mixtures with no more than two simultaneous speakers (also the focus of this thesis). The various approaches that have been taken towards this problem can be classified broadly into two classes: the model-based or top-down approaches and the feature-based or bottom-up approaches (see [49] for detailed descriptions of these various approaches, and [7] for the performance of some of these approaches on the task of ASR on speech mixtures). In both approaches, the general procedure has been to first analyze the speech mixture at a given time instant or time-frequency region for its contents using some *features* like the well-known Mel-Frequency Cepstral Coefficients (MFCCs), Autocorrelation Function (ACF) etc. Following this, the parameterization features are then hypothesized to belong to either the target speaker or the masker speaker. The difference between both approaches lies in the method used to make this hypothesis, as explained below. Following this stage, the mixture signal at the time instants or time-frequency regions hypothesized as belonging to the target is used to reconstruct the target speech, while that hypothesized as belonging to the masker is used to reconstruct the masker speech.

In the model-based approaches, the speech mixture is modeled as a sum of signals arising from different sources in the presence of noise. The observation vectors are the features refered to above (usually the MFCCs), and the statistical distributions of these observation vectors are modeled using traditional statistical modeling tools like Gaussian Mixture Models (GMMs). Typically, models are trained for all the possible speakers expected in the mixture, and each individual speaker's model is created by collecting the features from speech signals having that speaker and a simultaneous competing speaker. During the actual segregation process, the likelihood of the observation vector of the current analysis frame is calculated conditioned on the model of each speaker, and the speaker whose model gives the maximum likelihood of generating that observation vector is hypothesized as having generated the speech in that analysis frame. If the speech signal is further decomposed into a number of frequency channels and the observation vector represents a Time Frequency Unit (TFU), this gives a spectro-temporal description of which source contributed which TFU to the mixture. Sometimes, in order to exploit the temporal continuity of the features or to impose temporal smoothness on the TFU-assignment process (which TFU contains features from which speaker), a Hidden-Markov Model (HMM) is also used to model temporal information. One of the limitations of the model-based approaches is that the algorithm requires prior training data to create models for the speakers, and this limits the generalization of the algorithm to a database which is different from the one used to train the speaker models. Furthermore, these methods do not contribute to the analysis of the speech signal and it is hard to make intuitive inferences from the outputs of the algorithm, especially if the segregation performance is bad and the parameters of the system need to be changed.

In the feature-based approaches, certain features of the mixture speech signal are used to analyze its content, and features which can potentially identify and separate the sources contributing to the mixture are then exploited to perform separation. Pitch, energy onset and offset locations, envelopes of auditory filter outputs, modulation patterns of the filter outputs in different channels, autocorrelations and cross-channel correlations of some of the features that have been used by researchers for the description and segregation of voiced regions [49]. Spectral slope, durations of the unvoiced regions etc. are some of the features used for the description of the unvoiced regions. Typically, these features are analyzed with the purpose of identifying which of the individual sources might have generated them, and this is done by comparing these features across time and frequency to identify which features "belong together" and which do not. Features which are very close to each other by some measure of similarity are hypothesized as having come from the same source / speaker, and are grouped together. Temporal and spectral continuity constraints are also incorporated during this assignment process. Spectro-temporal regions which gave rise to these features are accordingly labeled with their corresponding source identity. Advanced variants of these algorithms have been reported in the literature in recent years (c.f. [42]), and these rely on supplementary information, e.g. online speaker identification (SID) simultaneous to the grouping process to make the decisions more robust, especially in the unvoiced regions. Finally, as in the model-based approaches, input speech from the time frames or spectro-temporal regions hypothesized as coming from the same speaker is used to reconstruct the speech from that speaker. The

advantage of these methods over the model-based methods is that their segregation performance is more consistent across multiple databases and therefore they are more widely applicable.

As can be seen, both the classes of approaches described here function by dividing the input mixture signal into regions where one of the speakers dominates over the other, with the hypothesis that the features in those regions should closely match those of the dominating speaker. As such, the input signal is practically split into two non-overlapping streams, presumably the speech signal of each speaker. However, in reality, there do exist several spectro-temporal regions wherein the two speakers may be equally strong and wherein the allotment of the spectro-temporal region to only one speaker will not be the optimal decision. In fact, in such cases, the current approaches would yield output speech signals wherein one of the speech outputs would have a significant amount of leakage from the other speaker, while the other speech output would have a "hole" or silence wherever this phenomenon occurs. As such, the current approaches yield speech signals which are perceptually not very favored by listeners as they contain artifacts due to such leakage and holes. In the segregation algorithm proposed in this thesis, we develop a method to spearate out the speech signals even in those regions where both speakers are equally strong, thus reducing the artifacts. As such, the speech outputs obtained by the proposed algorithm should be perceptually more preferable than those obtained by the current approaches. We show this is indeed the case by the use of objective measures to compare the proposed algorithm with current segregation approaches.

A lot of the techniques that fall under the feature-based category rely explicitly or implicitly on the pitch of each individual source to perform speech separation. A variant of the feature-based algorithms which rely on the pitch is the harmonic-based algorithms which has two variants: harmonic enhancement and harmonic suppression. In both cases, a pitch estimate is obtained first by one of the classical single-pitch tracking methods, and this estimate is assumed to be the pitch of the "dominant" speaker. Following this step, the harmonic enhancement technique involves reducing the relative strength of the background or interference by boosting the harmonics of the dominant pitch using a harmonic filter. On the other hand, the harmonic cancellation method involves suppressing the dominant speaker and boosting the non-dominant speaker, by using a harmonic filter that has its zeros located at the harmonics of the dominant pitch. Variants of both these methods have been reported in the literature since the late 1980s [53, 9]. A major research question in the usage of such algorithms is the design of efficient time-varying filters that perform the required function effectively - such filters must be time-varying (pitch-dependent) and be accurate in removing or enhancing harmonics, while keeping the other frequencies intact. Filters which satisfy such tough criteria are hard to design, and as such the harmonic-based methods have not gained much popularity. Some methods approach the problem of designing such filters in the time-domain. In this thesis, it will be shown that the proposed algorithm is a generalization of such approaches, and that the harmonic based approaches perform filtering in a non-optimal way while the proposed algorithm filters the harmonics in an optimal way with respect to the mean-squared error.

1.3 Previous Work on Speech Enhancement

Speech enhancement refers to improving the signal-to-noise ratio (SNR) where the noise can be babble (background speech of multiple talkers), sound emanating from machines like vehicles etc., or improving the quality of speech which is corrupted by a change or nonlinear disturbances in transmission medium. The problem of speech enhancement has also received a tremendous amount of research attention over the past several decades, and the approaches towards the problem can also be classified into two categories: statistical or model-based approaches and feature-based approaches.

A bulk of the statistical approaches includes the speech enhancement techniques based on modifying the Short-Time Spectral Amplitude (STSA) [29] of the noisy speech signals. The tech-

niques based on subtractive type algorithms (c.f. [17]) assume that the background noise is locally stationary to the degree that noise characteristics computed during the speech pauses are a good approximation to the noise characteristics during the speech activity. Thus, an estimate of the noise spectrum is made in the speech pause regions, and this noise spectrum is then subtracted in the speech-present regions to yield a relatively less noisy speech signal. In addition to the basic spectral subtraction algorithm, several extensions and improvements have been proposed by incorporating variations in the subtraction parameters like the over-subtraction factor (subtracting an amplified version of noise spectrum), the spectral flooring factor (thresholding the noise spectrum to be subtracted) as well as the intelligibility of the enhanced speech (by designing an algorithm that adapts the subtraction parameters in time and frequency based on the masking properties of the human auditory system). It has been shown that under certain assumptions about the spectral characteristics of the speech signal and the noise, the spectral subtraction method is the maximum likelihood estimator of the variance of the speech spectral components [17].

[17] proposed a system that utilizes the minimum mean square-error short-time spectral amplitude MMSE-STSA estimator to enhance speech signals. This method assumes that each of the Fourier expansion coefficients of the speech and of the noise process can be modeled as Gaussian random variables with zero mean. Moreover, it is also assumed that these coefficients are independent of each other. The quality of the enhanced speech is better using a version of the MMSE estimator that takes into account the speech presence uncertainty. The residual noise is perceived more as white noise than as musical noise and is attributed to the smooth variation of the *a priori* signal-to-noise ratio estimate. The MMSE-STSA algorithm was extended by [16] to compute the STSA estimator that minimizes the mean-square error of the log-spectral amplitude, which is a more relevant criterion for perceivable distortions in speech. [30] replaced the squared-error cost function used in the MMSE estimator by perceptually more relevant cost functions that take into account the auditory masking effects. All of these speech enhancement methods make various restricting assumptions about the temporal and spectral characteristics of the speech signals and the corrupting noise. The performance of some of these methods deteriorates when the speech signals are corrupted by fluctuating noise.

Feature-based speech enhancement techniques include those based on models of the human auditory system, as well as those that extract relevant features from the speech signals and exploit their behavior to enhance noisy speech signals. The auditory based systems usually try to mimic the processes known to be involved in human audition, to separate sounds exhibiting speech like characteristics from those not doing so. The latter kind of systems employ signal processing techniques to estimate whether each TFU in the analysis corresponds to a speech signal or a non-speech signal, using various criteria for making decisions. For example, the Monaural Speech Separation (MSS) technique by [23] uses the pitch estimate as its feature, to identify if each T-F unit exhibits pitch or modulation behavior matching the pitch estimate, and accordingly makes a decision of whether the TFU contains a speech signal or noise. The Modified Phase Opponency (MPO) technique [13] uses the frequency behavior of the TFU as its feature (i.e., narrow band versus wide band) to decide whether the signal within the TFU came from a speech or non-speech source. These methods are parallel to the feature-based speech segregation methods in the sense that both methods attempt to decompose the input signal into two components - a target speech signal and a masker speech or noise signal. Both the feature-based enhancement and segregation methods perform this operation by making a decision of whether a particular TFU belongs to the target speech or not (though the criteria for such decisions may be different for different approaches). Thus, they both suffer from the issues discussed in the section 1.4.

1.4 Current Speech Segregation Approaches And Their Limitations

As has been discussed in sections 1.2 and 1.3, most current speech separation algorithms perform segregation by identifying spectro-temporal regions of speech (i.e., TFUs) where one of the

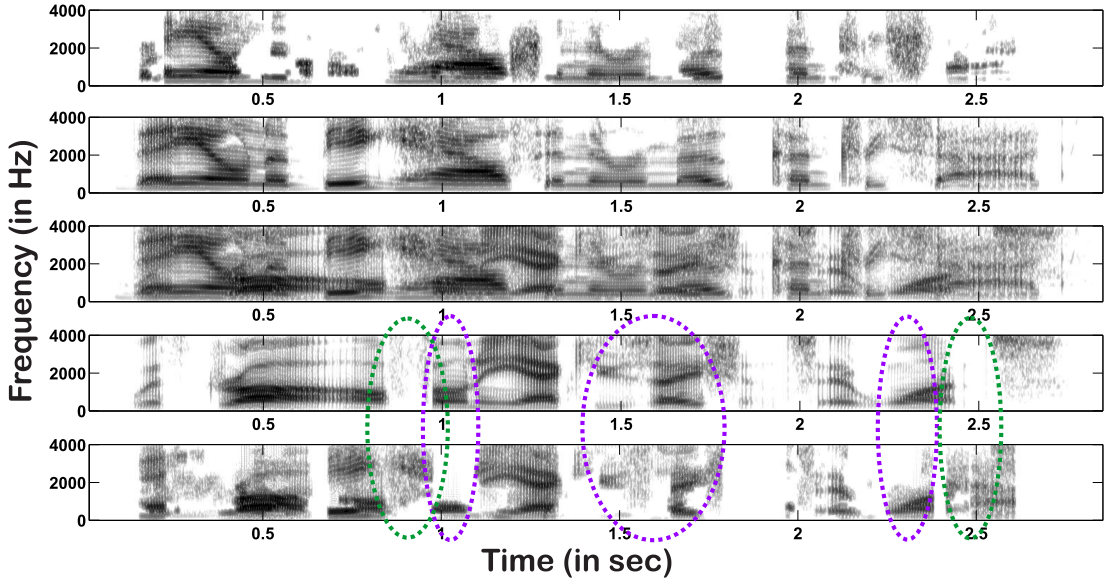

Figure 1.1: Illustration of the performance of a speech segregation system [42]. Middle Panel: Spectrogram of mixture signal containing speech from speakers A and B, Second Panel: Spectrogram of signal from speaker A, Fourth Panel: Spectrogram of signal from speaker B, First Panel: Spectrogram of signal from speaker A as recovered by [42], Fifth Panel: Spectrogram of signal from speaker B as recovered by [42]

speakers dominates over the other, and then separate the streams out by assigning each TFU to the appropriate source. The various algorithms differ by the method used to label TFUs according to source, but the overall segregation strategy is similar across the methods. One of the major problems with this strategy is that it causes a high proportion of "leakage errors" and "missed speech". This can be explained with the aid of Figs. 1.1 and 1.2.

Fig. 1.1 shows a spectrogram of a mixed speech signal containing speech from a female (speaker A) and male (speaker B) speakers, along with the segregated streams as obtained by one of the current segregation algorithms [42]. The middle panel shows the spectrogram of the mixture signal, with the second and fourth panels showing the spectrograms of the original clean speech signals that were combined to obtain the mixture. The first and fifth panels show the spectrograms of the speech signals as recovered by [42], which are supposed to be the segregated versions of the signals in the second and fourth panels respectively. As can be seen clearly, there are several spectro-temporal regions in both signals A and B, wherein the signal was not recovered well by the segregation algorithm. In particular, the recovered signals show clear spectro-temporal "holes" or missed speech, where the original speech signal had energy while the recovered speech signal is silent. Few such instances have been highlighted in magenta in the fifth panel. Similarly, there also are regions in the recovered speech signals where the output is non-zero, even though the original speech signal was silent - these are the "leakage" errors. Due to these leakage errors and missed speech, the final reconstructed speech signals are far from ideal perceptually, and the distortions introduced by these errors are significantly debilitating for most applications. While this problem has been illustrated for the algorithm of [42], it may be noted that these distortions are actually existent for most of the current speech segregation algorithms, which function by allocating spectro-temporal regions to only one speaker during separation. Thus, most of the current model-based and feature-based algorithms would suffer these kinds of distortions.

The reasons for these leakge and missed speech errors can be understood by understanding the principle behind these segregation algorithms. The limitation of these methods is that by virtue of the analysis performed by these algorithms, there can exist no T-F units where both speakers are present in the output. Thus, even if the mixture signal contains speech from both

speakers simultaneously in a spectro-temporal region, such regions are allotted only to a single, more dominant speaker. TFUs which in reality contain speech from both speakers are forced to be labeled as coming from only a single speaker, and are made to contribute to the reconstruction of only one speaker (even if they contained speech energy from both). If such TFUs are used for reconstruction, then the resultant speech for one speaker (say A) would contain extra speech coming in from the other speaker (say B) in such frames (the so-called Leakage Error in A) and the resultant speech of the other speaker (B) would lose this region of speech in reconstruction (the so-called Missed Speech in B). Therefore, one of the major issues of the current speech separation algorithms is that they tend to label T-F units as "dominated", rather than "shared" units. This causes the reconstruction to be faulty. They attempt to "divide" the mixture signal among two sources, rather than segregate it.

Fig. 1.2 shows the same information as in Fig. 1.1, but with some additional information. The middle panel once again shows the mixture spectrogram. The second and fourth panels show the spectrograms of the original clean speech coming from speakers A and B respectively. The spectro-temporal regions where the energy of A is greater than that of B (i.e., where A is the dominating speaker) are shown in blue on the second panel. Similarly, the spectro-temporal regions where B dominates over A are shown in red in the fourth panel. Since the existent algorithms try to assign TFUs in terms of which speaker dominates where, it is these *latter* two profiles (i.e., the blue and red spectro-temporal regions) that they aim at recovering accurately. These are the so-called Ideal T-F (ITF_{DOM}) maps. Following this, those TFUs labeled as blue are directly used (meaning, not processed to remove any of the weaker signal) to reconstruct speech from speaker A, and those labeled red are directly used to reconstruct speech from speaker B. TFUs which in reality contain speech from both speakers are forced to be labeled as coming from only one speaker, and are made to contribute to the reconstruction of only one speaker (even if it contained speech energy from both). As can be seen, there are significant spectro-temporal regions of the speech signals that these ITF_{DOM} maps fail to cover, and which the segregation algorithms can never hope to recover. More serious is the observation that these ITF_{DOM} will vary with the relative strengths of the speech signal and the background interference. In case the target speech from B is weak and the interference from A is strong, the ITF_{DOM} of B will be very sparse and not help recover much of the original speech signal of B, as is the case in Fig. 1.2. This susceptibility of the ITF_{DOM} to the interference is another major factor that renders these algorithms far from successful. Examples of regions where the ITF_{DOM} does not cover the speech signal completely and leaves out some missed speech are shown in magenta circles in the fourth and fifth panels.

The first and fifth panels show the spectrograms of the actual reconstructed speech signals, along with the above ITF_{DOM} maps overlaid as in the earlier case. As can be seen, the reconstructed speech signals closely match the ITF_{DOM} maps in most locations (especially for A), which confirms the above statement that the current segregation algorithms are successful only in recovering the dominant regions. Furthermore, the reconstructions also contain speech in regions not indicated by the respective ITF_{DOM} maps - however, upon closer observation, it can be seen that such regions very closely match the ITF_{DOM} of the *other* speaker. Two such cases are shown in green circles in the fifth panel of the Fig. 1.2. This should be expected because it is a consequence of the fact that the other (masker) speaker is dominant in such regions, and so the energy from there leaks into the reconstruction of the target speaker. The reconstructed speech for speaker A might thus contain extra speech coming in from speaker B (Leakage Error) and the reconstructed speech of speaker B would lack this region of speech (Missed Speech). Notice that this error is not due to any inability of the separation algorithm to recover the ITF_{DOM} masks - in fact, it is due to the inherent inability of the ITF_{DOM} masks to describe the original signals accurately. Indeed, the ground truth that these algorithms aim to achieve are actually incomplete representations of the speech signals to be recovered. Therefore, the tendency to label TFUs as "dominated", rather than "shared" causes reconstruction to be percpetually not very preferable.

Implications of this observation are numerous. (1) For the case of the higher frequency channels used in typical segregation systems, since they are usually based on the gammatone filterbank and thus have a larger bandwidth, it is highly unlikely that the signal content in such TFUs would be from a single source. Thus, the assignment of high-frequency TFUs to only one

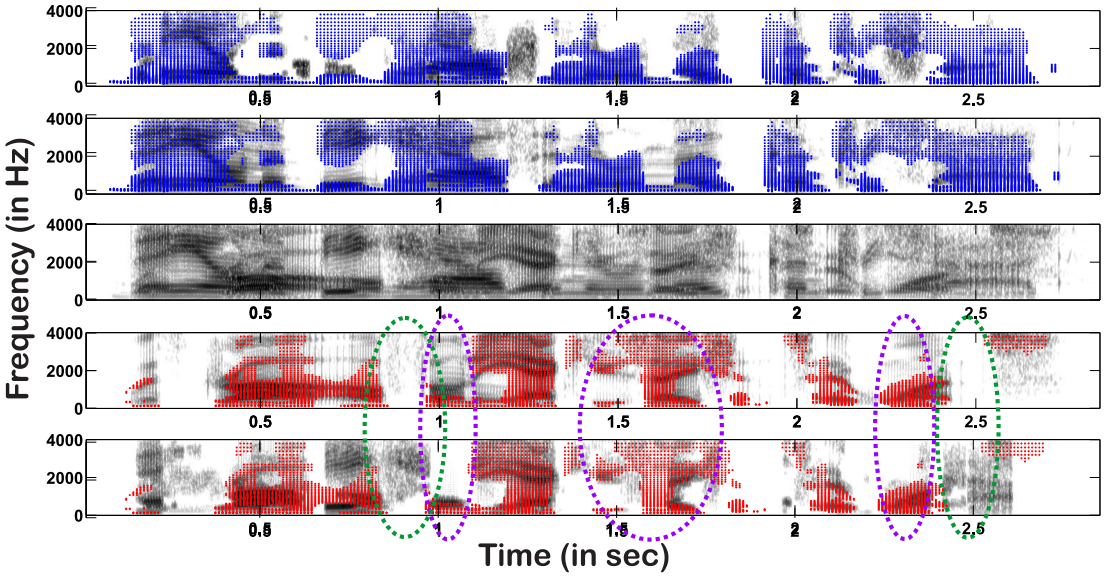

Figure 1.2: Explanation of the performance of a speech segregation system [42]. Middle Panel: Spectrogram of mixture signal containing speech from speakers A and B, same as the one shown in 1.1, Second Panel: Spectrogram of signal from speaker A, with the regions where speaker A dominates over speaker B overlaid in blue, Fourth Panel: Spectrogram of signal from speaker B, with the regions where speaker B dominates over speaker A overlaid in red, First Panel: Spectrogram of signal from speaker A as recovered by [42], with blue again representing regions where A dominates over B Fifth Panel: Spectrogram of signal from speaker B as recovered by [42], with red again representing regions where B dominates over A

speaker is especially inaccurate, and segregation based on this strategy will not achieve accurate reconstructed streams. (2) In the case where a target speech signal of interest is weaker compared to the interfering masker speech signal (i.e., the Target-to-Masker Ratio, or TMRm is less than 0 dB), a significant portion of the target TFUs would be dominated and therefore irrecoverable by the current segregation approaches. In essence, if the algorithm needs to track a specific target speaker but the target is located farther from the microphone than any masking sounds, the hope of fully recovering the target is low. (3) Users of cochlear implants typically are unable to separate the individual sources in an environment containing multiple sources [43, 26]. A segregation algorithm based on the domintated-TFU principle can only provide dominant regions of the target and therefore might not provide them with sufficient information to understand the target stream.

In our work, therefore, we try to introduce the philosophy that it is necessary and indeed inescapable in practical applications to "distribute" rather than "divide" the signal content among the individual speakers to achieve proper speech separation. We propose that instead of estimating the ITF_{DOM}, an alternate solution is to estimate *all the non-silent regions* of both utterances, what we call the Complete Ideal T-F mask ITF_{COM}. As is obvious, if the energies in each TFU of both signals are estimated accurately, a reconstruction based on the ITF_{COM} would be more accurate than one based on ITF_{DOM}.

Missing Feature Theory [6] is, in fact, an attempt to bridge the gap between the ITF_{DOM} and ITF_{COM} by proposing interpolations at the unknown regions of the ITF_{DOM} in a way as to be consistent with the known regions. We use a different approach to generate the ITF_{DOM}. Using a least-squares model that attempts, for every TFU, to accurately estimate the constituent signals of the mixture, we reconstruct each of the two participating streams by combining the estimates across all TFUs. Thus, while previous attempts at segregation assign TFUs to either of the two speakers, we attempt to tease out and correctly assign the contributions of each speaker in each TFU.

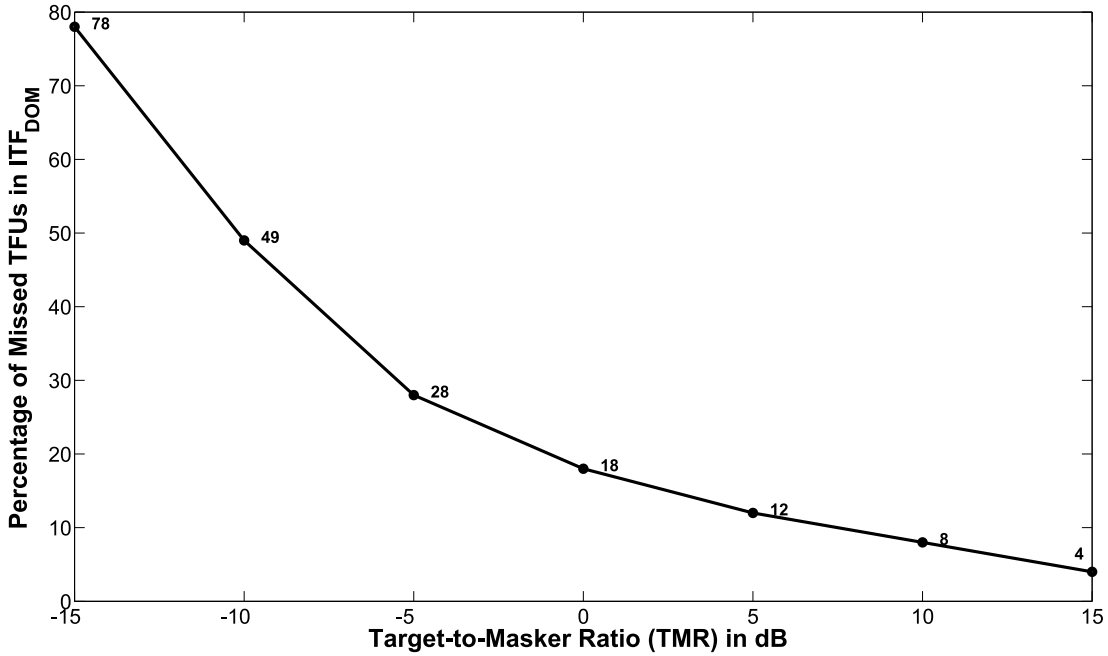

Figure 1.3: Plot showing the percentage of target speech energy lost by using the ITF_{DOM} representation of the target stream. At low TMRs, a significant portion of the target stream is not even attempted to be segregated since it doesn't fall into the ITF_{DOM} representation.

While it has been argued in the literature [28] that the labeling of TFUs as dominated instead of shared does not reduce reconstruction quality significantly, we have found in our analysis that this method of separating speech is often indequate, especially when the TMR is near zero or lower. Fig. 1.3 illustrates this observation. As was mentioned, the ITF_{DOM} does not completely describe the target since it does not represent regions of the mixture which were dominated by the masker. Even if the ITF_{DOM} is recovered accurately, the target signal is not close to the original, since some of the original non-silent TFUs of the target were not even present in the ITF_{DOM}. In this figure, the effect of dominated versus shared is explored in a quantitative method. The target and masker signals are added at different TMRs ranging from 15 dB to -15 dB. For the figure, the ITF_{COM} was generated by finding all regions of the original target speech signal which had energy exceeding a certain threshold. The ITF_{DOM} was generated by finding all regions of the original target signal which had energy greater than the original masker signal. Ideally, the ITF_{DOM} should be equal to the ITF_{COM} for all TMRs but due to the concept of "dominated" TFUs, it is not so. For each TMR, the energy of all the TFUs present in the ITF_{COM} but absent in the ITF_{DOM} represents the energy that the ITF_{DOM} representation would fail to represent and would be completely lost from the target. The percentage of energy lost by the ITF_{DOM} representation is shown in Fig. 1.3. As can be seen, at TMR of 0 dB, 18% of the target speech energy is completely lost and irrecoverable by current algorithms. At -5 dB, 28% or more than one-fourth of target speech energy is lost from the reference mask; at -15 dB, nearly 80% of the target energy is lost and irrecoverable.

From Fig. 1.3, the statement is supported that if T-F units are indeed labeled as dominated instead of shared, that would mean that a lot of significant speech information would be lost in the reconstruction process, which is a bad decision to take especially if both sources are almost equally strong. Furthermore, for the case of the higher channels which have a larger bandwidth (and thus, for example, can support multiple harmonics coming from two simultaenous speakers), it is highly unlikely that the signal content in that T-F unit must be from a single source. Thus, it is necessary to analyze each T-F unit and "distribute" the signal content among the individual speakers in order for us to achieve proper speech separation. This motivates our approach of

including a class of "shared" speech.

Another limitation of the current speech separation techniques is that they do not seem to be generic enough to separate more than two speakers. The existent multi-pitch algorithms are based on a theory that is increasingly hard to generalize to multiple speakers. The autocorrelation method will not be significantly effective in estimating the pitch for three speakers, as has been demonstrated in 2. Furthermore, the concept of assigning speech portions using the concept of dominant TFUs will render the three-speaker case even harder to deal with, and can cause significant distortions in the reconstructed speech signals due to the dynamics of energy dominance among the three competing speakers. However, we have tried to develop a speech segregation method that can be modified easily to extend to the case of three simultaneous speakers. The algorithm relies on the pitch estimated from the algorithm described in 2, which also can be potentially extended to work for the three speaker case. As such, the proposed segregation system is better suited for truly segregating multiple speakers.

Thus, summarizing, there are various factors that motivate us to look for a new approach to the segregation problem:

1. There are two sources of error that can effect performance of current segregation algorithms: error due to the reference used (ITF_{DOM} versus the more complete ITF_{COM}), and error in estimation of that reference.

2. There is no proposed approach to separate the streams when both sources are almost equally strong. In essence, the mixture spectrogram is broken into a number of fragments, with each fragment going to one of the speakers. We focus on peeling off layers of the mixture spectrogram, where each layer represents one speaker.

3. Difficulty in generalization of current algorithms to more than 2 speakers.

It must be noted that these limitations are generic to all the speech segregation methods which rely on identifying TFUs as belonging to only one of the speakers in the mixture and allocate spectro-temporal regions to only one speaker, as opposed to sharing the energy in such regions among both speakers. Furthermore, the discussion in this section with respect to feature-based techniques is also equally applicable to speech enhancement techniques which rely on making binary decisions of whether the TFUs belong to speech or not, e.g. [23]. The concept of dominated versus shareed TFUs has significant consequences both for speech segregation and speech enhancement, which is one of the motivations why the proposed speech extraction algorithm attempts to share the energy between the different acoustic sources, rather than make binary decisions regarding the dominant source.

1.5 Proposed Approach To The Problem

We propose to develop a single-channel speech separation algorithm that can separate a target speech signal both from interfering masker speech or background noise. The proposed algorithm is based on a combination of feature-based and model-based approaches. In particular, we use a feature-based technique to estimate the pitch of each of the individual components of the speech mixture. Following this, we try to fit a model to one of the features of the speech signal, namely its spectrum. The pitch-estimation process is primarily based on using temporal information, while the segregation process is based on modeling spectral information using a time-series. In order to make the pitch estimation process robust, and also to develop the flexibility of dealing with individual spectro-temporal regions, we process the signal by first decomposing it along a number of frequency channels and processing each *channel* on a frame-wise basis. The algorithm performs segregation by modeling each TFU as a combination of complex sinusoids of frequency corresponding to the pitch of both participating speakers. Thus, it depends on the knowledge of the pitch tracks of both speakers. The contributions of the two speakers are obtained

by fitting a Least-Squares model to the observed mixture signal within the TFU. It should be noted that our algorithm is significantly different from previous segregation attempts using sinusoids [37, 9] since the latter model estimate only the amplitudes of the participating sinusoids, while our algorithm directly models the time series using sinusoids and estimates both the amplitudes and phases. Further, spectral based methods are susceptible to the effects of windowing as discussed in Chapter 3, and furthermore, recovery of phase information of the two streams is very difficult from spectral methods. However, our algorithm models the signal in the time domain. Indeed, one of the advantages of the proposed model is that it is also able to accurately recover the phase information of both speakers, which was a limitation of previous sinusoid-based models. It may be noted that while the description given below seems applicable to the problem of speech segregation, it can be easily used for the purpose of speech enhancement as well. In the case of speech enhancement, the pitch estimate for the second speaker is assumed to be uniformly zero, which yields only one target speech signal estimate.

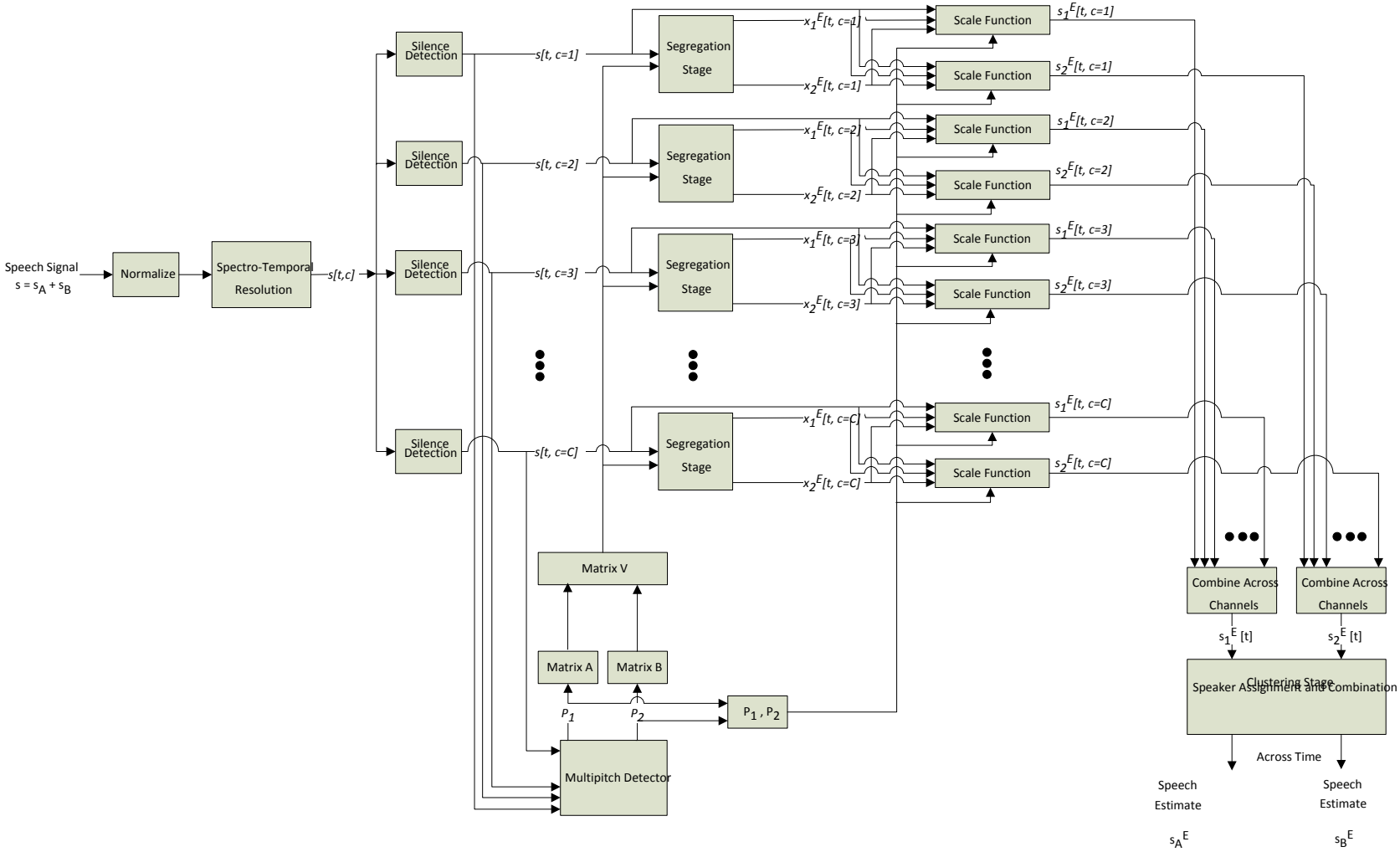

Figure 1.4: Block diagram of the proposed algorithm for speech segregation

In the remainder of this thesis, the label A is used to refer to one of the speakers participating in the speech mixture to be separated (the target) and the label B is used to refer to the other speaker (the masker). Fig. 1.4 shows the overall block diagram of the proposed algorithm, with all the individual internal blocks as well as their interconnections clearly shown. The input speech signal, called s, which is the sum of two individual (and unknown) speech signals s_A and s_B, is first normalized to have zero mean-value and unit variance or power (this step is not significant, as long as the power is within a reasonable range of values). The sampling rate of this input speech signal is called as F_s. Following normalization, the signal is decomposed into a number of channels or frequency regions using a filterbank, and is processed at a particular frame rate. We prefer to use a filterbank which has (near) perfect-reconstruction properties, since the signals from all these channels would later be added together to yield the two final segregated speech (acoustic) signals and therefore, the filterbank must introduce minimal possible distortions. By this point, the input speech signal is decomposed into a number of time units (frames) and frequency units (channels). The next block helps in identifying which of these time-frequency units ($s[t, c]$ to represent time t and channel c, hereafter referred to as TFUs) are silent, or which are non-silent. The latter potentially contain speech from either one or both participating speakers A and B, and only these will be analyzed in all the following blocks from this point. At a given time instant, the signals from all channels $s[t, c = 1, 2, .., C]$ (i.e., fixed time t, all frequency-channels $c = 1, 2, .., C$ where C is the total number of channels) are used together in the multi-pitch detector block, to find the pitch frequency estimates of both speakers A and B. The multi-pitch detector is described in full detail in Chapter 2. The pitch estimates yielded are labeled P_1 and P_2, both frequency values in Hz (it is still not known at this point which pitch frequency belongs to which speaker). These pitch estimates are then used in the block labeled "Segregation Stage", along with the sample frequency F_s to separate out the two overlapping signals within that TFU. This segregation block models the input signal as a sum of complex exponentials corresponding to the harmonics of both pitch frequencies, and tries to fit a Least-Squares model to this observed mixture in terms of the complex exponentials. The contributions of the individual speakers are obtained by solving the Least-Squares equations. Each frequency channel has its own segregation block, and so there are C such blocks in all. The speech signal from each TFU, $s[t, c]$ is sent to its corresponding segregation block (i.e., the c^{th} segregation block), and this c^{th} segregation block yields three different signals, namely $x_1^E[t, c]$ (the estimate corresponding to pitch P_1), $x_2^E[t, c]$ (the estimate corresponding to pitch P_2) and $x^E[t, c]$ (the estimate corresponding to the total input signal $s[t, c]$ as estimated by the proposed model). Thus, at this point, the input signal has been decomposed into two components, each corresponding to one of the two speakers. It may be noted that these segregated components are only the *voiced* components of the speech from A and B. These blocks belong to the Segregation Stage of the algorithm, and are described in full detail in Chapter 3. The obtained voiced estimates, however, may be unreliable since they may contain interference from the competing voiced source, or background noise. Thus, a scaling function is used to modify the estimates appropriately, so that reliable estimates are given more weight and unreliable noisy estimates are given less weight in the final segregated output. The scaling function is calculated in the block labeled "Scaling Function", to yield signals $s_1^E[t, c]$ and $s_2^E[t, c]$ corresponding to $x_1^E[t, c]$ and $x_2^E[t, c]$ respectively. In exceptional cases where the scaled estimates of the voiced components may be much weaker than the background noise, the scaling function may not be sufficient to eliminate the interference and it may be more preferable to simply discard the estimates and not use them. The block labeled as "Switch" identifies the reliability of each scaled estimate, and decides if the signal $s_1^E[t, c]$ (or $s_2^E[t, c]$) should indeed be used for final reconstruction or not. In case $x_1^E[t, c]$ (or $x_2^E[t, c]$) is found to be unreliable by the afore-mentioned switch, the signal $s_1^E[t, c]$ (or $s_2^E[t, c]$) is uniformly attenuated by a large number (20 dB in the rest of this thesis). In spectro-temporal regions where one or both the speakers are unvoiced, the unvoiced components also need to be estimated. This is done by relying on the estimates obtained from the voiced components, as well as the scaling functions. The procedure of performing the scaling, as well as obtaining the unvoiced components, is described in Chapter 4. Once the signals $s_1^E[t, c]$ and $s_2^E[t, c]$ are estimated for a given t and for all channels $c = 1, 2, .., C$, they are both combined together by synthesis filters to yield the total signals across all channels and for a given time instant, namely,

$s_1^E[t]$ and $s_2^E[t]$. Following this, the sequential grouping block analyzes these two signals for certain features, and comparing them with features evaluated in the past (stored in the delay elements), makes a decision of which of $s_1^E[t]$ or $s_2^E[t]$ belongs to A and which to B. Currently, the features being used for making the speaker assignments are the MFCCs of the two component signals, but the future work following this thesis will include exploring other more reliable fetures. Once the appropriate labeling is done by this grouping block, the signals $s_A^E[t]$ and $s_B^E[t]$ are obtained as the estimates of speaker A and B respectively, for a given time instant t. The algorithm to perform appropriate assignment of speakers is described in [31] and briefly at the end of Chapter 4. These estimates are then combined with the previous estimates of speaker A and B from the past time instants $t-1$, $t-2$, ... etc. by the well-known overlap-add method. Thus, the overall individual speech signal estimates s_A^E (estimate of S_A) and s_B^E (estimate of S_B) are obtained.

As mentioned above, in case of the problem of speech enhancement the multi-pitch algorithm will yield a single pitch estimate, and this pitch estimate can be used to extract the speech of the relevant target speaker.

1.6 Outline of the Thesis

The rest of this document describes the various blocks of the proposed segregation system, as well as the experiments used to test their quantitative performance on various tasks. Chapter 2 describes the multi-pitch detection algorithm that yields the pitch estimates for the segregation system. Chapter 3 describes the speech segregation model for the estimation of the voiced components of both speakers, as well as compares the proposed approach to other closely related works in the literature. Chapter 4 describes how the estimates of the voiced components are scaled appropriately, to yield perceptually better acoustic outputs. This chapter also describes the use of the voiced estimates for recovering the unvoiced regions of the individual speech signals. Finally, the chapter briefly explores the speaker assignment problem. Chapter 5 describes the evaluation of the proposed speech segregation system on several tasks including both objective and subjective metrics, as well as comparison with other systems proposed in the literature. Finally, chapter 6 summarizes the conclusions drawn from this thesis, and broaches on the future work that follows from the work described herein.

Chapter 2

MULTI-PITCH TRACKING

2.1 Introduction

This chapter describes a new multi-pitch detection algorithm that is designed to estimate the pitch of two participating speakers in a speech mixture. The algorithm relies on a new feature developed in this thesis, called the 2-D Average Magnitude Difference Function (AMDF), and on certain features extracted from the 2-D AMDF, to extract the periodicity information of the two speakers. The proposed algorithm has been shown to yield robust pitch estimates, even in low Target-to-Masker Ratios (TMRs) [46]. We begin by first exploring previous work on multi-pitch detection, highlighting some of their limitations to motivate the need for the proposed algorithm.

2.2 Previous Work On Multiple Pitch Tracking

The problem of multiple pitch tracking has been addressed since the early 1980s, and a number of algorithms relying on various acoustic features have been pursued for the purpose. These algorithms can be broadly classified into the model-based and feature-based algorithms.

The model-based approaches are an outcome of the model-based speech separation strategies discussed in Chapter 1. The two variants of this approach are to either fit a model to the mixture as a sum of source signals and then learn the parameters of the model from a large database, or to fit a model to the observed speech signal itself. In the former case (c.f. [50]), the speech signal is assumed to be a sum of source signals, and given the observation, the likelihood of its coming from each source is calculated under a specific statistical model. The observation is then hypothesized as to have come from the source which gives the maximum likelihood of having generated that observation under that model. Once the decision is made, the pitch estimate is then obtained from the analysis of that observation. As one example of the latter case (c.f. [25]), the spectrogram of the mixture signal is modeled as a sum of sources, and each source is modeled as a mixture of Gaussians in both the time and frequency domains. With the constraint that the Gaussians should be located at harmonic locations along the frequency axis for each source, the means of the Gaussians are then estimated for the observed spectrogram which gives the locations of the harmonics of each source signal. The estimation of pitch from the harmonic information is then a trivial step. An issue with the model-based approaches is the training required to learn the parameters of the models, and another is the generalization of the models to various databases.

The feature-based approaches on the other hand rely on certain properties of the signal to come up with pitch estimates. These typically include the autocorrelation, the average magnitude difference function and their variants, the spectrum of the signal, etc. The earliest approach to pitch tracking was the spectrum-based approach, where the idea was to find the "dominant" fundamental that has generated most of the peaks in the spectrum , and then remove all of its harmonics from the spectrum. Following this step, the same algorithm was used to find the next pitch estimate by collecting the remaining harmonics in the spectrum, which would arise from the second speaker's pitch. The drawback of spectrum-based techniques is their sensitivity to the

length and shape of the analysis window used, as well as their susceptibility to noise [22]. In particular, since male and female speakers exhibit different ranges of pitch, the optimal frequency resolution required by these populations for accurate pitch estimation are significantly different. This makes it difficult to arrive at a good set of parameters for the window length and shape that could yield robust estimates for both populations of speakers, especially when both genders occur simultaneously in the *same* speech mixture.

The use of autocorrelation as a measure of periodicity has led to the development of the modern successful pitch detection algorithms. The autocorrelation function (ACF) of a signal compares the current version of a signal to a delayed version of itself, by multiplying the two versions together. This function of delay or lag shows a maximum when the signal is most similar to itself. Thus for periodic signals, it will show local maxima at lag values equal to the pitch period and its multiples. This property has been used to develop algorithms that calculate the autocorrelation of the speech signal and then assign the peak as the pitch period estimate of the input signal. Numerous improvements and variations, like the enhanced ACF [41] where the autocorrelation is added to a time-compressed and expanded version of itself, and multiple window length analysis, have ensured the success of the algorithm by reducing pitch halving and doubling errors. Corresponding multiple pitch estimation approaches take the method further by first estimating the dominant pitch from the maximum of the ACF, and then filtering the signal in the time domain by a filter whose frequency response would cancel the harmonics of the dominant pitch [9]. In effect, this step corresponds to the harmonic cancelation method described in section 1.2. Following this cancellation, the second pitch estimate is obtained using the ACF of the resultant signal. The issue with this approach is that harmonic cancelation does not always effectively cancel the effects of the dominant pitch. Further, the harmonic cancelation procedure is highly susceptible to the formant structure of both the speakers. In particular, in regions of speech where the first formant, F_1, is close to the pitch of the dominant speaker, it is difficult to filter out the effects of F_1 which often results in erroneous second pitch estimates. Finally, the design of an efficient cancellation filter that works equally well for all ranges of pitch is a difficult (and still an open) problem. It may be noted, though, that the segregation algorithm proposed in this thesis can be viewed as a step in that direction, since the algorithm *can* be interpreted as a time-varying filtering operation with property of extracting (or in general focusing on) harmonics corresponding to a specific pitch.

The most successful approach towards multi-pitch detection is the so-called spectro-temporal approach (c.f. [52]), wherein the signal is first split into a number of channels modeling the human auditory processing system. The autocorrelation is calculated for each of the channels. This signal is then summed across all channels to yield the summary ACF (SACF). The peak of this SACF is identified as the dominant pitch and it has been shown to be more robust since it combines information across a number of channels, thus using multiple sources of information. Following estimation of the pitch from SACF, the dominant pitch and all its factors are removed from the analysis, and the next dominant pitch is then found from the SACF. This is used as the pitch estimate of the second speaker, akin to the procedure mentioned above. The issue of this spectro-temporal approach to multiple pitch detection is that the second peak of the ACF (as indeed, sometimes the first peak as well) need not correspond to the actual pitch period of the speaker. Instead, it may correspond to a peak resulting from the harmonic interaction of the actual pitch periods of the two speakers, especially when the common factors or multiples of the two pitch periods correspond to lag values within the possible range of the pitch analyzer. This phenomenon is described in greater detail in 2.3.

[23] designed a variation of the above approach, in which the SACF is first calculated as described above. Following this procedure, all channels contributing to the dominant pitch are labeled as belonging to the dominant speech, and the other channels are labeled as belonging to the non-dominant speech. The channels corresponding to the dominant pitch are then collected together to re-estimate the true pitch of the dominant speaker, as it has been observed by the authors that the estimated (dominant) pitch need not be the same as the true pitch. Enforcing continuity constraints, the pitch of the dominant speaker is then re-estimated using the channels identified as dominant as well as the current and future frames. The same method is applied to the

channels identified as non-dominant, to estimate the pitch of the second speaker. It may be noted that for frames where background and target speech are almost equally strong, this corresponds to simply picking the second peak of the ACF as described above. If, on the other hand, the second speaker is not strong enough, then the peaks in the ACF due to the second pitch would be very weak and mostly indistinguishable. In such cases, the pitch-tracker shows great unreliability in tracking the pitch of the second speaker, often completely missing out the presence of a second speaker. Thus, this method usually works well in the estimation of the dominant pitch, but exhibits problems in the estimation of the pitch of the weaker speaker.

[52] developed an approach which can be viewed as a hybrid of the feature based and model based approaches. The algorithm is first trained on a dataset for which the true pitch of the constituent source signals are pre-calculated. The channel-wise autocorrelations are calculated as usual, but not summarized across channels. Instead, for each channel, the peak of the ACF is noted, and is compared to the actual pitch period of the dominant signal (obtained from the reference). The difference between these two is calculated for the set of training data, for each channel. A probability distribution model of this difference is obtained across the entire dataset for each channel and the algorithm is trained on these models. For test data, the peak of the ACF is used to obtain a hypothesis for the pitch period (using the distributions obtained from the training data), conditioned on one of three possible hypotheses: a single voiced source, two voiced sources, no voiced source. Using the hypothesis for different channels, a net pitch estimate is obtained for the frame. Across multiple frames, a Hidden Markov Model (HMM) is used to preserve pitch continuity and create robustness of pitch estimates. The issues with this algorithm are as mentioned previously for model-based approaches - the need to train on a database, and the question of generalization of models. Once again, the point to note here is that the peak of the ACF does not necessarily correspond to the pitch period. In fact, instead of mathematically accounting for the differences in the ACF peaks and true pitch periods, this phenomenon is implicitly accounted for in their approach by the use of probability models to correct for the pitch estimates.

2.3 Motivation For Developing A New Multi-Pitch Tracker

To understand the reasons why multi-pitch trackers have such a tough time tracking the pitches of individual speakers, it is necessary to analyze the behavior of the feature they extract for pitch estimation, namely the Autocorrelation Function (ACF) or the Average Magnitude Difference Function (AMDF). The ACF of a signal $x[n]$ is defined as follows:

$$R_n[k] = \sum_{m=-\infty}^{\infty} x[n+m]w[m]x[n+m-k]w[m-k] \tag{2.1}$$

The interpretation of this is that the ACF value for a given lag k is equal to the windowed version of the signal $x[n+m]w[m]$ multiplied with a version of itself delayed by a lag k, $x[n+m-k]w[m-k]$, and averaging the result thus obtained. When a signal exhibits periodicity, the product terms in the autocorrelation are large and the autocorrelation function shows peaks at lag values equal to the periodicity of the signal, because it is at those delay values that the signal is most similar to itself and consequently the product is highest. Therefore, in order to estimate the pitch period of a periodic signal, we can calculate the autocorrelation as a function of the lag, and then identify the lag at which the autocorrelation signal shows a maximum. Practically, if the window size is large enough, there are multiple such peaks, each of which corresponds to a multiple of the pitch period. Thus, the least common divisor of all these peak locations is taken as the pitch period estimate. This is the method in which typical single pitch trackers estimate the pitch of a speech signal.

An alternate feature used to estimate the pitch of speech signals is the Average Magnitude

Difference Function (AMDF). The AMDF is defined as:

$$\gamma_n[k] = \sum_{m=-\infty}^{\infty} |x[n+m]w[m] - x[n+m-k]w[m-k]| \tag{2.2}$$

The AMDF is very similar to the ACF as the expression shows, except that the AMDF exhibits dips at pitch period locations. When a signal exhibits periodicity, the difference between the terms in the AMDF is very small, and thus the AMDF shows dips at lag values equal to the periodicity of the signal. The pitch period of the (quasi) periodic signal like voiced speech can be estimated by using the locations of the dips, similar to using the location of peaks for the ACF. The AMDF is used in the rest of this work because it has been shown to have a more robust behavior compared to the ACF when the envelope of the speech signal is greatly fluctuating and less susceptible to the effects of formants [10], and also because of its relatively less computational load (ACF involves multiplication while the AMDF involves addition).

In the case of multi-pitch tracking, the AMDF or ACF method is extended and the first maximum in the ACF is taken to estimate the pitch period of the stronger (dominant) signal. The maxima corresponding to this dominant pitch period and its multiples are then ignored in the next iteration, where the next most dominant peaks are identified and their locations are marked as the pitch period of the second signal. However, the problem of multi-pitch tracking is not fully solved because of the issues of (1) the dominant peak not exactly corresponding to the pitch period of the dominant speaker, and (2) the periodicity of the second speaker not manifesting itself strongly as compared to that of the dominant speaker. It is obvious that as the number of speakers increases, the problem of multi-pitch tracking will become harder. In the three-speaker case, since the periodicity of the third speaker could potentially be drowned by the other speakers, the ACF is hardly expected to reveal the peaks corresponding to the third speaker. Even if the pitch periods of the first two speakers were determined accurately (which is as yet an unsolved problem), the requirement of removing the peaks at multiples of the first and second pitch periods leaves potentially little room to identify the third peak. In the presence of noise, the peaks due to spuriousness could potentially be stronger than that due to the third speaker, and this can make identification of the third pitch impossible.

Fig. 2.1 illustrates the problem encountered by multi-pitch trackers, and the reason why they sometimes give an incorrect pitch estimate for a frame that is extracted from a sample of speech with more than one speaker. Fig. 2.1(a) shows the summary autocorrelation function (SACF) across all analysis channels for a mixture of two speech signals. The x-axis shows the ACF as a function of the lag and the y-axis shows the strength of the ACF. The correct pitch periods of the individual source signals which contributed to this mixture are also shown as vertical lines in green and blue. It can be seen that the ACF peaks are located at the lags corresponding to the pitch period of signal 2, and its multiples. Thus, algorithms which pick the maximum peak to identify the pitch of the dominant signal would give a correct estimate of that pitch. However, upon eliminating the maximum peak and its multiples, the second maximum peak would give an estimate around 12.8 ms instead of the correct value of 8.5 ms. This means that the second pitch estimate would be erroneous using this algorithm. This figure illustrates the situation when the periodicity of the second speaker does not manifest itself as strongly as the pitch of the dominant speaker - in such cases, the peaks due to inter-harmonics arising from the interference of the two pitch periods might be stronger than the peak due to the pitch itself.

Fig. 2.1(b) illustrates the Average Magnitude Difference Function (AMDF) for the same frame. The AMDF shows dips at locations where the ACF shows peaks, and in case of periodic frames, the period of the signal can be estimated by locating the dips of the AMDF. It can be seen that the locations of the dips of the AMDF are able to capture the pitch periods of both the signals. The minimum dip of the AMDF locates the pitch period of signal 1, and when the dip locations corresponding to its multiples are removed from the analysis, the next dip that is found corresponds to signal 2. However, like ACF, the dip locations of the AMDF are near but not exactly at the correct pitch periods. This problem occurs because the inter-harmonics arising due to the interference between the two pitch periods causes the dips to shift from their original

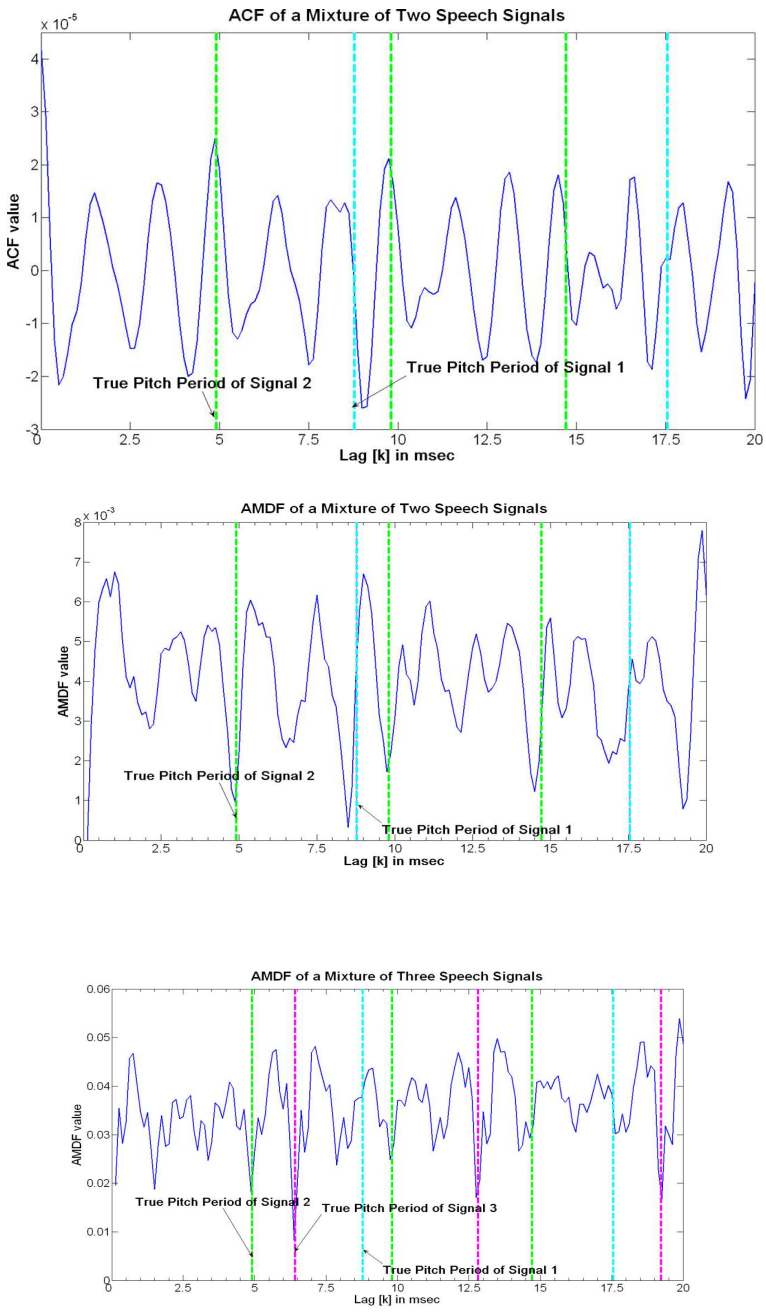

Figure 2.1: ACF and AMDF of the sum of speech signals, with the periods of the source signals and their multiples overlaid

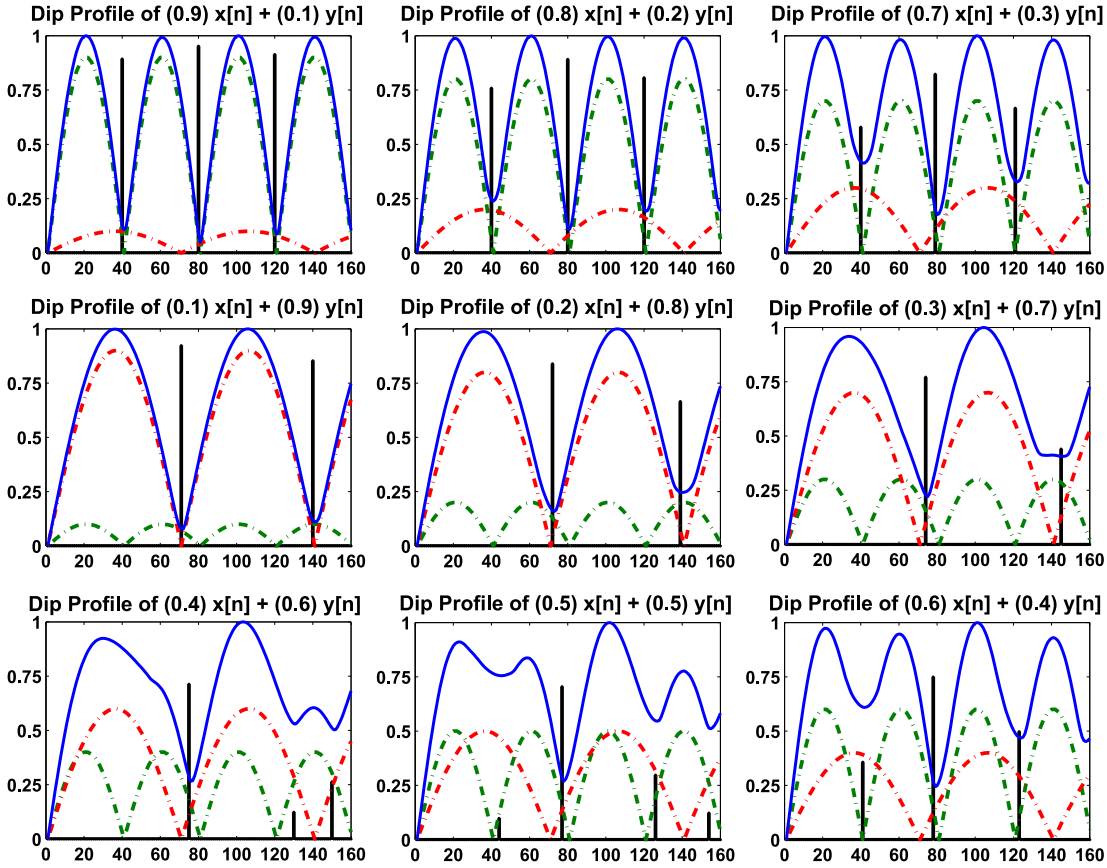

Figure 2.2: Illustration of the AMDF as a function of varying contributions of the two sinusoids. Green indicates AMDF of $x[n]$, Red that of $y[n]$ and Blue indicates AMDF of the sum of both.

locations, thus causing the pitch estimates to be off their true values. This phenomenon becomes more pronounced for some frames than for others, and depends on the interference pattern. Fig. 2.1(c) illustrates the AMDF for a sum of three speech signals. In this case, the AMDF shows dips at locations corresponding to the pitch periods of two of the speech signals, but the pitch effect of the third signal is completely lost. The inter-harmonics of the first two (i.e., the identified) signals are relatively stronger than the periodicity of the third one.

This behavior is further illustrated in Fig. 2.2, where the AMDF is illustrated for the sum of two sinusoids of different periods, namely 40 ms and 70 ms. Here, the sinusoids are added in different relative strengths and their AMDFs are plotted. The relative contribution of each sinusoid is given at the top of the panel in each figure. The blue line indicates the AMDF of the resultant signal; the green and red dotted lines are the AMDFs of the individual component sinusoids of period 40 ms and 70 ms, respectively. The black lines indicate the position and relative strengths of the dips in the AMDF (blue line). This means that the location of the tallest black line would be called the first pitch estimate by all the algorithms reported above, and the location of the black line which is not at a multiple of the first pitch estimate would be reported as the second pitch estimate. In the top panel, the three subfigures show the cases where the signal $x[n]$ dominates significantly over $y[n]$, and in those cases the algorithm would typically identify only the first pitch period and call the signal as having only one constituent source. In the middle panel, $y[n]$ dominates significantly over $x[n]$ and correspondingly the dips are all located near multiples of the period of $y[n]$; as such, the algorithms reported above would again typically find only one source. In the last panel, the signals are added in comparable proportions. In that case, the dip locations vary according to mixing ratio. The pitch estimates then would be two different values,

but would not both be correct.

This simple example shows the problem suffered by pitch trackers which rely on using peaks from the ACF or AMDF. The peaks or dips of such period extracting functions depend greatly on the relative proportions of mixing of the two signals. While getting the pitch estimate of the dominant signal is usually fairly easy, this cannot be said about the pitch estimate of the second signal. Indeed, as shown above, the dominant pitch estimate also might not correspond to the exact true pitch location, but may be off by a few samples. This illustration now explains the motivation behind the algorithm of [52] which relied on using trained models of a specific parameter, namely the difference between the ACF peak and the true pitch period.

This example also illustrates why the estimation of pitch in case of three or more speakers would be an even more daunting job. The complicated behavior of the inter-harmonics arising from three or more sources would be too difficult to predict and one would be hard-pressed to find ways to choose the best dips for proper estimation of all constituent pitch periods. Indeed, all results reported in the literature thus far focus on estimating the pitch from a mixture of only two speakers, and to our best knowledge, there are no results reported on pitch estimation involving more than that. This might be due to the fact that the ACF or the AMDF of such a complex mixture becomes too hard to analyze and the search for a "third" or "fourth" maximum peak becomes an almost impossible task, especially if spurious peaks are caused by background noise.

2.4 Proposed method for Multi-Pitch Detection

A more accurate way to perform multi-pitch tracking, which can also be easily generalized (at least theoretically) to work for more than two speakers lies in the extension of the AMDF to more than one dimension. We define the 2-D AMDF as follows:

$$\gamma_n[k,l] = \frac{1}{W-(k+l)} \sum_{m=0}^{(W-1)-(k+l)} |x[n+m] - x[n+m-k] - x[n+m-l] + x[n+m-k-l]| \quad (2.3)$$

$$r_n[k,l] = \frac{1}{W-(k+l)} \sum_{m=0}^{(W-1)-(k+l)} x[n+m].x[n+m-k].x[n+m-l].x[n+m-k-l] \quad (2.4)$$

where $x[n]$ is the signal being analyzed, W is the window length for analysis (AMDF calculation), k and l are lag parameters and $\gamma_n[k,l]$ is the AMDF of the signal $x[n]$ evaluated at time instant n. In comparison to the 1-D AMDF defined in section 2.3, there are two different lag variables as compared to a single lag variable in the 1-D AMDF, which explains why this measure is called the 2-D AMDF. In this case, the AMDF varies as a function of not one lag (x-axis) but two lag parameters (along both x and y axes). An example of the 2-D AMDF for a mixture of two pure sinusoids is shown in Fig. 2.3. The 2-D AMDF can be clearly seen as a function of two variables, namely k and l. Just like in the case of the 1-D AMDF or 1-D ACF, the 2-D AMDF rises and falls depicting the periodic nature of the two signals in the mixture. In fact, the local minima of the 2-D AMDF occur at the multiples of the pitch periods of both the constituent signals, with the global minimum occurring at the lag values equal to the pitch periods of the two signals. While Fig. 2.3 shows the 2-D AMDF for an ideal case (sum of two pure sinusoids), the 2-D AMDF for a typical speech signal looks very similar to the case shown here. As such, this feature can also be extracted for real speech mixtures and be used to design a multi-pitch tracker that can simultaneously estimate the pitch of both speakers.

This particular measure was initially proposed as the dual-difference function (DDF) by de Cheveigne in the early 1990s [9], defined slightly differently than shown above. There has not been much focus on its use as a pitch tracker, which is probably due to the susceptibility of this measure to noise as well as effects of windowing and varying envelope, as often encountered in

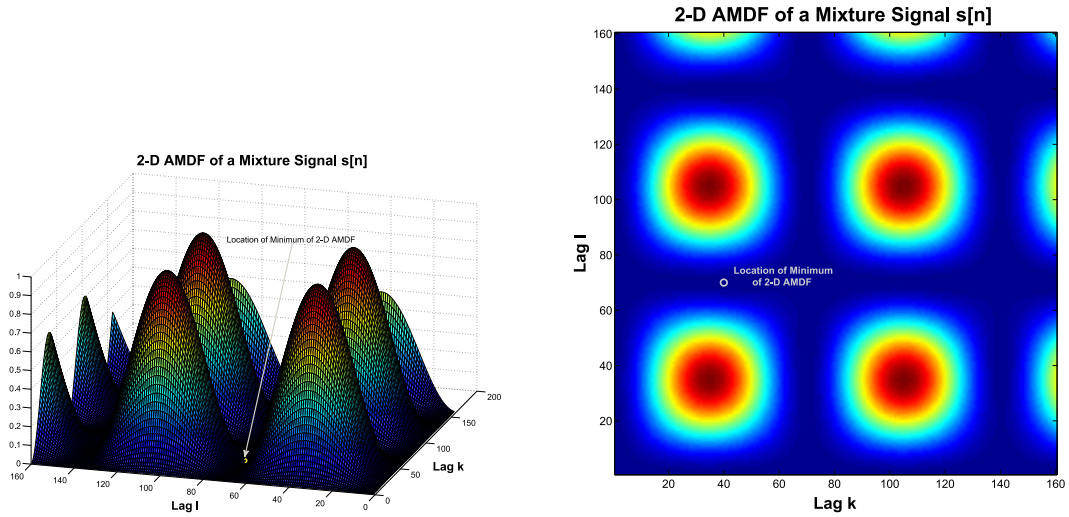

Figure 2.3: Illustration of the 2-D AMDF of a mixture signal. Red regions represent high AMDF values, and blue represent low AMDF values. The AMDF minima or pitch estimates would be in the "bluest" region. The pitch periods of the two signals in the mixture are identified as the location where the 2-D AMDF reaches its lowest value, marked in a grey circle in this figure. The figure on the left shows the 2-D AMDF as 3-dimensional data, while the figure on the right shows the same information on a plane.

speech. However, in this work, we modify the 2-D AMDF measure to robustly estimate the pitch in multi-talker and noisy environments. We accomplish this by by modifying the definition of the 2-D AMDF and combining information from multiple channels in an intelligent way, at the same time adapting it to the environment. A more important reason why we have chosen to extend this approach is its ease in generalization to multiple speakers. In order to track three speakers, we could define a 3-D AMDF.

Consider the 2-D AMDF as defined above. This AMDF shows a similar behavior as compared to the 1-D AMDF, i.e., it will show dips at lag values that correspond to delays where the signal is very similar to itself. However, the variation is now in 2 dimensions, and the pitch period can be estimated by finding the minimum of the AMDF along both the dimensions. Fig. 2.4 shows the 2-D AMDF of the same set of sinusoids whose 1-D AMDF was displayed in Fig. 2.2. The x-axis represents the lag value k and the y-axis represents the lag value l. Due to the three dimensional nature of the data represented (both lag dimensions and one AMDF strength dimension), the representation is made on 2 dimensions by color coding the AMDF strength. Red regions show the locations where the AMDF has a high value, and blue regions show the locations where the AMDF has a low value. Pitch estimates of the two sources are found by locating the minimum of the 2-D AMDF, i.e., the location where the image is the "most blue in color". Automatic extraction of the minimum location (marked in black circles) gives pitch estimates of both the constituent signals. In this particular case, since the signals are pure tones, the minima of the pure tones are not apparent (they will be located at the points which are "most blue" in the figure).

The practical implementation of the multi-pitch algorithm is shown in 2.5. This implementation is an extension of the 1-D pitch detector developed in [12], with some additional important blocks to estimate pitch of two speakers (e.g., blocks exclusively focusing on identifying instances of pitch-matching between speakers). The underlying assumption for this algorithm is that the maximum number of simultaneous speakers in the input signal is two. The description of the multi-pitch algorithm now follows. The input speech signal is first split into a number of channels using a filter-bank. Then, for each channel, the signal within that channel is processed on a frame-wise basis. The 2-D AMDF is computed for each frame, as described in 2.3. Following this, the

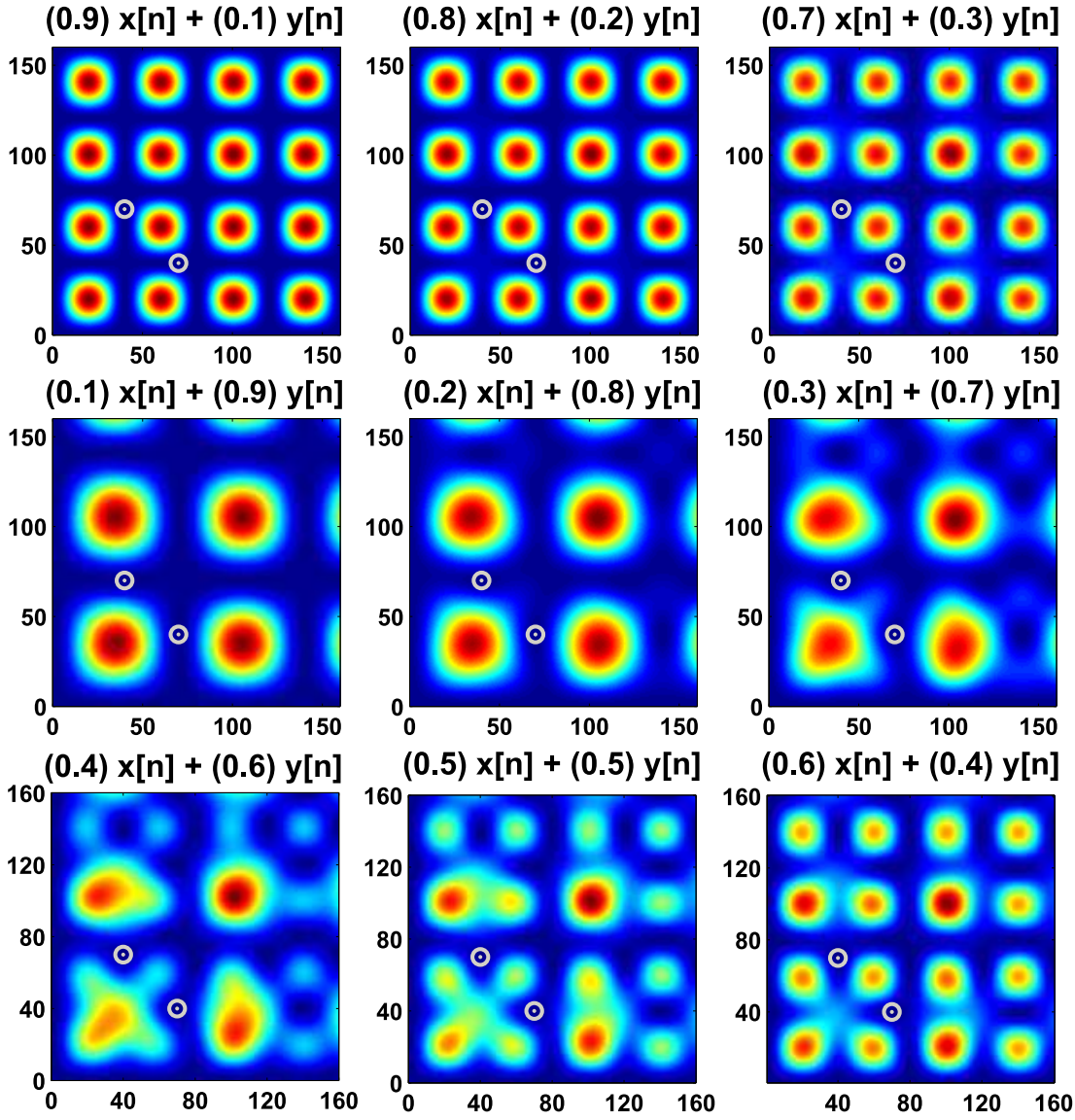

Figure 2.4: Illustration of the 2-D AMDF as a function of varying contributions of the two sinusoids shown in Fig. 2.2. Red regions represent high AMDF values, and blue represents low AMDF values. The AMDF minima or pitch estimates would be in the "bluest" region.

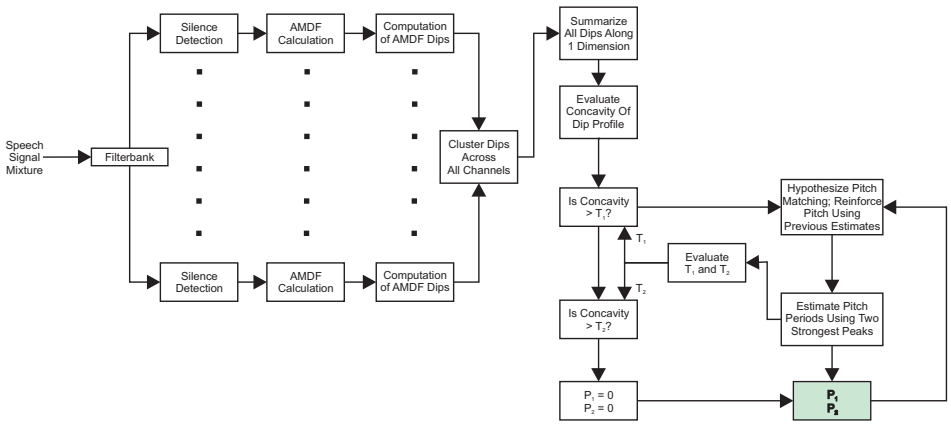

Figure 2.5: Block diagram of the proposed multi-pitch detection algorithm

2-D AMDFs are analyzed for what are called the strengths of the dips in the AMDFs. Next, for each time instant (frame), the dips from all channels are added together to get a 1-D summary dip profile that combines information across all frequency channels and both the lag dimensions. This summary dip profile is then analyzed for its concavity; depending on this concavity, the estimates of both the pitch values are obtained. If the concavity is less than a lower threshold, T_1, then the algorithm hypothesizes that neither of the two speakers is voiced, and outputs both pitch estimates P_1 and P_2 to be zero. If the concavity is greater than a higher threshold, T_2, the algorithm hypothesizes that either both pitch values are the same, or one of the pitch values is a multiple of the other. In such case, the previous pitch estimates are used to make the appropriate decision and estimate the pitch values accordingly. Finally, if the concavity lies between T_1 and T_2, then the two pitch estimates are found by identifying those lag locations at which the summary dip profile shows its two strongest local maxima. In all three cases, in addition to the pitch estimates P_1 and P_2, the algorithm also calculates the confidences C_1 and C_2 of these pitch estimates respectively, using the summary dip profile. In order to make the algorithm adaptive to the environment (which may be noisy), the varying strength of the signal being analyzed and the varying contributions of the two speakers to the current frame being analyzed, the thresholds T_1 and T_2 are changed adaptively as a function of the confidences of the two pitch estimates C_1 and C_2. The function that maps these confidences C_1 and C_2 to the thresholds T_1 and T_2 was learnt from large volumes of data containing speech mixtures in noise, but once learnt, the function has been found to be robust to various kinds of noises and SNRs/TMRs.

2.4.1 Two Dimensional AMDF And Its Properties

The basic ingredient of the algorithm is the 2-D AMDF, which is defined in 2.3. The AMDF compares the input signal to a delayed version of itself, for two lag or delay values, yielding a function of two variables k and l. For a perfectly periodic signal, pitch periods can be estimated by finding the 2-dimensional point where the AMDF is zero. The advantage offered by the 2-D AMDF over the 1-D AMDF is insensitivity to interaction between the harmonics. Inter-harmonics do exist, but they usually influence the AMDF dips through only one dimension; an inter-harmonic might cause a local minimum at some k_x by virtue of its effect in the k dimension. But in order for it to cause a wrong pitch estimate, it must simultaneously influence the l dimension at some l_y as well, and that would yield a pitch estimate (k_x, l_y) different from the true (T_0,T_1). This, for ideal combinations of pure tones, does not happen mathematically and in case of quasi-periodic signals like speech, was empirically found to be a very rare occurrence. It is for this reason that the 2-D AMDF was used as the feature to identify the pitch periods of the two participating speakers in

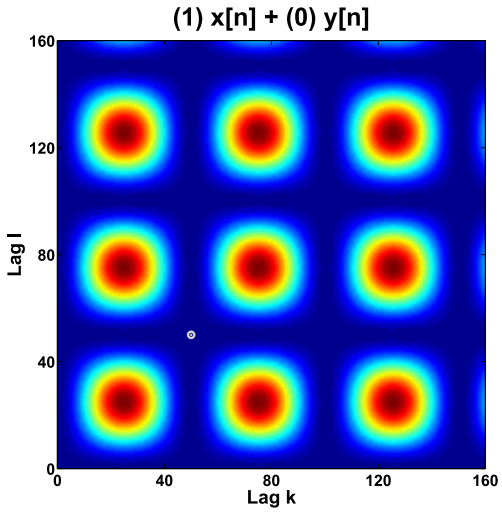

Figure 2.6: 2-D AMDF of a signal $s[n]$ which has just one periodic component $x[n]$ and the other periodic component $y[n]$ is zero

the speech mixture.

We believe that the improved performance of the 2-D AMDF over the 1-D AMDF is due to the behavior of the inter-harmonics. The 1-D AMDFs suffer from the problem of the inter-harmonics arising from the two constituent pitch periods which is why they do not always give the correct pitch track. However, the 2-D AMDFs do not suffer this problem. Inter-harmonics might exist either in the k-direction or the l-direction but seldom exist simultaneously in both directions. Indeed, if we pick any row or column of the 2-D AMDF, we will see dips and peaks at periods that need not necessarily be at the pitch period of either source, or its multiples (in fact, the 1-D AMDF is simply the first row or column of the 2-D AMDF, i.e., γ_n[k, 0] or γ_n[0,l]). But if we look across both the dimensions, then for an inter-harmonic to cause wrong pitch estimates, it must exist in both the k- and l-dimensions, which is rare. In fact, we believe that inter-harmonics due to N individual sources in a mixture cannot exist in N-dimensional space, and can only exist in (N-1)-dimensional or lower dimensional space and as such will not deteriorate pitch estimates obtained from the N-dimensional feature extracted from the mixture signal, though this is just an empirical observation thus far and we need to perform some mathematical analysis to verify this.

From the expression of the 2-D AMDF, it can be seen that one of the most important properties of the 2-D AMDF is its symmetry. In particular, due to its definition, it can be seen that γ_n[k,l] = γ_n[l,k]. This is useful because it implies that in the actual process of calculating the 2-D AMDF, it is only required to do so for values of $k \leq l$. Following this, the 2-D AMDF for the remaining lag values can be simply filled in by virtue of the symmetry of this feature. This helps in a saving of the computation, as the number of lags over which the 2-D AMDF needs to be calculated is now reduced by half.

A useful property of the 2-D AMDF is that it can also be used when there is only one speaker present. Fig. 2.6 shows the 2-D AMDF for a signal $s[n]$ which contains only one periodic signal. This figure illustrates that when there is only one voiced speaker (i.e., only one periodic component), the 2-D AMDF shows its dips at (approximately) the same location in both the lag dimensions - thus, both pitch estimates are the same. In actual pitch detection of speech mixtures, such behavior helps identify the frames in which there is only one voiced speaker.

A second useful property is that the 2-D AMDF can be generalized to a 3-D AMDF so that the pitch periods of three source signals can be estimated simultaneously. The 3-D AMDF is

defined as follows:

$$\gamma_n[k,l,p] = \frac{1}{W-(k+l+p)} \sum_{m=0}^{(W-1)-(k+l+p)} |x[n+m] - x[n+m-k] - x[n+m-l] - x[n+m-p] + x[n+m-k-l] + x[n+m-l-p] + x[n+m-p-k] - x[n+m-k-l-p]| \quad (2.5)$$

$$r_n[k,l,p] = \frac{1}{W-(k+l+p)} \sum_{m=0}^{(W-1)-(k+l+p)} x[n+m].x[n+m-k].x[n+m-l]. x[n+m-p].x[n+m-k-l].x[n+m-l-p]. x[n+m-p-k].x[n+m-k-l-p] \quad (2.6)$$

where $x[n]$ is the signal being analyzed, W is the window length for analysis (AMDF calculation), k, l, p are lag parameters and $\gamma_n[k,l,p]$ is the AMDF of the signal $x[n]$ evaluated at time instant n. This is a function of three dimensions each of which is a lag parameter (k, l, p). The minimum value of the AMDF across all three dimensions can be found, and the location of this minimum gives the pitch period of the three source signals that contributed to the mixture. Using simulations, it was verified that the above-defined 3-D generalization is indeed valid for multi-pitch estimation, because it can lead to not just the pitch periods but also their multiples, as seen in the 1-D ACF or AMDF cases. However, a drawback of using the 3-D AMDF is the large amount of stationary data required to compute this feature, which can be a major constraint as discussed next.

The 2-D AMDF calculation involves lags in 2 dimensions, and the windowed signal must be long enough to allow this calculation. In particular, if the lag values k and l take a range of values from 0 to, say, T_{max}, then for a given n, the *minimum* length of the signal to be analyzed for the computation of the 2-D AMDF will range from $x[n]$ to $x[n+2T_{max}]$. Furthermore, the calculation of the 2-D AMDF requires averaging of the data over a number of samples to attain robustness of the feature - assuming we want to average over M samples, the whole operation of calculating the 2-D AMDF requires a window length of $W = M + 2T_{max}$. This typically yields a requirement of a very long window. As an example, assuming a minimum detectable pitch frequency of 80 Hz and a sampling frequency of 8000 Hz, practical constraints require the analysis window length W to be at least 45 msec, which is more than twice the normal length of analysis windows in typical speech applications. Since the underlying assumption in the calculation of the 2-D AMDF is that the periodicity properties of the signal $x[n]$ are not changing during the window W, this requires that the signal be stationary for at least 45 msec. This explains the first limitation of the 2-D AMDF - it requires an assumption of stationarity over a much longer window than allowed in typical speech applications. In order to compensate for the temporal over-smoothing (and consequently incorrect pitch estimates) that this assumption might cause, the frame rate of the algorithm is set to be very high in our practical implementation. In particular, we choose a frame rate of 400 fps, i.e., one frame every 2.5 msec.

Another limitation of the 2-D AMDF is that like most temporal pitch determination algorithms, it is also susceptible to pitch period doubling errors, as a multiple of the lag value can also cause the AMDF to dip to its local minimum. Furthermore, since speech is a quasi-periodic signal, the AMDF will seldom fall to zero but will only fall to a local minimum. Finally, the 2-D AMDF was also found from our experiments to be susceptible to additive noise. For these reasons, as also to counter the other kinds of errors caused during the calculation of the 2-D AMDF due to practical constraints like effects of the window length and violation of the stationarity assumption, a measure of the degree or strength of an AMDF dip is calculated in the next block of the algorithm to find the "strongest" dip. This extra step significantly improves the performance of the 2-D AMDF as a pitch detecting feature, and significantly increases the potential of the proposed multi-pitch algorithm.

We will now describe the different blocks of the multi-pitch detection algorithm.

2.4.2 Analysis Filterbank and Silence Detection

The input signal is split into a set of channels by an auditory gamma-tone filter bank with center frequencies (CFs) based on the ERB scale, and ranging from 100 Hz to just below half the sampling rate, with the aim of modeling the human auditory hair cell processing [32]. Following this, the output of each channel is windowed to W samples to form a Time-Frequency Unit (TFU), and all TFUs corresponding to the same time are grouped as a frame. Silence detection is performed next. At the frame level, a frame is judged non-silent only if its energy is no more than 35 dB below the maximum energy computed across all frames in the utterance. At the T-F level, for all non-silent frames identified, a channel is considered non-silent only if its energy is no more than a pre-set threshold (in our system, 45 dB) below the maximum channel energy that has been computed up to that point, including the present frame. Following this, all the remaining steps discussed below are performed for each non-silent T-F unit. The purpose of using the gammatone filterbank is to partly mimic the human audition process of the inner ear hair cells splitting the speech signal into different bands, effectively weighting the frequencies as they would be considered significant by the human auditory system.

2.4.3 Computation of AMDF Dip Strengths

To accurately and robustly estimate the periodicity, the local minima of the 2-D AMDF as well as their dip strengths are calculated. The local minima of the 2-D AMDF help identify the potential pitch estimates, since the 2-D AMDF is ideally expected to fall to zero (and in practice achieves a low value) at lags equal to or multiples of the pitch periods of both signals. Thus, identification of the local minima narrows the search for the potential lag values to those lags where the 2-D AMDF achieves a local minimum. Next, since this yields multiple candidates and only one of these candidates is the actual location identifying both the pitch estimates, a method is required to identify which of the several candidates is the correct one. The method proposed here is to define a quantity called the "strength" of each dip (local minimum), and then select the dip with the greatest strength as the final pitch candidate. In the current incarnation of the multi-pitch algorithm, since several channels are used to collectively estimate the pitch, these dip strengths are calculated for each channel and then added across all channels. The final dip strength is obtained as the sum of these dip strengths across channels, and this is then used to estimate the pitch.

From analytical geometry, it is known that the local minima can be calculated by finding all locations where the gradient of the AMDF in the k and l dimensions is zero, and the Hessian is positive definite (> 0). This latter condition of the Hessian translates to the determinant of $H(\gamma_n[k,l])$ being greater than zero, i.e., $det(H(\gamma_n[k,l])) > 0$. While ideal signals like pure tones yield a Hessian matrix which is perfectly positive definite, real speech signals often yield a Hessian which may be positive definite. Furthermore, the effects of quantization of the lag domain and the consequent sampling of the 2-D AMDF may cause the Hessian to sometimes have a determinant with a slightly negative value, even at local minima. As such, in practice, the determinant of the Hessian is compared with a threshold, $H_{threshmin}$, which is typically set equal to -0.02 in the algorithm. This is how the local minima of the 2-D AMDF are identified - the locations where the gradient is zero and the determinant of the Hessian $> H_{threshmin}$. Thus, we have

$$[k_i, l_i]_{min} = \left\{ [k,l] \;\middle|\; \begin{array}{c} \nabla\gamma_n[k,l] = (\frac{\partial\gamma_n[k,l]}{\partial k}, \frac{\partial\gamma_n[k,l]}{\partial l}) = (0,0), \text{and} \\ \\ det(H(\gamma_n[k,l])) = det(\begin{bmatrix} \frac{\partial^2\gamma_n[k,l]}{\partial k^2} & \frac{\partial^2\gamma_n[k,l]}{\partial k \partial l} \\ \frac{\partial^2\gamma_n[k,l]}{\partial k \partial l} & \frac{\partial^2\gamma_n[k,l]}{\partial l^2} \end{bmatrix}) > H_{threshmin} \end{array} \right\} \tag{2.7}$$

Next, the strength of each local minimum is defined by comparing the AMDF value of this minimum to the AMDF value of the local maxima surrounding this minimum in all four directions. These

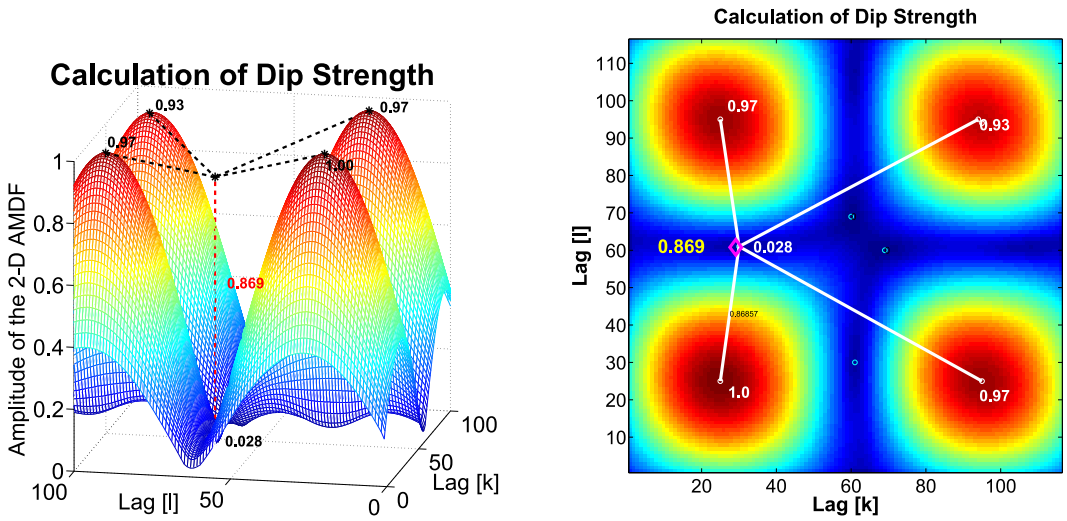

Figure 2.7: Calculation of the dip strength of a particular minimum. (Left) Local naxima around this minimum are used to interpolate the AMDF value (black lines) at the location of the minimum. The dip strength of the minimum is then obtained as the interpolated AMDF value minus actual AMDF value (red vertical line). (Right) Same procedure illustrated in the (k,l) plane. The black lines in the (Left) figure are shown in white here, and the red line is now represented by a single magenta diamond. The values of the 2-D AMDF at the minimum and neighboring maxima locations are shown in both figures, as is the evaluated dip strength. The blue dots represent the other local minima of the 2-D AMDF.

local maxima are located by setting the gradients zero but requiring the Hessian to be negative definite. In line with the above reasoning the local maxima are identified in practice as those locations where the gradient is zero and the determinant of the Hessian $< H_{threshmax}$ ($= 0.02$ in practice). Thus, we have

$$[k_i, l_i]_{max} = \left\{ [k,l] \;\middle|\; \begin{array}{c} \nabla\gamma_n[k,l] = (\frac{\partial\gamma_n[k,l]}{\partial k}, \frac{\partial\gamma_n[k,l]}{\partial l}) = (0,0), \text{and} \\ det(H(\gamma_n[k,l])) = det(\begin{bmatrix} \frac{\partial^2\gamma_n[k,l]}{\partial k^2} & \frac{\partial^2\gamma_n[k,l]}{\partial k \partial l} \\ \frac{\partial^2\gamma_n[k,l]}{\partial k \partial l} & \frac{\partial^2\gamma_n[k,l]}{\partial l^2} \end{bmatrix}) < H_{threshmax} \end{array} \right\} \tag{2.8}$$

Following this, a convex-hull like approach is followed to calculate the strength of each local minimum. Fig. 2.7 illustrates this procedure for a specific local minimum.

For each identified local minimum, the four nearest local maxima located in each of the four quadrants around the local minimum are first identified. In Fig. 2.7 (left), these local maxima are identified as the four black points on the AMDF surface, along with the AMDF values at those locations. In Fig. 2.7 (right), the maxima locations as well as the AMDF values at those locations are represented in white. The AMDF values at the four maxima are then used to interpolate the value of the AMDF at the location of the minimum. There are a number of different ways that multiple points can be used to interpolate the function value at a fifth point, and we used a Kernel-Based method [11] in our implementation. The interpolated value of the AMDF at the minimum location $\underline{a} = (k_a, l_a)$, obtained from the AMDF values at the maxima at locations $\{\underline{b_1}, \underline{b_2}, \underline{b_3}, \underline{b_4}\}$ (where $\underline{b_i} = (k_{b_i}, l_{b_i})$) is given by

$$\gamma_{interp}(\underline{a}) = \begin{bmatrix} \gamma(\underline{b_1})\gamma(\underline{b_2})\gamma(\underline{b_3})\gamma(\underline{b_4}) \end{bmatrix} \begin{bmatrix} d_{\underline{b_1},\underline{b_1}} & d_{\underline{b_1},\underline{b_2}} & d_{\underline{b_1},\underline{b_3}} & d_{\underline{b_1},\underline{b_4}} \\ d_{\underline{b_2},\underline{b_1}} & d_{\underline{b_2},\underline{b_2}} & d_{\underline{b_2},\underline{b_3}} & d_{\underline{b_2},\underline{b_4}} \\ d_{\underline{b_3},\underline{b_1}} & d_{\underline{b_3},\underline{b_2}} & d_{\underline{b_3},\underline{b_3}} & d_{\underline{b_3},\underline{b_4}} \\ d_{\underline{b_4},\underline{b_1}} & d_{\underline{b_4},\underline{b_2}} & d_{\underline{b_4},\underline{b_3}} & d_{\underline{b_4},\underline{b_4}} \end{bmatrix}^{-1} \begin{bmatrix} d_{\underline{a},\underline{b_1}} \\ d_{\underline{a},\underline{b_2}} \\ d_{\underline{a},\underline{b_3}} \\ d_{\underline{a},\underline{b_4}} \end{bmatrix} \tag{2.9}$$

where $\gamma(\underline{x})$ is the value of the 2-D AMDF at the location (k_x,l_x), and $d_{\underline{p},\underline{q}}$ is the distance between the locations (k_p,l_p) and (k_q,l_q). Using the fact that the distance $d_{\underline{p},\underline{q}} = d_{\underline{q},\underline{p}}$ and $d_{\underline{p},\underline{p}} = 0$, the above expression yields a simpler, symmetric matrix which needs to be inverted:

$$\gamma_{interp}(\underline{a}) = \begin{bmatrix} \gamma(\underline{b}_1)\gamma(\underline{b}_2)\gamma(\underline{b}_3)\gamma(\underline{b}_4) \end{bmatrix} \begin{bmatrix} 0 & d_{\underline{b}_1,\underline{b}_2} & d_{\underline{b}_1,\underline{b}_3} & d_{\underline{b}_1,\underline{b}_4} \\ d_{\underline{b}_1,\underline{b}_2} & 0 & d_{\underline{b}_2,\underline{b}_3} & d_{\underline{b}_2,\underline{b}_4} \\ d_{\underline{b}_1,\underline{b}_3} & d_{\underline{b}_2,\underline{b}_3} & 0 & d_{\underline{b}_3,\underline{b}_4} \\ d_{\underline{b}_1,\underline{b}_4} & d_{\underline{b}_2,\underline{b}_4} & d_{\underline{b}_3,\underline{b}_4} & 0 \end{bmatrix}^{-1} \begin{bmatrix} d_{\underline{a},\underline{b}_1} \\ d_{\underline{a},\underline{b}_2} \\ d_{\underline{a},\underline{b}_3} \\ d_{\underline{a},\underline{b}_4} \end{bmatrix} \tag{2.10}$$

Inversion of symmetric matrices of the above form are a well-studied problem in numerical analysis and can be solved fast using special techniques. In particular, this matrix which needs to be inverted is always of the same form and therefore its inverse is deterministic - thus, in practice we evaluate the actual inverse matrix instead of inverting it during runtime. Following interpolation, the strength of the dip at location $\underline{a}$ is then defined as the difference between the interpolated value and the actual value of the AMDF local minimum:

$$strength(\underline{a}) = \gamma_{interp}(\underline{a}) - \gamma(\underline{a}) \tag{2.11}$$

In Fig. 2.7 (left), the interpolated value is shown as a black dot in between the four AMDF maxima. The strength of the dip would then be the length of the red dotted line, which extends from the interpolated to the actual value of the AMDF at the minimum location. In Fig. 2.7 (right), the dip strength is given in yellow font near the location of the local minimum (shown as a pink diamond).

Since the dip strength is the difference between the interpolated and actual AMDF values, it captures to what degree the AMDF falls at a local minimum relative to its nearest maxima. This is a more accurate description of the significance of the local minima, and is less susceptible to the effects of practical signal processing applications. It is so because the signal processing causes the maxima also to decrease in strength as do the minima due to fewer number of samples being used in the calculation of the 2-D AMDF, and therefore, the dip strength would give an accurate description of how strong a dip is. Normally, the minima occurring at greater lag locations would have a AMDF value lower than the ones at lower lag locations (i.e., $\gamma(\underline{p}) < \gamma(\underline{q})$ for minima p and q if $|p| < |q|$) due to the windowing involved in the AMDF calculation. This is especially true for lag locations corresponding to the multiples of the actual pitch. This can influence the multi-pitch algorithm to yield incorrect pitch estimates. Using the interpolation strategy, however, the appropriate dip would have a high interpolated value and low AMDF value, making this dip strength maximum compared to all other dips. Thus, defining the strengths of dips and using them instead of the actual AMDF values at the dip locations is a more robust method of estimating the pitch.

2.4.4 Summary of Dip Strengths

In order to further improve the robustness of the algorithm to estimate multiple pitches in noisy environments, the dip information is summarized across multiple channels (frequencies) for each time instant. By combining dip strengths across all channels, the summary dip profile thus obtained will contain a more accurate yet robust description of both the pitch values in the speech signal. Furthermore, the information across both the lag dimensions is also combined into one single dimension, as this operation helps in dealing with certain particularly difficult pitch estimation cases. For example, there can occur certain frames wherein the pitch of one of the signals (speakers) is a multiple of that of the other (hereon refered to as "pitch matching" in the rest of this thesis): in such cases, the 2-D AMDF (as well as the usual 1-D ACF or AMDF) will not be able to identify that two distinct periodic signals are present and will simply show dips at multiples of the fundamental frequency of both signals. However, by combining the information from both dimensions into one

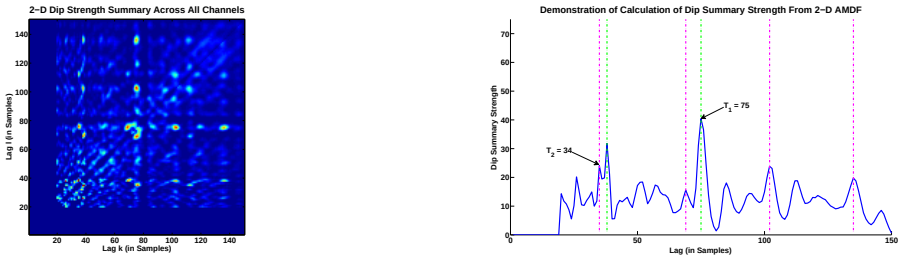

Figure 2.8: Calculation of the summary dip strength. (Left) The summary dip strength is obtained by adding the dip strengths across all channels, for each value of the lag dimensions k and l. (Right) The corresponding 1-D summary dip strength is obtained by adding the summary dip strength along one of the two dimensions. The example here is shown for a single frame where the pitch periods of both the constituent signals were 34 and 75 samples. The 1-D summary dip strength shows its dominant local maxima at these lag locations and their multiples, as is expected.

single dimension, and extracting information about what is called the Concavity of the resulting dip-strength summary, information about this so-called pitch-matching between speakers can also be extracted.

The summary dip strength across channels is obtained by simply summing the individual 2-D dip profiles for each channel across channels. In particular, if for channel c, the dip strength at lag location (k_0, l_0) is equal to s_c, then the dip summary strength at lag location (k_0, l_0) is simply equal to $\sum_{c=0}^{C} s_c$, where C is the total number of channels along which the signal is decomposed. Fig. 2.8(left) demonstrates the summary dip profile for a speech signal whose two components were of pitch periods corresponding to 34 and 75 samples. This 2-D summary dip strength is then collapsed into a single dimension, by summing the dip strength across one of these dimensions (say l dimension, so that the dip summary is now a function of only the k dimension). The feature thus obtained will be refered to as the summary dip profile, and contains the pitch-specific information about both the speakers in the speech mixture. Any "significant" local maxima of this dip profile will occur at lags equal to the pitch period (or its multiples) of either of the two component speech signals. The word "significant" includes all those local maxima, whose concavity is above a certain threshold as described in 2.4.5. Fig. 2.8(right) shows the dip profile for the same signal as shown in Fig. 2.8(left), with the significant maxima identified in different colors; those maxima corresponding to speaker A (pitch period 34 samples) are identified in magenta, while those corresponding to speaker B (pitch period 75 samples) are identified in green.

2.4.5 Concavity of the Dip Profiles

One of the important features introduced in this multi-pitch detection algorithm is the use of a concavity feature to identify the number of voiced speakers present, even in the difficult case where one of the pitch values is a multiple of the other. Given a dip profile $\lambda[k]$ (which is now a function of only one dimension k), the local maxima of the dip profile are first identified. Following this, all the local maxima are collected into groups so that the maxima which have a common factor fall in the same group. In case some of the maxima potentially fall into multiple groups, they are made members of all such groups. Following this, for each group, the concavity of that group of maxima is evaluated by evaluating the second derivative of the dip profile at all the maxima within that group, and averaging these values. The concavity of the summary dip profile is then said to be the maximum of the concavities of all the identified groups of maxima.

In particular, let G_i be the i^{th} group of maxima:

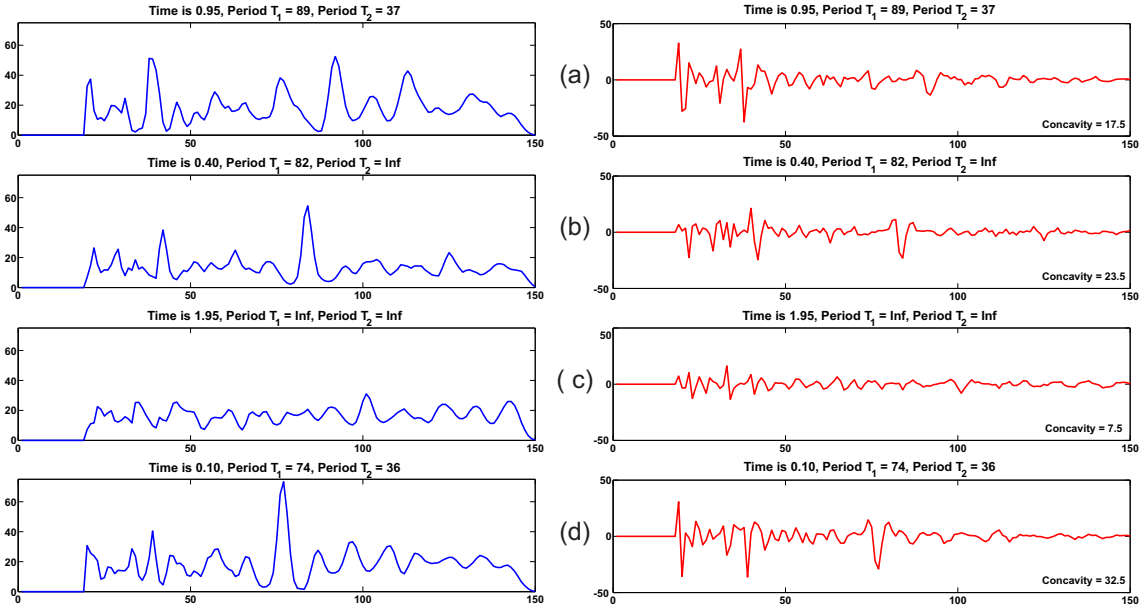

Figure 2.9: Dip Profiles and their Concavity values, for four different kinds of frames. (a) 2 simultaneous voiced speakers with distinct pitches, (b) 1 voiced speaker, (c) 0 voiced speakers, (d) 2 simultaneous voiced speakers with pitch-matching, i.e., one pitch is (close to) a multiple of the other

$$G_i = \left\{ k_j \;\middle|\; \frac{\partial \lambda[k]}{\partial k}\Big|_{k_j} = 0, \frac{\partial^2 \lambda[k]}{\partial k^2}\Big|_{k_j} < 0, \text{ and } \right.$$
$$\left. k_j \text{ have a common factor, say } f_i. \right\}$$

Then the concavity of this group G_i is given by

$$\rho_i = E\left[\frac{\partial^2 \lambda[k]}{\partial k^2}\Big|_{k \in G_i} \right]$$

Finally, the concavity ρ of the dip profile is equal to

$$\rho = max|\rho_i|$$

Typically, for the case of unvoiced speech (i.e., wherein both speakers are unvoiced), since there is no periodic structure in the signal, the groups G_i are small (i.e., contain few members due to lack of maxima at multiples) but numerous (due to the spurious nature of aperiodic energy). On the other hand, for the case of voiced speech, the groups G_i are larger since the maxima occur at multiples of a common fundamental (and therefore there will be several maxima with a common factor) and the number of such groups itself is small (since there are very few maxima outside those pertaining to the pitch periods of the voiced components).

The utility of the concavity measure defined above is apparent from Fig. 2.9. Here, the dip profiles of sum regions of speech are shown, along with the concavity measures corresponding to each of the dip profiles. Four different scenarios are illustrated: (1) two simultaneous voiced speakers with distinct pitches, (2) only one voiced speaker, (3) no voiced speaker and (4) two simultaneous voiced speakers, with one pitch being a multiple of the other. In the case of voiced speech, because the local maxima in each channel occur at lags corresponding to the pitch periods of the two signals, the summary dip profile will also exhibit a large dip strength at those lags corresponding to the pitch periods. This is illustrated in Fig. 2.9 (a), where the blue line represents

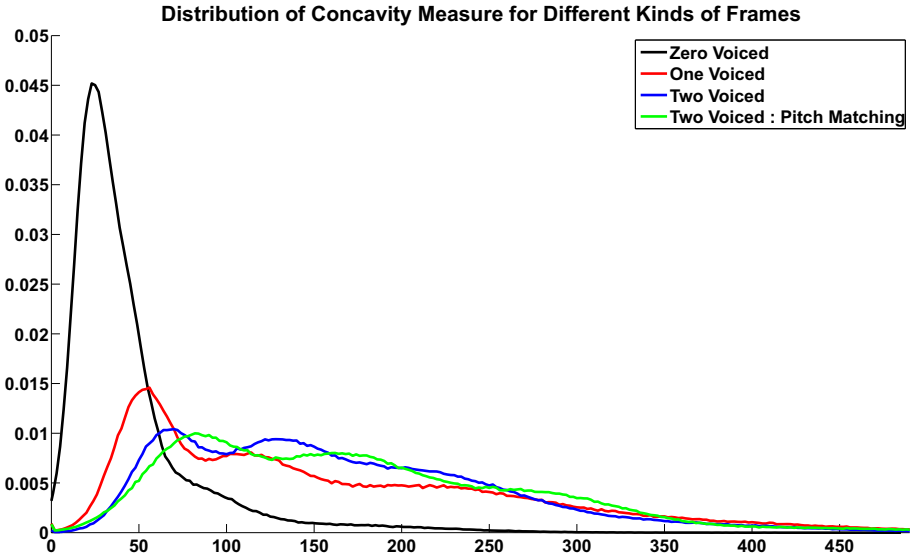

Figure 2.10: Distribution of the Concavity of the dip profiles for different kinds of voicing-unvoicing combinations of the two speakers

the summary dip profile for that frame, and the red line shows the second derivative of the dip profile. It can be seen that the local maxima of the dip profile occur at lags corresponding to the pitch periods, T_1 and T_2; the concavity value ρ for this dip profile is labeled in the figure with the derivative plot. For the case of only one voiced speaker, the peaks of the dip profiles occur only at the pitch period and its half. In the case of zero voiced speakers, due to lack of strong periodicity, the dip profile does not hit a significantly high number, and the concavity is also low as evidenced in the plots. Finally, in the case of two voiced speakers wherein the pitch of one is a multiple of the other, the dip profile shows a stronger maximum than in the case of one or two voiced speakers, and the concavity of the profile is also much higher than either of the latter cases. This illustrates the usefulness of the concavity measure in circumstances when the pitch of the two speakers "match" each other. The high value of concavity in the case of pitch-matching, as opposed to the case of single or two voiced speakers, helps to differentiate between frames in which there is no voicing, or voicing with no pitch-matching, or voicing with pitch-matching.

Fig. 2.10 substantiates the usefulness of the concavity measure by demostrating the distribution of this measure for the four different cases described above. The concavity values were collected from a database with various gender combinations (same and different gender, male and female) at 0 dB Target-to-Masker Ratio (TMR) and separated out depending on the classification of the frame among the above four categories. It can be seen that while there is some overlap between the distributions of the concavity for these four cases, these distributions are still well-separated and make a valid case for using this parameter to identify the kind of frame the algorithm is currently dealing with. Especially, the cases of one voiced speaker and two-voiced speakers with pitch-matching are well separated. This gives an excellent cue to identify how many voiced speakers are present, even if the number of pitch estimates obtained was just one. In such a case, if the concavity was low, it would suggest that there is only one voiced speaker in the current frame. On the other hand, if the concavity was quite high, it would suggest that there are actually two simultaneous voiced speakers, and the single pitch estimate was obtained because one pitch is a multiple of the other. With that information available, additional steps can then be used to identify the second pitch estimate (which would be a multiple of the one already obtained).

The multi-pitch detection algorithm thus exploits the concavity to identify how many simultaneous pitch estimates are required to be estimated, and uses that information to decide how many pitch estimates to look for. The algorithm achieves this by comparing the concavity of the current frame with a running threshold, T_1, to identify if pitch-matching occurred. If pitch match-

ing is hypothesized, then additional steps are taken to estimate the second pitch, as explained in section 2.4.6. If not, then the concavity is compared with a lower (running) threshold, T_2, to check if at least one voiced speaker is present in the current frame. If yes, then the decision about how many distinct voiced speakers are present, as well as their pitch estimate(s), are obtained as described in section 2.4.6. If neither of the thresholds is crossed, then the algorithm hypothesizes an unvoiced frame, and returns both pitch values as zero. In all cases, the "significant" dip clusters are those groups which cross the relevant threshold for the voicing-unvoicing decisions. It is these dip clusters or groups which are processed further to identify the pitch estimates for the current frame. The local maxima within these groups would correspond to lags which are either equal to the pitch period of either speaker, or a multiple thereof, and therefore the clusters containing these maxima would be the ones which exceed the voicing thresholds T_1 or T_2.

As can be seen from Fig. 2.10, the distributions of the concavity for the four cases of dip profiles are not as well separated from each other as to ensure very low voicing-unvoicing decision errors. As such, it is inefficient to set the thresholds T_1 and T_2 simply using the distributions of the concavity. The algorithm therefore uses adaptive thresholds instead of fixed values. The lowest and highest possible values these thresholds can take, are fixed using the distribution curve in Fig. 2.10. The current values of these thresholds, though, are obtained as functions of the confidences of the pitch estimates from the previous frame(s). This way, the concavity is compared with thresholds which are more relevant to the current region of speech being processed (as opposed to a global threshold) and it is also ensured that the thresholds themselves do not exceed the bounds suggested by the global distribution of the data for different kinds of mixtures.

It may be noted that in practice, to ensure the continuity of the pitch estimates over consecutive frames, the groups of maxima whose common factors lie close to that corresponding to the pitch estimates from the previous frame are emphasized by a factor of 10% so that these clusters have a higher concavity owing to strong voicing from previous frames.

Once the decision is made about how many voiced speakers are present in the current frame (i.e., how many pitch estimates to expect) and the corresponding significant maxima are identified by comparing the concavity of the dip profile with pre-set thresholds, the next block finally estimates the pitch values using these significant maxima.

2.4.6 Estimation of Pitch Values and their Confidences

The final stage of the pitch detection algorithm is the identification of the pitch estimates and their confidence values. This is done by comparing the concavity of the dip profile with pre-set thresholds, and analyzing those groups of maxima whose concavities exceed the appropriate threshold.

In case the concavity of the dip profile exceeds neither of the thresholds T_1 or T_2 described in section 2.4.5, the dip profile is hypothesized as being unvoiced, and the pitch estimates P_1 and P_2, as well as their confidences C_1 and C_2 are all set equal to zero. The thresholds T_1 and T_2 are then set equal to their default values, which were obtained using the distribution from Fig. 2.10.

For a dip profile whose concavity exceeds the voicing threshold T_1 but not the threshold T_2, the frame is hypothesized as containing one or two distinct voiced speakers. In such cases, one of the members of the groups whose concavity exceeds T_1 would typically equal the pitch estimate of one of the speakers. It is rare (but possible) that the group concavity exceeds the threshold yet none of its members is equal to the pitch estimate. Thus, all the groups whose concavity exceeds the threshold are first sorted in descending order of concavity. Next, the top two groups or clusters are identified, and the maxima within these groups are labeled as pitch candidates. For each of the two identified clusters, the strongest maximum is identified as the corresponding pitch estimate, and the strengths of these maxima are identified as the confidences of the pitch estimates. These yield the pitch estimates P_1 and P_2, with their corresponding confidences C_1 and C_2. If either of the confidences C_1 or C_2 is less than a pre-determined threshold T_C, the corresponding pitch estimate is considered unreliable and is set to zero. In case either of the previous pitch estimates

are not close to the current pitch estimates but closer to a multiple of the current estimates, then the current estimates are corrected to match the estimates from the previos frames to reduce pitch doubling or halving errors.

For a dip profile whose concavity exceeds the higher threshold T_2, the algorithm hypothesizes pitch matching between speakers (i.e., one pitch is a multiple of the other) and the group whose concavity exceeds the threshold is selected to find the pitch candidates. The location of the maximum of the dip profile is hypothesized as the first pitch estimate, and is matched to its previos pitch estimate. The second pitch estimate is found by finding the maximum in the selected group which is closest to the other pitch estimate from the previous frame. In case the two estimates from the previous frame are close to each other, the same maximum from the current frame is assigned to both pitch estimates P_1 and P_2. Finally, the strengths of the dip profiles at P_1 and P_2 are called the confidences of the estimates, C_1 and C_2 respectively. If either of these confidences falls below the confidence threshold T_C, the corresponding pitch estimate and confidence are set to zero.

Thus, at the end of this stage, both the pitch estimates P_1 and P_2, as well as their confidences C_1 and C_2, are estimated by the multi-pitch algorithm.

2.5 Pitch Assignment to the Appropriate Speaker

Until this point, the algorithm to identify the pitch estimates of the two speakers participating in the speech mixture has been described. However, given the identity of the speakers A and B in the mixture, the pitch estimates obtained are still labeled as P_1 and P_2 and have not been assigned to the appropriate speaker. The question of whether the pitch estimate P_1 came from the speaker A or B has not yet been addressed in this chapter. More generally, the pitch estimates from consecutive frames must be linked with each other so that two distinct speaker pitch tracks are maintained by the algorithm. While a simple distance-based metric, wherein the pitch estimates from the current frame can be compared with the pitch estimates and the pitch estimates which are close to each other are linked togther, can be used to assign the pitch to the appropriate speaker, this method can often fail. In particular, whenever the pitch values of the two speakers themselves get close to each other (i.e., as close as the distance used to make speaker assignments), it is difficult to assign the pitch to the correct speaker. Furthermore, there can occur frames when both pitch estimates go to zero (i.e. both speakers are simultaneously unvoiced) and at a later time one of the speakers shows voicing - in such situations, it is difficult to decide which speaker to assign this new non-zero pitch value to. This problem of assigning the pitch estimates to the correct speakers therefore requires more detailed study, and will be discussed in greater detail in Chapter 4.

2.6 Performance of the Multi-Pitch Detector

The proposed multi-pitch detection algorithm was tested on a database of synthetic mixtures created by adding together speech signals from the TIMIT database [18]. Three classes of mixtures were created: different gender (FM), same gender (male, MM) and same gender (female, FF). For each class, 200 pairs of sentences with lengths closest to each other were identified. In the case of the FM database, half the dataset had the male as the target speaker and half the dataset had the female as the target speaker. Care was taken during this process that no speaker or no utterance was the same in any pair. Each pair of signals was relatively normalized so that the ratio of their energies was 0 dB, i.e., both signals were equally strong. These were then added together in different target to masker ratios (TMRs), ranging from -9 dB to 9 dB in steps of 3 dB. This procedure gave a total of 600 mixture signals (200 for each class) for each of the seven

TMRs. In order to evaluate the output of the proposed algorithm, reference pitch values from the original speech signals before they were mixed were automatically extracted using ESPS Wavesurfer [44]. Since no manual correction was made, and automatic pitch extraction can be unreliable in boundaries between voiced and unvoiced regions, 4 frames on either side of all boundaries (total 20 msec) were discarded during the evaluation of the algorithm.

Fig. 2.11 shows the performance of the multi-pitch algorithm on a speech mixture sample. Both the true (reference) pitch values, as well as the estimates obtained by the algorithm, are shown. The top two panels show the "raw" pitch estimates, wherein the estimated pitch tracks are plotted as-is. As mentioned before, the assignment of the pitch estimates to the appropriate speakers is not yet done at this stage, which explains the switching of the pitch tracks between the two speakers. The bottom two panels show the pitch estimates as they would appear if they were assigned to the correct speaker, along with the corresponding true pitch values. It can be seen that the voiced and unvoiced regions are correctly identified, and the two pitch values are estimated correctly by the algorithm in most frames, with the majority of the errors being pitch doubling or halving errors. In addition, these errors mainly occur near the transition regions of speakers, i.e., whenever the number of voiced sources changes, and can be attributed majorly to practical problems arising due to windowing of the data. Otherwise, the proposed method gives reasonably good pitch estimates, and the quantitative values also show performance as good as or better than some of the state-of-the-art multi-pitch algorithms (c.f. [52]).

The quantitative analysis of the performance is done by defining the percentage of insertion, deletion and substitution errors for the target speaker alone. In general, the possible errors can be listed as follows. For each frame, the algorithm can give incorrect estimates of the number of voiced speakers (insertion errors when overestimating the number of voiced speakers or deletion errors when underestimating it), or report incorrect estimates of pitch (substitution errors). In order to capture all these errors and quantify performance, the errors were classified according to whether they were insertion errors, deletion errors or substitution errors. If the number of speakers estimated by the algorithm was greater than the true number of speakers, and one of the reported pitch estimates corresponded to the masker, it was called an insertion error. If the estimated number of speakers was less than the true number and the missing pitch estimate would have corresponded to the target, it was called a deletion error. For each frame where any pitch was estimated, if the reference pitch was non-zero and either one (or both) of the reported pitches varied by more than 8 Hz from the true pitch of the target, the error was called a substitution error. It may be noted that there may be frames that have insertion (or deletion) and substitution errors together. During the evaluation process, frames with pitch doubling or halving were not identified as errors. Furthermore, as long as these pitch estimates were obtained correctly, the estimate was considered correct even if there was switching between speakers. Finally, it may be noted that for the FM database, the target was male for half the dataset and female for the other half, thus avoiding any gender-specific factors in the performance. Fig. 2.12 shows the results of the algorithm for the three classes of mixtures: FF, MM & FM.

It may be noted that the algorithm shows performance that degrades very gracefully with decreasing TMR, with the relative error increasing only very slightly. However, the error rate at postivie TMRs appears to be rather higher than expected or desired. A preliminary study was made in order to explore the reasons for the errors caused by the algorithm, and it was found that some of the reasons why the algorithm fails can be attributed to the acoustic properties of the mixture signal - practical issues in multi-speaker environments. In particular, there could be some frames where one speaker is significantly dominated by the other (usually happens when one speaker is at the beginning or end of phonation, or uttering a semivowel, and the other speaker is uttering a vowel). Also, there could be some frames in which the pitch of both the speakers is close within the resolution of the proposed algorithm. In such cases, the algorithm would fail to extract the pitch of one of the speakers. A quantitative study of the number of errors explained by these phenomena was made. In all frames showing any of the 2 speaker errors, the relative energies of the speakers was calculated. If the ratio was more than 10 dB, then the frame was called energy dominated - this would help identify all those regions where one of the speakers was significantly stronger than the other. If the pitch of the two speakers were multiples or within 8 Hz of each

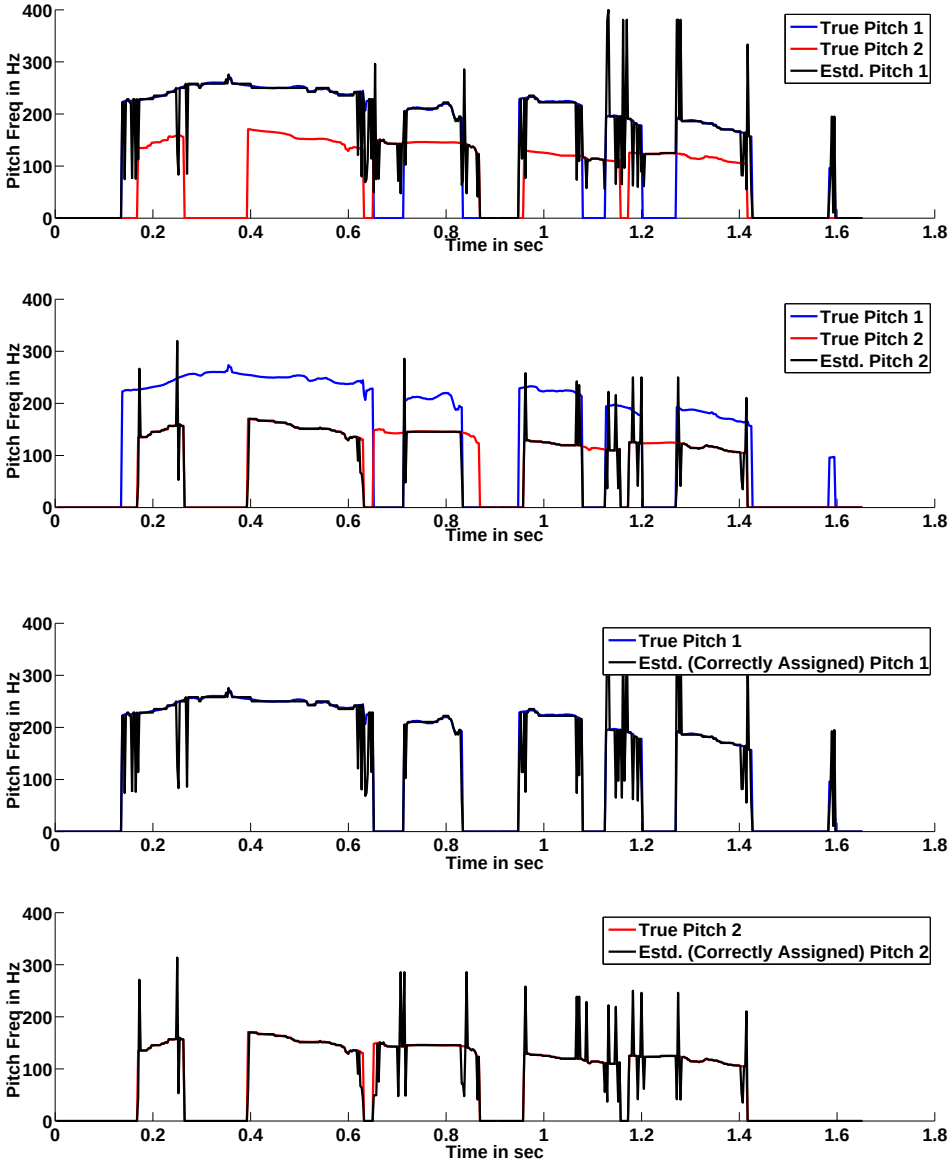

Figure 2.11: Pitch tracks showing the performance of the multi-pitch algorithm. Panels (1) and (2) show the true pitch vaues of the speech signals in the mixture, in red and blue. The two estimated pitch tracks are shown in black. In this case, the estimates are not assigned to the appropriate speaker. Panels (3) and (4) show the estimated pitch tracks as they would appear if they had been correctly assigned to the appropriate speaker.

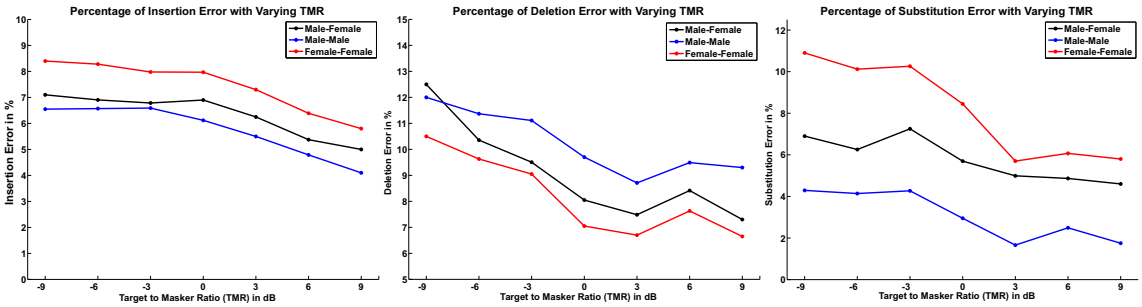

Figure 2.12: Performance of the multi-pitch algorithm on a database of speech mixtures with different gender combinations, and at varying TMRs

other, the frame was labeled as pitch-matched. The results showed that on average, about 68% of the erroneous frames were due to energy domination of one speaker over the other, and pitch matching accounted for 8% of the remaining errors. This accounts for more than 75% of the errors of the algorithm, and therefore it *is* to be expected that the algorithm perform as it actually does. In order to get the performance higher, further research must be done to improve the robustness of the multi-pitch algorithm in such cases where energy dominance or pitch matching occur.

2.7 Chapter Summary

In this chapter, we described a multi-pitch detection algorithm that is capable of identifying the presence of two simultaneous voiced speakers and yield their pitch estimates. The algorithm can also give a single pitch estimate in case there is only voiced source in the input signal. The algorithm depends on a new feature called the 2-D AMDF, which is an extension of the traditional AMDF used in pitch detection algorithms. This feature is not susceptible to the effects of harmonic interactions between two voiced sources, and thus is an ideal feature for identifying two pitch estimates. We design an algorithm to take advantage of this feature, and at the same time attempt to make it reliable for real-world conditions. As such, we add additional systems to quantify the behavior of the 2-D AMDF in terms of its dip strengths, and use this information to calculate pitch. We rely on the statistics of the behavior of these dips to identify how many voiced sources are simultaneously present in a given frame of signal analysis, and then identify their pitch values. Finally, we evaluate our algorithm on a database of speech mixtures and find our algorithm to be reliable for the task of speech segregation.

Chapter 3

SEPARATION OF THE VOICED COMPONENTS OF OVERLAPPING SPEECH SIGNALS FROM A SPEECH MIXTURE

3.1 Introduction

In this chapter, we will discuss the speech extraction algorithm in detail, both in the context of speech segregation and speech enhancement. The model is first described for the case of segregation, and later enhancement is shown to be a special case. Following this, some implementation details are discussed in the context of the speech enhancement case. Following the intuitions developed here, the proposed algorithm is then compared with other potentially similar methods in the literature, and the distinction of the propsoed algorithm over the others is highlighted. The segregation model described here is applicable to the voiced portions of speech - the chapter ends by identifying the tasks involved in the estimation of the unvoiced regions and evaluating the reliability of the estimated voiced regions.

The method of separating the speech signals depends on solving an over-determined system of linear equations that directly reveals the contributions of the two speech signals in the speech mixture. The segregation procedure starts by first analyzing the speech signal through a number of channels using a set of analysis filters. The signals obtained from the filterbank are then used together to estimate the pitch periods of the two speakers, as described in Chapter 2. Following this, the pitch estimates are used to extract the harmonics of the two speakers - this is the focus of the current chapter and Chapter 4. The separated streams are then assigned to the appropriate speakers by relying on certain features which could help in identifying which extracted component came from which source. The procedure is briefly covered in Chapter 2. Following this, all the streams coming from the same speaker are combined together to yield the individual speech stream of that speaker, and this is repeated for both speakers. This process yields the two speech signals that composed the speech mixture.

There are several novel aspects of the proposed algorithm to separate speech signals. First, this method is the first approach towards actually pulling apart the contributions of both speakers even in speech regions where both speakers overlap or one speaker is significantly stronger than the other, as opposed to the other current approaches which assign such speech regions to only the stronger or dominating speaker. Second, in contrast to most segregation methods that focus on generating segregated versions that perform well for automatic tasks like speech recognition and do not focus on good reconstruction in the acoustic domain, the proposed algorithm yields separated speech signals in the acoustic domain that are perceptually of better quality than the original mixture signals, thereby making the algorithm useful not only for automatic tasks like speech recognition or speaker identification, but also for improving speech quality in noisy communication or for distorted speech. Third, this method can be used to separate not only overlapping speech signals, but also speech from background noise. Finally, the algorithm can simultaneously estimate both the amplitudes and phases of the contributions of the two speakers in the mixture, as opposed to current methods which simply rely on the amplitude information to segregate speakers.

3.2 Modeling Voiced Speech Using A Set Of Complex Exponentials

Consider a stationary periodic signal $x[n]$, potentially complex, consisting of the harmonics of a periodic signal of frequency ω_0. This signal can be represented as follows:

$$x[n] = \sum_{i=1}^{N} (\alpha_i^+ e^{j\omega_0 in} + \alpha_i^- e^{-j\omega_0 in}) \tag{3.1}$$

where each value of i represents each harmonic of the basic frequency ω_0, α_i is a complex coefficient representing the relative contribution of the harmonic $i\omega_0$, and (+) and (-) represent the positive and negative frequency harmonics.

This means that the signal $x[n]$ is composed of a sinusoid of the basic frequency ω_0 and also its multiples $2\omega_0$, $3\omega_0$, $4\omega_0$, etc. with contributions from each frequency component being equal to α_1, α_2, α_3, α_4, etc. respectively. For a sampling frequency of F_s Hz, if the period of the signal is F_0, then the number of harmonics

$$N = \left\lfloor \frac{F_s}{F_0} \right\rfloor \tag{3.2}$$

We use this sum-of-exponentials model to model our speech signal, and try to estimate the contribution of each of the harmonics, i.e., α_i coming from each speaker. For an observation sequence $x[n]$ that satisfies the above model, given the task of estimating the unknown amplitudes α_i^+ & α_i^-, the unknown set can be estimated by using $M \geq 2N$ different values of $x[n]$. This is done by substituting $n = 1, 2, \ldots, M$ in the equation 3.1 and obtaining M equations in the N unknown coefficients:

$$\begin{array}{rcl} x[1] & = & \sum_{i=1}^{N} (\alpha_i^+ e^{j\omega_0 i1} + \alpha_i^- e^{-j\omega_0 i1}) \\ x[2] & = & \sum_{i=1}^{N} (\alpha_i^+ e^{j\omega_0 i2} + \alpha_i^- e^{-j\omega_0 i2}) \\ x[3] & = & \sum_{i=1}^{N} (\alpha_i^+ e^{j\omega_0 i3} + \alpha_i^- e^{-j\omega_0 i3}) \\ \vdots & & \\ x[M] & = & \sum_{i=1}^{N} (\alpha_i^+ e^{j\omega_0 iM} + \alpha_i^- e^{-j\omega_0 iM}) \end{array} \tag{3.3}$$

Expressing in matrix form, we have

$$\begin{bmatrix} x[1] \\ x[2] \\ x[3] \\ \vdots \\ x[M] \end{bmatrix} = \begin{bmatrix} e^{j\omega_0.1.1} & e^{j\omega_0.1.2} & \ldots & e^{j\omega_0.1.N} & e^{-j\omega_0.1.1} & e^{-j\omega_0.1.2} & \ldots & e^{-j\omega_0.1.N} \\ e^{j\omega_0.2.1} & e^{j\omega_0.2.2} & \ldots & e^{j\omega_0.2.N} & e^{-j\omega_0.2.1} & e^{-j\omega_0.2.2} & \ldots & e^{-j\omega_0.2.N} \\ e^{j\omega_0.3.1} & e^{j\omega_0.3.2} & \ldots & e^{j\omega_0.3.N} & e^{-j\omega_0.3.1} & e^{-j\omega_0.3.2} & \ldots & e^{-j\omega_0.3.N} \\ \vdots & & & & & & & \\ e^{j\omega_0.M.1} & e^{j\omega_0.M.2} & \ldots & e^{j\omega_0.M.N} & e^{-j\omega_0.M.1} & e^{-j\omega_0.M.2} & \ldots & e^{-j\omega_0.M.N} \end{bmatrix} \begin{bmatrix} \alpha_1^+ \\ \alpha_2^+ \\ \vdots \\ \alpha_N^+ \\ \alpha_1^- \\ \alpha_2^- \\ \vdots \\ \alpha_N^- \end{bmatrix} \tag{3.4}$$

which can be re-written as

$$\begin{array}{rcl} \underline{x} & = & [\mathbf{V}^+ \mathbf{V}^-]\, \underline{\alpha} \\ & = & \mathbf{A}\underline{\alpha} \end{array} \tag{3.5}$$

It may be recollected that $\underline{x}$ is a vector of observations (known), $\mathbf{A}$ is a matrix constructed from knowledge of the pitch frequency ω_0 (and thus known), and $\underline{\alpha}$ is the only unknown element here. This unknown vector $\underline{\alpha}$, which represents the contribution of the individual harmonics to the signal $x[n]$, can be estimated using the known quantities $\underline{x}$ and $\mathbf{A}$. If the choice of the number of equations $M \geq 2N$, then this gives an over-determined system of equations, and the least squares error solution for the set of equations 3.5 is given by

$$\begin{array}{rcl} \underline{\hat{\alpha}} & = & (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \underline{x} \\ & = & \mathbf{A}^P \underline{x} \end{array} \tag{3.6}$$

where $\mathbf{A^P}$ is the pseudo-inverse of the matrix $\mathbf{A}$. This means that in order to arrive at the contributions of N harmonics in the signal $x[n]$, M consecutive samples of the signal need to be used for the estimation. Incidentally, it may be seen that for the signal $x[n]$ described as above, the matrix $\mathbf{V}^+$ (and $\mathbf{V}^-$) is composed of columns which form a set of basis functions or signals when M is a multiple of N. The coefficients $\underline{\alpha}$ may also be found by the Gram-Schmidt Orthogonalization procedure in such a situation.

Thus, if a time series x is represented in terms of N complex exponentials $v[k]$, the amplitudes and phases of the N complex exponentials can be estimated such that the reconstruction error has a minimum norm and this will yield the contributions of the individual harmonics to the compound signal. We use this LS fitting principle in our segregation algorithm by modeling the speech mixture as a sum of harmonics of two different pitch frequencies instead of a single one as shown above. Since the number of equations M determines the size of the window being used, the signal is processed on a frame-wise basis. The window length M for analysis must not be too small, since in order to get stable estimates of the unknown coefficients, it is necessary that $M \geq N$ - in fact, it is known that as $M \to \infty$ the estimates of the unknown coefficients will reach the Cramer-Rao lower bound under the assumption of Gaussian noise on the model [40]. However, at the same time, the window length M cannot be too large since the model assumes a constant pitch frequency ω_0, and for a long temporal window this assumption may no longer be valid. Thus, the window length parameter is chosen to be an appropriate trade-off between these two factors (long enough to give stable estimates of coefficients, but short enough to allow for assumption of stationarity). The window shift is chosen to be such that consecutive windows do not leave a gap in the processing of the signal, i.e., there is significant overlap between consecutive window locations - typically, it is chosen to be equal to half the window length. Furthermore, in order to obtain adequate frequency resolution across the spectrum, individual LS models are fit to the periodic components of different frequency regions. This also helps in identifying certain spectro-temporal regions where the model may not be reliable. It must be noted that the coefficients $\underline{\alpha}$ are otbained from the data, and their reliability (i.e., how close the estimates from the mixtures are to the true value from clean speech) is therefore not always high. A method to counter this would be to discard all the estimates $\underline{\alpha}$ when there is reason to believe these estimates are not reliable (this will be discussed in greater detail in Chapter 4). Now, if the model was fit across all the channels together and the model proved unreliable, this would result in the estimates of the whole temporal region being discarded. By dividing the mixture speech signal into a number of TFUs, and then assuming that the signal in each TFU is stationary (so that the coefficients $\underline{\alpha}$ are the same within the TFU and the model is thus applicable), we achieve two important things: (1) the signal within the TFU is better represented by the model, since the model corresponding to each channel can now have its own estimates independent of other channels, (2) the reliability of the model can be estimated individually for different channels. As is obvious, the finer the frequency resolution (i.e., the more the number of channels along which the input signal is decomposed), the better the model is expected to fit the data and the more flexibility we expect in detecting unreliable estimates.

3.3 System Overview

The input mixture signal is first passed through an analysis filterbank that decomposes the input into a number of channels. Analysis is done on a frame-wise basis with overlapping frames; this yields several TFUs describing the input. For each frame, if the energy of the TFU is below a threshold, that TFU is labeled silent, not analyzed further, and does not contribute to the reconstruction of either stream. For every non-silent TFU, the pitch value of both speakers at that frame is obtained from the multi-pitch algorithm described in Chapter 2, and estimated versions of the original signals are reconstructed as described below. Following this, the reconstructed signals are combined across frequency and then overlap-added to get the final streams of both speakers.

We currently show the performance of the algorithm for the two-speaker case. The algorithm can be generalized to the multi-speaker case and should be a part of future research. In the two-speaker case, depending on the pitch tracks, there can be three possible scenarios: (1) both speakers are voiced, (2) only one speaker is voiced and (3) neither speaker is voiced. We tackle the three cases separately, explicitly refering to the first two cases in this chapter and the third one in Chapter 4.

3.3.1 Segregation of Two Voiced Speakers

We use the model in 3.2. As mentioned previously, we rely on the knowledge of pitch of both speakers for segregation. For a given TFU, if the pitch of both speakers is non-zero, it implies that the mixture signal being analyzed was obtained as the sum of two (quasi) periodic signals, $s_A[n]$ and $s_B[n]$. Calling the signal in a particular TFU of the mixture speech signal as $x_{TF}[n]$ and the angular pitch frequencies of both speakers in that frame as ω_A and ω_B, the mixture $x_{TF}[n]$ can be written as

$$\begin{aligned} x_{TF}[n] &= s_{A,TF}[n] + s_{B,TF}[n] \\ &= \sum_{i=1}^{N_A}(\alpha_i^+ e^{j\omega_A in} + \alpha_i^- e^{-j\omega_A in}) + \sum_{k=1}^{N_B}(\beta_k^+ e^{j\omega_B kn} + \beta_k^- e^{-j\omega_B kn}) \end{aligned} \tag{3.7}$$

where each of the α_i represent contributions from the harmonics of speaker A, and each of the β_k represent contributions from the harmonics of speaker B. The indices i and k run from 1 to N_A and 1 to N_B respectively, where N_A is the number of harmonics of ω_A between 0 and half the sampling frequency, and N_B is the same number corresponding to ω_B - this is to account for the harmonics covering the bandwidth of $x[n]$. For example, if $x[n]$ was band-limited between 0 Hz and 3000 Hz, and the two individual pitch estimates were 200 Hz and 350 Hz, then we would have $N_A = 15$ and $N_B = 8$, corresponding to the harmonics covered (0 Hz to 3000 Hz, and 0 Hz to 2800 Hz). Since we have the pitch estimates from the two speakers using the multi-pitch algorithm, we know the values of ω_A and ω_B. Correspondingly, we also know the values of N_A and N_B. The unknown parameters are the values α_i, β_k. We will try to estimate these parameters in order to estimate the contribution of each speaker to the mixture.

The idea is that each of the coefficients α_i and β_k represent the amount of voiced energy contribution from speakers A and B, respectively, at the various frequency components. If the total energy from speaker A significantly dominates over that of speaker B, then we can conclude that the signal $x_{TF}[n]$ had energy dominantly from speaker A, and vice versa if otherwise. If the energy from both sources was of a comparable value, then we can conclude that the energy is "shared" by both speakers. This will become more apparent below. Thus, our model incorporates the concept of shared TFUs as against the dominated TFUs in other approaches. In this way, we try to estimate the ITF_{COM} rather than the ITF_{DOM}, both of which concepts were introduced in 1.4.

In order to solve for the unknown coefficients, we can set this problem up as a Least-Squares fitting problem. As shown earlier, by choosing the length of the TFU to be equal to $M > 2(N_A + N_B)$, the above model can be used to solve for the $2(N_A + N_B)$ unknown coefficients $\underline{\alpha}$ & $\underline{\beta}$:

$$\begin{aligned} \underline{x} &= \left[\mathbf{V_A^+ V_A^- V_B^+ V_B^-}\right]\left[\underline{\alpha}\underline{\beta}\right] \\ &= \mathbf{C}\underline{\gamma} \end{aligned} \tag{3.8}$$

using the same approach as outlined above, where $\mathbf{C} = [\mathbf{V_A^+ V_A^- V_B^+ V_B^-}]$ and $\underline{\gamma} = [\underline{\alpha}\underline{\beta}]^T$. This gives us the estimates of the coefficients as

$$\begin{aligned} \underline{\hat{\gamma}} &= (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H\underline{x} \\ &= \mathbf{C}^P\underline{x} \end{aligned} \tag{3.9}$$

from which we can then extract the estimates of the coefficients $\underline{\hat{\alpha}}$ & $\underline{\hat{\beta}}$ by picking appropriate elements of the vector $\underline{\hat{\gamma}}$. Having obtained the estimates of the coefficients $\underline{\hat{\alpha}}$ which define the

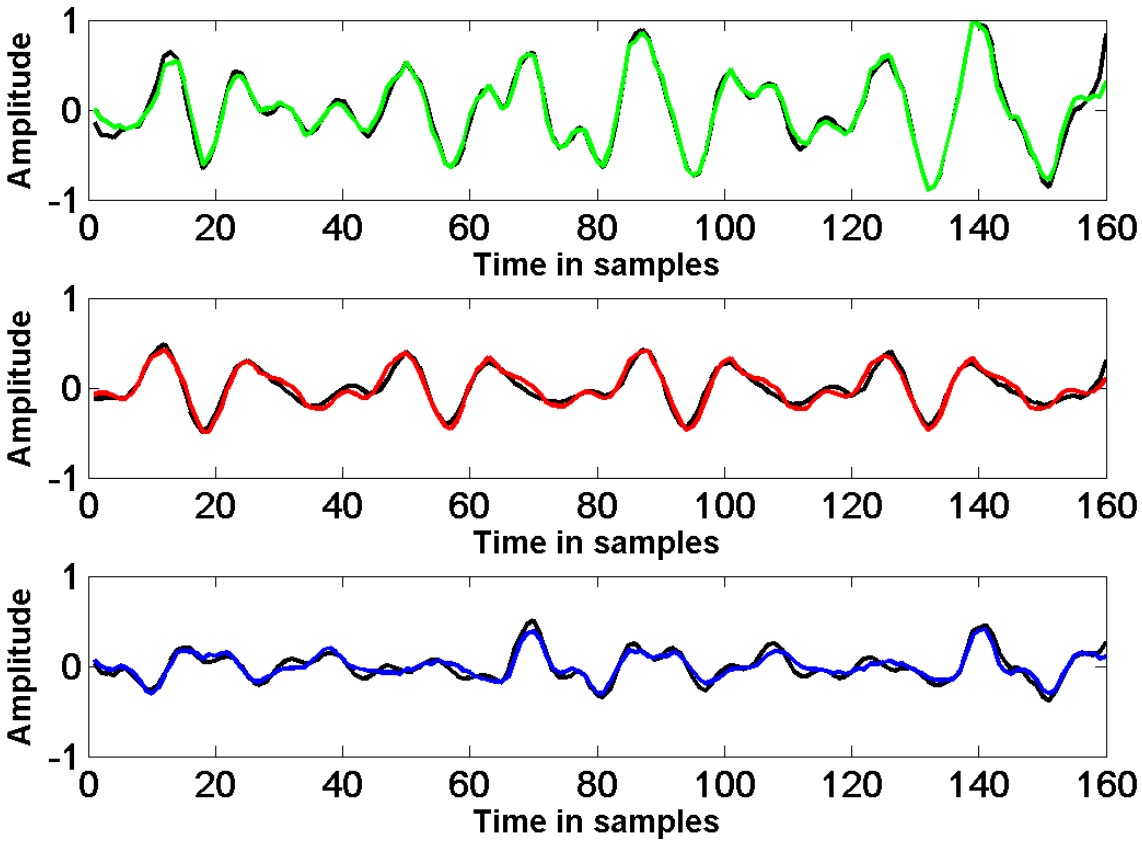

Figure 3.1: Demonstration of the segregation capability of the proposed segregation model. The original signals and the signals estimated by the proposed algorithm are shown. **(Top)** The original mixture signal (black) and the signal as reconstructed by the proposed model (green) **(Middle)** The signal from speaker A which contributed to the mixture (black) and its estimate using the proposed segregation model (red) **(Bottom)** The signal from speaker B which contributed to the mixture (black) and its estimate using the proposed segregation model (blue)

signal $s_{A,TF}[n]$ and $\underline{\hat{\beta}}$ which define the signal $s_{B,TF}[n]$ and which give the LS fit for the mixture signal, the individual signals that contributed to the mixture can be reconstructed as

$$\begin{aligned} \hat{s}_{A,TF}[n] &= \textstyle\sum_{i=1}^{N_A}(\hat{\alpha_i}^{+}e^{j\omega_A in} + \hat{\alpha_i}^{-}e^{-j\omega_A in}) \\ \hat{s}_{B,TF}[n] &= \textstyle\sum_{k=1}^{N_B}(\hat{\beta_k}^{+}e^{j\omega_B in} + \hat{\beta_k}^{-}e^{-j\omega_B in}) \end{aligned} \tag{3.10}$$

Thus, for each TFU, it is possible to reconstruct the periodic signals $\hat{s}_{A,TF}[n]$ & $\hat{s}_{B,TF}[n]$ that added together to yield the observed mixture signal, provided the pitch values ω_A and ω_B are known.

It may be noted that the model we have proposed tries to fit the data to a sum of harmonics, i.e., constrains the signal to adhere to a particular form. Thus, depending on how well the model fits the data, there will be an error signal which will define the applicability of the model to the data. We will use the L_2 norm of this error signal to quantify and evaluate how well the model represents the data (Chapter 4). Calling the error of the model as

$$\begin{aligned} \epsilon_{TF}[n] &= x_{TF}[n] - s_{A,TF}[n] - s_{B,TF}[n] \\ &= x_{TF}[n] - \textstyle\sum_{i=1}^{N_A}(\alpha_i^{+}e^{j\omega_A in} + \alpha_i^{-}e^{-j\omega_A in}) - \sum_{k=1}^{N_B}(\beta_k^{+}e^{j\omega_B kn} + \beta_k^{-}e^{-j\omega_B kn}) \end{aligned} \tag{3.11}$$

or in the vector form, as

$$\underline{\epsilon} \;=\; \underline{x} - \mathbf{C}\underline{\gamma} \tag{3.12}$$

we will define the power of the error signal:

$$E_{\epsilon_{TF}}[n] = ||\underline{\epsilon}||^2 \tag{3.13}$$

In cases where the proposed model does not fit the data well the power of the error signal, $E_{\epsilon_{TF}}[n]$, will be high. In such cases, the estimates of the two signals $\hat{s}_{A,TF}[n]$ & $\hat{s}_{B,TF}[n]$ are not reliable enough and should either not be used for the reconstructions of the final speech streams $\hat{s}_A$ & $\hat{s}_B$, or be modified in some way first before they can be used for reconstruction. The power of the error signal, $E_{\epsilon_{TF}}[n]$, will be used in Chapter 4 to both improve the estimates obtained by the harmonic model, as well as to predict the SNR and the aperiodic regions of the speech mixture.

It is well-known from literature (c.f. [20]) that the error signal $\underline{\epsilon}$ is that part of the data signal $\underline{x}$ which cannot be explained by the proposed model $\mathbf{C}$, and in effect is that component of the signal $\underline{x}$ which is orthogonal to all the columns of the matrix $\mathbf{C}$. Further, in case the matrix $\mathbf{C}$ is full rank (i.e., all its columns are mutually independent), the signals $\hat{s}_{A,TF}[n]$ & $\hat{s}_{B,TF}[n]$ are also orthogonal to each other. As a consequence, for most combinations of ω_A and ω_B the error signal $\underline{\epsilon}$ and the reconstructions $\hat{s}_{A,TF}[n]$ & $\hat{s}_{B,TF}[n]$ are orthogonal to each other. Given this, the energy of the mixture speech signal which was analyzed by the model is equal to the sum of the energies of the signals obtained by fitting the model to the data and the energy of the error or residual signal:

$$\begin{aligned} E_{x_{TF}}[n] &= E_{\hat{s}_{A,TF}}[n] + E_{\hat{s}_{B,TF}}[n] + E_{\epsilon_{TF}}[n] \\ &= E_{\hat{s}_{TF}}[n] + E_{\epsilon_{TF}}[n] \end{aligned} \tag{3.14}$$

In practical signal processing, there will very often occur cases wherein the proposed model is valid (yielding usable segregated signals) but the energy of the residual (error signal) is perceptually salient as compared to the energy of either of the recovered speech signals - this is especially true when the speech mixture has background noise nearly as strong as the speech component, or when both speakers are almost equally strong, or when there is pitch matching between speakers. In such cases, the amplitudes of the signals estimated can be boosted so that the (relative) perceptual effect of the residual can be reduced and the recovered signals are strong enough to yield perceptually preferable segregated signals. We perform this boosting by utilizing the energy of the segregated signals, $E_{\hat{s}_{A,TF}}[n]$ and $E_{\hat{s}_{B,TF}}[n]$, the energy of the original mixture signal, $E_{x_{TF}}[n]$ and the energy of the residual, $E_{\epsilon_{TF}}[n]$ - the exact procedure is described in Chapter 4 and partly in the following section. This step of modifying the estimates according to a certain criterion (discussed later) was found to be very effective in improving the quality of the segregated signals - as such, we make note of the above energy definitions.

3.3.2 Segregation of Voiced and Unvoiced Speech

For a given TFU, if one of the speakers (say B) is unvoiced (i.e., $\omega_B = 0$), then the observed mixture signal can be modeled as

$$\begin{aligned} x_{TF}[n] &= s_{A,TF}[n] + s_{B,TF}[n] \\ &= \textstyle\sum_{i=1}^{N_A} (\alpha_i^+ e^{j\omega_A in} + \alpha_i^- e^{-j\omega_A in}) + w[n] \end{aligned} \tag{3.15}$$

where $w[n]$ represents a noise source. This can be expressed for a window length $M > 2N_A$ as

$$\begin{aligned} \underline{x} &= \left[\mathbf{V}_\mathbf{A}^+ \mathbf{V}_\mathbf{A}^-\right] [\underline{\alpha}] + \underline{w} \\ &= \mathbf{V}\underline{\gamma} + \underline{w} \end{aligned} \tag{3.16}$$

where $\underline{w}$ is the noise vector for M samples. It is well-known that under the assumption of the noise being Gaussian-distributed, the minimum mean-square error solution for the estimates $\underline{\gamma}$ is given by Eqn. 3.9 [40, 20]. Thus, the estimate of the voiced component from speaker A is given by Eqn. 3.10. The unvoiced component of speaker B now needs to be calculated, and this will be discussed in Chapter 4.

3.4 The Case of Speech Enhancement

In the case of speech segregation, the observation sequence to be analyzed $x[n]$ is composed of harmonic components from two speakers with different angular pitch frequencies ω_A and ω_B. For the case of speech enhancement, there is a single voiced speaker which needs to be extracted from the noisy mixture, i.e., there exists only one pitch ω_0. Let the number of harmonics be $N_0 = \frac{F_s}{F_0}$. As such, the model proposed in Section 3.3.2 is applicable to this situation. Letting

$$\begin{aligned} x_{TF}[n] &= s_{0,TF}[n] + w[n] \\ &= \sum_{i=1}^{N_0}(\alpha_i^+ e^{j\omega_0 in} + \alpha_i^- e^{-j\omega_0 in}) + w[n] \\ &= [\mathbf{V}^+\mathbf{V}^-]\,[\underline{\alpha}] + \underline{w} \\ &= \mathbf{V}\underline{\alpha} + \underline{w} \end{aligned} \tag{3.17}$$

we have

$$\begin{aligned} \underline{\hat{\alpha}} &= (\mathbf{V}^H\mathbf{V})^{-1}\mathbf{V}^H\underline{x} \\ &= \mathbf{V}^P\underline{x} \end{aligned} \tag{3.18}$$

from which the voiced estimate of the speech signal in the noisy mixture is given by

$$\begin{aligned} \underline{\hat{s_{0,TF}}} &= \mathbf{V}\underline{\hat{\alpha}} \\ &= \mathbf{V}(\mathbf{V}^H\mathbf{V})^{-1}\mathbf{V}^H\underline{x} \\ &= \mathbf{P}_{\mathbf{V}}\underline{x} \end{aligned} \tag{3.19}$$

where $\mathbf{P_V}$ is called the Projection Matrix corresponding to the matrix $\mathbf{V}$. This operation implies that the voiced estimate $\underline{\hat{s_0}}$ is obtained from the observation vector $\underline{x}$ by a simple linear transformation of the latter. The exact transformation is defined by the projection matrix $\mathbf{P_V}$, which explains the name of this matrix (it projects $\underline{x}$ to $\underline{\hat{s_0}}$).

Therefore, through the extraction or seperation of the voiced components, the proposed segregation system is applicable for both the speech segregation and speech enhancement problems.

3.5 Physical Interpretation of the Proposed Model

We will now explore the exact nature of the proposed algorithm, and how it operates on the observation sequence $x[n]$ to yield the segregated sequences. For the sake of simplicity and mathematical tractability, we will try to understand the speech enhancement process, i.e., extraction of only one voiced component. We will then try to draw inferences about the segregation problem, i.e., extraction of two voiced components. In this process, we will also realize that while the proposed algorithm seems to yield a computationally expensive solution, the fact is that the solution boils down to a very simple averaging operation. Drawing from this conclusion, we will then focus (in Chapter 4) on the extraction of the speech component modeled by the proposed system (the voiced component) and that not modeled by the proposed system (the unvoiced component), and how to extract this information from the noisy speech signal.

It is seen from Eqns. 3.9 and 3.10 that the estimation of the voiced component $s_0[n]$ in the noisy signal $x[n]$ involves three steps: (1) computation of the pseudo-inverse of the matrix $\mathbf{V}$. (2) multiplication of the resultant matrix with the observed sequence $x[n]$ and (3) pre-multiplying this vector by the matrix $\mathbf{V}$. The computation of the pseudo-inverse, in turn, involves multiplication of two matrices, followed by inversion, followed by multiplication with another matrix. This is clearly computationally prohibitive, especially when the pseudo-inverse operation has to be performed for every time frame, and the matrix-vector multiplication needs to be performed for every time frame *and* every channel.

These computations can however be reduced significantly by carefully considering the special properties of the matrix $\mathbf{V}$. In general, for any matrix $\mathbf{A}$, it is difficult to arrive at a deterministic expression for the projection matrix $\mathbf{P_A}$. However, since the matrix $\mathbf{V}$ is a vanderMonde matrix

containing terms which are all powers of a unit-magnitude complex number $e^{j\omega_0}$, the pseudo-inverse of $\mathbf{V}$ and thence the projection matrix $\mathbf{P_V} = \mathbf{V}(\mathbf{V^H V})^{-1}\mathbf{V^H}$ can be computed to get explicit deterministic expressions for $\underline{\hat{s_0}}$ in terms of $\underline{x}$. We will consider two different cases.

3.5.1 Pitch Synchronous Speech Enhancement: Projection Matrix and Signal Estimate

Let us first consider the special case of pitch synchronous analysis, wherein the number of samples M is a multiple of the number of harmonics N_0, i.e., . That is, $M = pN_0$, where p is an integer ≥ 1. It can be proved in this case [48] that the Projection Matrix $\mathbf{P_V}$ is a block identity matrix of the form:

$$\mathbf{P_V} = \frac{N_0}{M}\begin{bmatrix} \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \text{(total } p \text{ such hori. terms)} \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \text{(total } p \text{ such hori. terms)} \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \text{(total } p \text{ such hori. terms)} \end{bmatrix} \text{(total } p \text{ such vert. terms)} \tag{3.20}$$

where $\mathbf{I}$ is an $N_0 \times N_0$ identity matrix. Therefore,

$$\underline{\hat{s_0}} = \frac{N_0}{M}\begin{bmatrix} \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} \end{bmatrix} \underline{x} \tag{3.21}$$

Effectively, the whole operation amounts to a simple averaging operation where each sample is replaced by the average of itself with all other samples within the window which are separated from it by any multiple of N_0 samples. For example, in the case when $M = 3N_0$, the samples at $x[k]$, $x[N_0+k]$ and $x[2N_0+k]$ will all be replaced by the average of the three values ($x[k]$, $x[N_0+k]$ and $x[2N_0+k]$). It should be noted here that since the window length M is exactly a multiple of the number of harmonics N_0, the number of samples which should be averaged is always equal to p irrespective of the value of sample number k within the window. Mathematically, this can be represented as:

$$\begin{aligned} \underline{\hat{s_0}[k]} = & \quad \tfrac{N_0}{M}(\underline{x[k \diamond N_0]} + \underline{x[N_0 + k \diamond N_0]} + \underline{x[2N_0 + k \diamond N_0]} + ... + \underline{x[(p-1)N_0 + k \diamond N_0]}) \\ = & \quad \tfrac{1}{p}(\underline{x[k \diamond N_0]} + \underline{x[N_0 + k \diamond N_0]} + \underline{x[2N_0 + k \diamond N_0]} + ... + \underline{x[(p-1)N_0 + k \diamond N_0]}) \end{aligned} \tag{3.22}$$

where the symbol $k \diamond N_0$ represents the reminder when N_0 divides k. Thus, the number of samples averaged here is always equal to p as shown in the equation (3.22). A sample projection matrix for the case of $N_0 = 7, M = 14$ is shown in Fig. 3.2.

3.5.2 Pitch Asynchronous Speech Enhancement: Projection Matrix and Signal Estimate

In the case of pitch asynchronous analysis, the number of samples M is not necessarily a multiple of the number of harmonics N_0. That is, $M = pN_0 + X$, where p is an integer ≥ 1

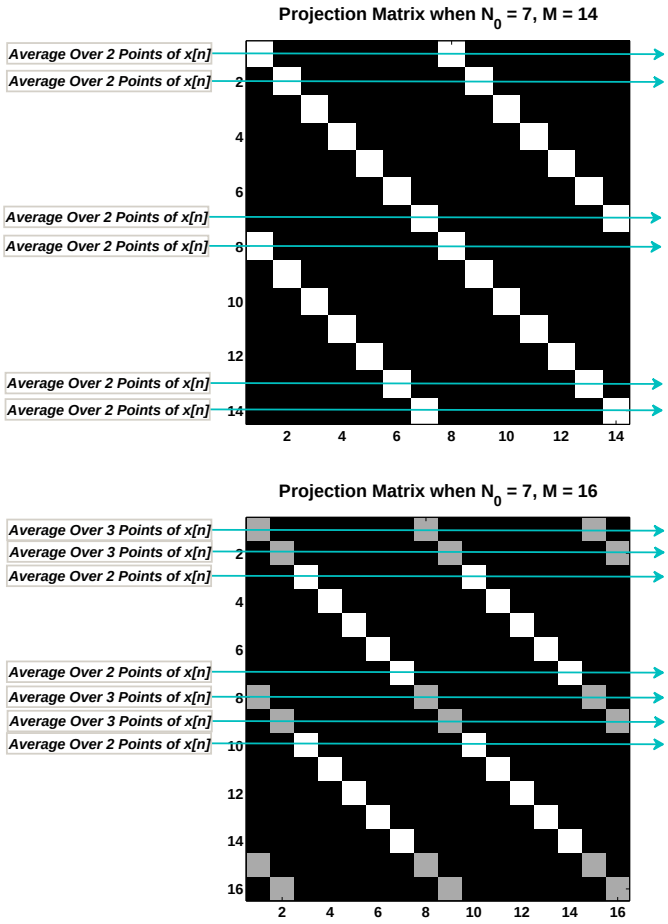

Figure 3.2: Projection matrices for the Pitch Synchronous (left) and Pitch Asynchronous (right) cases. This matrix weighs the input data signal $x[n]$, and the voiced estimate is a weighted average of the input as described by the respective projection matrix.

and X is another integer. In this case, the columns of the matrix $\mathbf{V}$ are *not* orthogonal to each other, and this results in the pseudo-inverse matrix for $\mathbf{V}$ being more complicated than in the pitch-synchronous case. The Projection Matrix $\mathbf{P_V}$ in the pitch-asynchronous case is given by [48]:

$$\mathbf{P_V}[\mathbf{m},\mathbf{n}] = \frac{N_0[n]}{M}\delta((m-n) \diamond N_0) \tag{3.23}$$

where

$$N_0[n] = \{\#\text{ of } m \text{ such that } (m-n) \diamond M_0 = 0\} \tag{3.24}$$

In this case, the operation of projecting $\underline{x}$ using the projection matrix $\mathbf{P_V}$ amounts to an averaging operation where each sample is replaced by the average of itself with all other samples within the window which are separated from it by any multiple of N_0 samples. However, as opposed to the pitch-synchronous case, the number of samples which should be averaged here is *not* always equal to p and depends on the value of sample number k within the window. Mathematically, the estimate of $\underline{\hat{s_0}[k]}$ is given as follows:

$$\underline{\hat{s_0}[k]} = \frac{1}{b_k - a_k + 1}(\underline{x[k - a_k N_0]} + \underline{x[k-(a_k-1)N_0]} + ... + \underline{x[k]} + ... + \underline{x[k+(b_k-1)N_0]} + \underline{x[k+b_k N_0]}) \tag{3.25}$$

where the values of a_k and b_k are given by

$$a_k = \text{ max integer such that } k - a_k N_0 > 0 \tag{3.26}$$

$$b_k = \text{ max integer such that } k + b_k N_0 <= M \tag{3.27}$$

A sample projection matrix for the case of $N_0 = 7, M = 16$ is shown in Fig. 3.2. It can be seen that even in the pitch-asynchronous case, the segregation process reduces to an averaging operation, though involving different number of terms for different parts of the analysis window.

To summarize, it should be noted that irrespective of the type of signal to be modeled, and the relative strength of background noise, the proposed algorithm amounts to a simple temporal averaging operation in the case of speech enhancement. Furthermore, the averaging is performed over samples which are a pitch period apart from each other - irrespective of the size of the window used for analysis. Thus, this operation is computationally very efficient. It is also very intuitive - noisy versions of speech signals are replaced by time-averaged versions. Each analsis window may contain several pitch cycles. and all these pitch cycles are averaged to yield a "common" average pitch cycle - each of the individual noisy pitch cycles is then replaced by this averaged pitch cycle. As expected, the time averaging operation results in smoothing of the noise and accentuation of the periodicity. Indeed, it is not a far leap of imagination that this operation is exactly the same as what is done in signal estimation in noise in communications [45, 36]. Fig. 3.3 compares the performance of the two methods of speech enhancement, and Fig. 3.4 clearly emphasizes the difference between the two kinds of averaging operations for the same signal as in Fig. 3.3.

On the flip side, it must be noted that the model is restricted to achieving exactly what was stated above - temporal averaging of samples a pitch period apart! This will work extremely well for signals which are made of a perfectly periodic clean signal plus noise - the averaging operation will reduce the noise and preserve the perfectly periodic signal. However, speech signals are quasi-periodic in nature and have a voice as well as unvoiced component. Furthermore, due to pitch jitter or shimmer, as also temporal modulations, the signal within each analysis window is never perfectly periodic. The proposed segregation system, therefore, will only model a component of the true speech signal, but not the entire voiced portion as expected. We will delve into this in Chapter 4, and try to also account for the aperiodic component of the speech signal in the voiced regions.

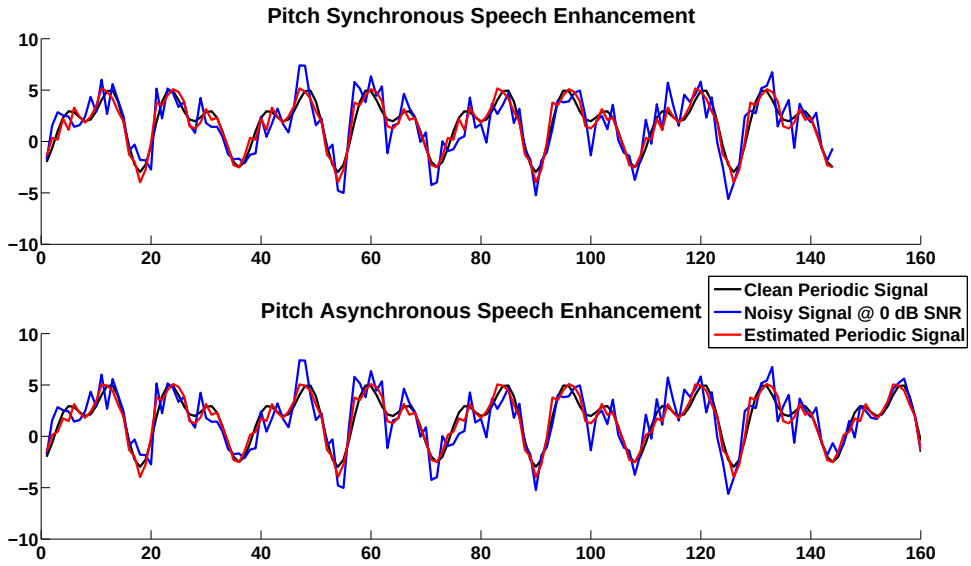

Figure 3.3: Performance of the Pitch Synchronous and Asynchronous methods of speech enhancement for the signal within a TFU. The local SNR is 0 dB.

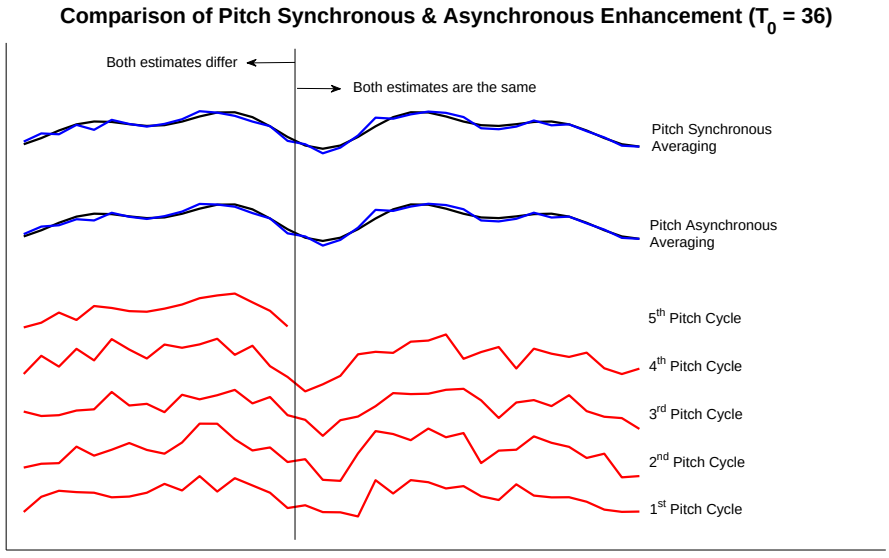

Figure 3.4: The difference between the averaging operations of the Pitch Synchronous and Asynchronous methods for speech enhancement.

3.5.3 Speech Segregation: Projection Matrices

In the case of speech segregation, the process of obtaining the individual projection matrices is more complicated. The matrix whose pseudo-inverse needs to be calculated is $\mathbf{C} = [\mathbf{V_A V_B}]$, and the estimate of the coefficients $\underline{\gamma}$ depends on both the matrix $\mathbf{C}$ *and* the data $x[n]$. Following this, the coefficients $\underline{\alpha}$ (for speaker A) and $\underline{\beta}$ (for speaker B) are picked out of $\underline{\gamma}$ by indexing - this being a non-linear operation. The final signal estimates for speakers A and B are obtained by multiplying the data matrices with these *non-linearly obtained* vectors. As such, since the coefficients are functions of both the data and the harmonic matrices, it may not be possible to obtain deterministic expressions for the projection operations $\mathbf{P}_{(\mathbf{A},\mathbf{B})\rightarrow\mathbf{A}}$ and $\mathbf{P}_{(\mathbf{A},\mathbf{B})\rightarrow\mathbf{B}}$. Furthermore, while obtaining the estimate of the speech mixture using the proposed algorithm *is* a linear (averaging) operation, the same cannot be said about the extraction of the individual components.

3.6 Application of the Proposed Model to Real-World Speech Signals

Thus far, the segregation model has been demonstrated to be theoeretically well grounded, and also useful for extraction and enhancement in noisy conditions as seen in the previous sections. However, the underlying assumption thus far has been that the signal to be modeled is periodic in nature, and that since speech is periodic in its voiced regions, the model should be applicable. The fact is that even in the voiced regions of speech, the signal is never perfectly periodic. Speech (even voiced) is quasi-periodic in nature, and due to phenomena like pitch jitter and shimmer as well as breathiness and frication, the signal being analyzed by the model is far from perfect periodicity. In addition, automatic pitch estimation can be inaccurate and in such cases, the model assumptions are invalid. Furthermore, since the signal is being processed frame-wise, there will occur some windows in which the signal exhibits rapid pitch change or significant modulations - the model will be inaccurate in such situations as well.

Due to various factors mentioned above, it is important to realize that the harmonic model is not always accurately applicable to the speech signal being analyzed, and that the deviations from the model should be taken into account somehow. In particular, we will look at a more accurate model of the speech signal in the context of speech enhancement, and extend our observations to the case of speech segregation. Consider a signal within a particular TFU during analysis (we will hereon drop the subscript "TF" in the rest of this section for the sake of brevity):

$$\begin{aligned} x[n] &= s[n] + v[n] \\ &= (s_p[n] + s_a[n]) + v[n] \\ &= s_p[n] + (s_a[n] + v[n]) \\ &= \widehat{s_p}[n] + \widehat{e_p}[n] \end{aligned} \tag{3.28}$$

where
$x[n]$ is the signal being analyzed,
$s[n]$ is the speech component in $x[n]$,
$v[n]$ is the noise component in $x[n]$,
$s_p[n]$ is the *periodic* component in $s[n]$,
$s_a[n]$ is the *aperiodic* component in $x[n]$,
$\widehat{s_p[n]}$ is the periodic component in $x[n]$ as estimated by the model, and
$\widehat{e_p[n]}$ is the residual component in $x[n]$ as estimated by the model.

Ideally, $\widehat{s_p[n]}$ should be as close as possible to $s_p[n]$, and $\widehat{e_p[n]}$ should be as close as possible to $s_a[n] + v[n]$, in which case the periodic speech component in the noisy signal has been well captured by the *model*. It should be noted that in general, it is extremely difficult to define the periodic component of a speech signal - as such, we resort to a slightly different definition for the periodic component $s_p[n]$. For a given TFU containing noisy speech $x[n]$ with speech component

$s[n]$, we define the periodic component $s_p[n]$ as the result of passing $s[n]$ through the enhancement algorithm, i.e., the signal obtained by modeling $s[n]$ using the proposed algorithm. Thus, we have a periodic component of the speech signal $s[n]$, $s_p[n]$, which we need to estimate from the noisy signal $x[n]$ to yield $\widehat{s_p[n]}$. The portion of the speech signal ($s[n]$) which is not modeled by the LS model, i.e., the residual signal $s[n] - s_p[n]$, is called the aperiodic component $s_a[n]$. While this unconventional definition of the periodic component is not always accurate, it is still useful to define periodicity thus since it now brings both the noisy and clean speech signals to the same model. Fig. 3.5 illustrates the various concepts defined here, as well as the performance of the speech enhancement algorithm, on two sample noisy signals. The top panel shows this information for the signal within a TFU from a low-frequency channel of a real-world noisy speech signal, while the bottom panel shows the same for a TFU from a high-frequency channel. The original clean speech ($s[n]$), the noise added ($v[n]$), and the resultant noisy speech ($x[n]$) are shown in the first column. In this particular example, the relative strength of noise is low in the low-frequency channel and high in the high-frequency channel. The periodic component of the speech signal (as defined in this section, $s_p[n]$) as well as the aperiodic component ($s_a[n]$), are shown in the second column along with the speech signal ($s[n]$). It is seen that since the high-frequency region of the speech signal exhibits a great deal of amplitude modulation, the periodic component estimate differs greatly from the actual speech component. This is due to the averaging performed by the proposed model - it may not be reflective of the periodic or aperiodic nature of that particular speech sample, but may instead be a consequence of the proposed model itself (we will try to compensate for this in Chapter 4 by adding back the aperiodic component to the periodic component, so that the speech component is well-preserved). The true periodic component in the speech signal as identified by the model, ($s_p[n]$) should then be recovered from the noisy speech signal by the proposed algorithm - this is shown in the third column of Fig. 3.5. Both the periodic component ($s_p[n]$) and its estimate ($\widehat{s_p[n]}$), as well as the estimation error ($\widehat{r_p[n]} = s_p[n] - \widehat{s_p[n]}$) are shown (different from $\widehat{e_p[n]}$). It may be seen that the error in estimation of the periodic component is not significantly high for either case.

At this point, we should recognize that the model proposed for speech enhancement has certain limitations regarding the signal it can model. In particular, the periodic component estimated from the noisy speech ($\widehat{s_p[n]}$) is not always equal to the periodic component ($s_p[n]$) and there is an estimation error. When the noise power is relatively small, the periodic estimate is very close to the true value as expected. However, when the noise power is comparable to the speech signal power, the estimates obtained by the model can be significantly erroneous. Therefore, we need to accept or reject the estimates obtained by the algorithm using some criteria. In the alternative, the estimate $s_p[n]$ must be appropriately scaled, so that it is accepted when reliable but scaled to a very small value when unreliable. This process of obtaining a scaled estimate $\widetilde{s_p[n]}$ from the estimate $\widehat{s_p[n]}$ was found to be perceptually more acceptable than simply rejecting it, and is decribed in more detail in Chapter 4.

In addition to the error in estimation of the periodic component, another issue needs to be accounted for: the periodic component we are aiming to estimate is itself not completely representative of the speech signal and there is an aperiodic component $s_a[n]$ which we also need to estimate. If we are able to estimate an aperiodic component $\widetilde{s_a[n]}$ from the residue signal $\widehat{e_p[n]}$ which is fairly representative of the actual aperiodic component $s_a[n]$, then this estimate can be added to the estimate of the periodic component to get the total speech signal estimate, i.e.,

$$\begin{array}{ccccc} & s[n] & = & s_p[n] + s_a[n] & \\ x[n] & \rightarrow & \widehat{s_p[n]} & \rightarrow & \widetilde{s_p[n]} \\ \widehat{e_p[n]} & = & x[n] - s_p[n] & \rightarrow & \widetilde{s_a[n]} \\ & \widetilde{s[n]} & = & \widetilde{s_p[n]} + \widetilde{s_a[n]} & \end{array} \tag{3.29}$$

Thus, the speech signal in the voiced portion $\widetilde{s[n]}$ can be estimated by combining the estimates obtained from the model and from the residue. The exact process we use to recover the unvoiced component will also be described in Chapter 4.

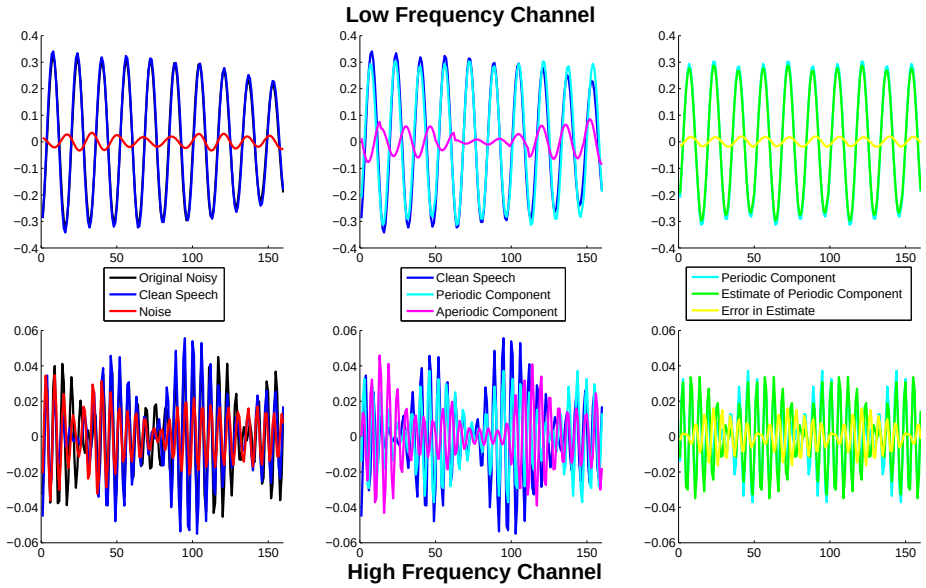

Figure 3.5: The various kinds of signals which need to be accounted for in the speech enhancement problem. The top and bottom rows illustrate the same information for a low-frequency and high-frequency channel respectively. The column shows the clean and noisy speech, as well as the noise. The second column breaks down the speech component into its periodic and aperiodic components. Finally, the third column breaks the periodic component into its estimate as obtained by the algorithm, and the estimation error.

Using a parallel reasoning, the speech segregation problem also involves the estimation of speech components of speech from speakers A and B, called $\widetilde{s_A[n]}$ and $\widetilde{s_B[n]}$ respectively, from the mixture signal $x[n]$. Similar to the flow of control identified in Eqn. 3.29, the flow of control can be described as follows:

$$\begin{array}{cccccc} & s[n] & = & s_{A_p}[n] + s_{A_a}[n] + s_{B_p}[n] + s_{B_a}[n] & & \\ x[n] & \rightarrow & (\widehat{s_{A_p}[n]}, \widehat{s_{B_p}[n]}) & \rightarrow & & \widetilde{s_{A_p}[n]} \\ x[n] & \rightarrow & (\widehat{s_{A_p}[n]}, \widehat{s_{B_p}[n]}) & \rightarrow & & \widetilde{s_{B_p}[n]} \\ \widehat{e_p[n]} & = & x[n] - (s_{A_p}[n] + s_{B_p}[n]) & \rightarrow & & (\widetilde{s_{A_a}[n]}, \widetilde{s_{B_a}[n]}) \\ & \widetilde{s_A[n]} & = & \widetilde{s_{A_p}[n]} + \widetilde{s_{A_a}[n]} & & \\ & \widetilde{s_B[n]} & = & \widetilde{s_{B_p}[n]} + \widetilde{s_{B_a}[n]} & & \end{array} \tag{3.30}$$

where the labels of the different variables are self-explanatory in context of the above discussion. The process of deriving the terms $\widehat{s_{A_p}[n]}$ and $\widehat{s_{B_p}[n]}$ from the mixture signal $x[n]$ has already been described in this section. The procedure of scaling these esimates to yield $\widetilde{s_{A_p}[n]}$ and $\widetilde{s_{B_p}[n]}$, as well as estimating $\widetilde{s_{A_a}[n]}$ and $\widetilde{s_{B_a}[n]}$ from the residual signal $\widehat{e_p[n]}$, is described in Chapter 4.

Before concluding this chapter, it is also interesting to note that the proposed approach for speech extraction is close to some other approaches proposed in the literature in the past. It is illustrative to explore how the proposed method differs from those approaches and what it offers as an advantage over the others.

3.7 Comparison of the Proposed Method to Some Similar Approaches

While there have been several different methods described to deal with the problem of speech enhancement or segregation, three approaches should be taken note on in context to the proposed algorithm. These three approaches are very close to the proposed algorithm in the sense that either the model they propose are similar to the proposed model, or the final operations of the proposed algorithm are very similar to those approaches. In all cases, we will endeavor to highlight the distinction of our proposed approach. Finally, it should also be noted that all the approaches described here are only applicable for the voiced portions of the input signal, while this thesis focuses on recovering the entire constituent speech signals (both voiced *and* unvoiced portions).

3.7.1 Bayesian Harmonic Models

[8] have proposed a harmonic model for music signals which is exactly the same as the model proposed in this algorithm. However, both methods differ distinctly in how the parameters of the model are solved for, and how the algorithms account for departure from periodicity. In particular, the model proposed in [8] for a monophonic sound is as follows:

$$x[n] \quad = \quad \sum_{i=1}^{N_0} (\alpha_i[n] \cos\omega_0(i+\delta_i)n + \beta_i[n] \sin\omega_0(i+\delta_i)n) + w[n]$$

where N_0, $\alpha_i[n]$, $\beta_i[n]$, ω_0 and δ_i are the unknown parameters to be estimated. The parameters $\alpha_i[n]$ and $\beta_i[n]$ account for the harmonic contributions across the spectrum, while ω_0 denotes the pitch frequency and N_0 the number of harmonics. The parameters δ_i are detuning parameters which account for departure of the signal from being perfectly periodic (i.e., account for quasi-periodicity). For estimation of these parameters, the distribution of these parameters is

learnt under a Bayesian setting using a large database. During runtime, these parameters are hypothesized using the models obtained through training and then post-filtered to obtain smooth estimates. Because of the large number of parameters involved, the training and estimation process for this system can be quite large, and the authors rely on Markov Chain Monte Carlo (MCMC) simulations for estimation of the probabilties, which highlights the complexity of the problem. Furthermore, in the case of separation of two sources, the number of parameters to be estimated will double, and this implies that the training process with be more time-consuming as well as require more training data (the Curse of Dimensionality). Furthermore, there is no accounting for the reliability of the estimates obtained in this approach.

The model proposed by this thesis for a mixture signal with one speaker is as follows:

$$x_{TF}[n] \quad = \quad \sum_{i=1}^{N_0}(\alpha_i e^{j\omega_0 in}) + w[n]$$

where the only set of parameters to be estimated is α_i. The pitch frequency ω_0 is estimated using a completely different algorithm (thus also yielding N_0), and that decouples the complex problem into a set of two different simpler problems. In particular, the solution obtained by this approach can be simplified to a simple averaging operation as illustrated in Section 3.5. Finally, the model accounts for the quasi-periodic nature of speech, as well as reliability of the estimates, using a completely different approach. While the approch does involve learning mappings from a 3-D space to a 1-D space, the process is still very less expensive than the one proposed in [8]. Thus, while the proposed models are same, the approach to solving for the parameters and using them for the application is much different in this thesis.

3.7.2 McAulay-Quatieri Model

The model for the speech signal as proposed in [38] is the same as the one proposed in this thesis. In this case as well, the set of unknown parameters to be estimated inludes the pitch, while in the proposed thesis the pitch is estimated by a different algorithm. [38] solve for the unknown coefficients in the frequency domain instead of in the time-domain as done in this thesis. The coefficients are solved for by sampling the spectrum of the mixture signal at the harmonic locations to yield the individual spectral amplitudes and phases at those harmonic locations. In order to unwrap the phase before usage for reconstruction, continuity constraints are maintained from the beginning of the signal to the current analysis frame. On the other hand, the proposed method solves for both the amplitudes and phases in the time-domain, and more specifically does not require the tracking of phase information for reconstruction. In other words, the analysis of each frame is independent of the previous frame. Furthermore, the model proposed in [38] models the mixture signal across the spectrum, while we model the signal within a TFU. Finally, we also rely on a measure of reliability of the estimates to decide if the estimated signals should indeed be used for reconstruction.

3.7.3 Harmonic Enhancement and Cancellation Models

Harmonic enhancement or cancellation methods are approaches towards separating speech signals on the basis of their harmonic nature [9]. The set of harmonic enhancement technqiues attempt to increase the strength of the harmonics of a particular speaker relative to that of the other. This is achieved in the time domain by processing the signal through a comb filter defined by the impulse response:

$$h[n] \quad = \quad \frac{1}{K}\sum_{k=0}^{K-1}\delta(n - kT_A)$$

where T_A is the pitch period of speaker A. As can be seen, this operation is exactly the same as the averaging operation discussed in Section 3.5.1. However, a major difference is that while

this operation is optimal (in the sense of minimization of the mean square error (MSE)) for the case of speech enhancement, it is not the case for the case of speech segregation. Indeed, as we observed in Section 3.5.3, the projection matrix, and hence the method of temporal averaging, in the case of speech segregation is data-dependent. As such, in the sense of minimizing the MSE, the proposed algorithm is superior to the harmonic enhancement methods for speech segregation since the latter are non-optimal for the task. In addition, as we have seen in Section 3.5.2, the optimal averaging operation depends on the length of the analysis window. The set of harmonic enhancement methods are thus only optimal in the pitch-synchronous case. Finally, the filtering operation is usually performed on the entire speech signal, while in this thesis we propose to do it on a channel-wise basis for better perceptual quality.

The harmonic cancellation methods are closely related to the harmonic enhancement methods. The goal here is to reduce the relative amplitude of one speaker with respect to the other, so that the other speaker becomes the dominant one. This operation is achieved by a filtering operation defined as:

$$h[n] \quad = \quad \tfrac{1}{2}\,[\delta(n) - \delta(T_A)]$$

Since this is another kind of averaging operation, the comparisons between the proposed method and harmonic enhancement models also apply here. In particular, this set of approaches is also non-optimal for the segregation of speech signals.

3.8 Chapter Summary

In this chapter, we discussed the algorithm to separate voiced components of competing speech signals. We also demonstrated the applicability of this algorithm to extract the voiced regions of speech signals in the presence of noise. We then demonstrated that the proposed algorithm is intuitively very simple and appealing for the case of speech enhancement, and that the speech segregation case is not as simple. We next highlighted the various components of the speech signal that have been modeled by the proposed algorithm, and what other components remain to be modeled and recovered. We will now cover these in the following chapter.

Chapter 4

RECOVERY OF THE APERIODIC REGIONS FROM NOISY SPEECH MIXTURES

4.1 Introduction

In this chapter, we will cover several important aspects of the speech extration algorithm. In particular for the voiced regions of the noisy speech mixtures, we will explore how to modify the estimates of the periodic components (i.e., the components of the speech signal which will be captured by the proposed model) according to their reliability. We will also explore how to capture the aperiodic components (i.e., the components of the speech signal which will not be captured), as well as the noise components, in the voiced regions. We will next explore the use of these aperiodic and noise estimates to estimate the unvoiced energy and noise energy in the unvoiced regions of the speech signal. These various issues will be discussed in the context of both speech enhancement and speech segregation. Following this, we will demonstrate the utility of the whole speech extraction approach on several noisy speech mixtures, illustrating the performance for both the enhancement and segregation problems.

For the sake of simplicity, we will begin with the case of speech enhancement (i.e., one voiced speaker) as we did in Chapter 3. Having made several critical observations and concluded a rational approach for improving the performance in the case of enhancement, we will then proceed to generalize this approach for the case of speaker segregation.

4.2 Effect of the Local SNR on the Speech Enhancement Problem

We begin by trying to understand the operation of the algorithm for noisy speech under different signal-to-noise ratios (SNRs). As was discussed in Section 3.5, the speech enhancement operation reduces to a simple averaging operation - the effect of this averaging operation is now explored in different SNR settings. Let us consider a TFU containing a speech signal $s_{TF}[n]$ in the presence of noise $v_{TF}[n]$ to give the observation $x_{TF}[n]$. For a window of length M, these sets of values can be vectorized to give us the set of variables $\underline{s}$, $\underline{v}$ and $\underline{x}$ respectively, where we are also dropping the subscript $_{TF}$ and noting that the discussion in the rest of this section applies to the activity within a TFU of the signal. Furthermore, without loss of generality, let us assume that the signals $\underline{s}$ and $\underline{v}$ are defined so that they have the same power (in case $\underline{v}$ does not equal $\underline{s}$ in power, it can be scaled to equal the other in power, and this new signal can now be labeled $\underline{v}$). Then, if the local SNR in the TFU is given by $\frac{1}{\lambda^2}$ ($\lambda < 1$) in the linear scale, the signals can be written as:

$$\begin{aligned} \underline{x}_\lambda &= \underline{s} + \lambda \underline{v} \\ &= \underline{s}_p + \underline{s}_a + \lambda \underline{v} \end{aligned}$$

where, as described in Section 3.5.3, the signal $\underline{s}_p$ is that component of the speech signal which can be modeled by the harmonic model, and $\underline{s}_a$ is the component of the speech signal which cannot be

modeled. From the discussion in Section 3.5, in terms of the Projection Matrix $\mathbf{P}$, these can be written as follows:

$$\begin{aligned} \underline{s}_p &= \mathbf{P}\underline{s} \\ \underline{s}_a &= \underline{s} - \underline{s}_p \\ &= \underline{s}_p - \mathbf{P}\underline{s} \\ &= (\mathbf{I} - \mathbf{P})\underline{s} \end{aligned}$$

Thus, upon speech enhancement using the proposed method operating on the noisy signal $\underline{x}$, we have the estimate of the periodic component from the noisy speech signal as:

$$\begin{aligned} \hat{\underline{s}_{p_\lambda}} &= \mathbf{P}\underline{x}_\lambda \\ &= \mathbf{P}(\underline{s} + \lambda\underline{v}) \\ &= \underline{s}_p + \lambda\mathbf{P}\underline{v} \\ &= \underline{s}_p + \lambda\underline{v}_p \end{aligned}$$

where the term $\mathbf{P}\underline{v}$ can be written as $\underline{v}_p$, i.e., the periodic component of the noise vector. The estimate of the residue sidgnal upon enhancement can be written as

$$\begin{aligned} \hat{\underline{s}_{a_\lambda}} &= \underline{x}_\lambda - \hat{\underline{s}_{p_\lambda}} \\ &= (\underline{s}_p + \underline{s}_a + \lambda\underline{v}) - (\underline{s}_p + \lambda\mathbf{P}\underline{v}) \\ &= \underline{s}_a + \lambda(\mathbf{I} - \mathbf{P})\underline{v}_p \\ &= \underline{s}_a + \lambda\underline{v}_a \end{aligned}$$

where the term $(\mathbf{I} - \mathbf{P})\underline{v}$ is written as $\underline{v}_a$ in parallel with the above terms. Summarizing the important equations:

$$\begin{array}{cccc} \text{Given speech} & \underline{s} & = & \underline{s}_p + \underline{s}_a \\ \text{and noise} & \underline{v} & = & \underline{v}_p + \underline{v}_a \\ \text{then for noisy speech} & \underline{x}_\lambda & = & \underline{s} + \lambda\underline{v} \\ \text{we have} & \hat{\underline{s}_{p_\lambda}} & = & \underline{s}_p + \lambda\underline{v}_p \\ \text{and} & \hat{\underline{s}_{a_\lambda}} & = & \underline{s}_a + \lambda\underline{v}_a \end{array} \tag{4.1}$$

This set of very simple equations gives us a very intuitive understanding of the enhancement process - the periodic component as estimated from the noisy signal is equal to the true periodic component of the speech signal, plus a noisy periodic component which is weighted by the factor λ. Similarly, the aperiodic component as estimated from the noisy signal is equal to the true aperiodic component of the speech signal, plus a noisy aperiodic component which is weighted by the same factor λ. Since the enhancement operation is a simple averaging operation, it is obvious that as the amount of noise λ increases, the estimate $\hat{\underline{s}_{p_\lambda}}$ will be farther away from the true speech component $\underline{s}$. In fact, if λ is small, then $\hat{\underline{s}_{p_\lambda}} \simeq \underline{s}_p$ and if λ is very large, then $\hat{\underline{s}_{p_\lambda}} \simeq \lambda\underline{v}_p$. For intermediate values of λ, the estimate $\hat{\underline{s}_{p_\lambda}}$ is an averaged version of both signals. But in all cases, the amplitude of $\hat{\underline{s}_{p_\lambda}}$ will be greater than that of $\underline{s}_p$. Similar observations apply for the aperiodic component and its estimate as well. Importantly, it must be noted conditioned on a *given* realization of the noise variable $\underline{v}$, the effect of the SNR λ is linear on the estimate of the periodic component. This means that if we want to compare the performance of the enhancement algorithm on two different noisy signals $\underline{x}_1$ and $\underline{x}_2$ which are composed of the same speech vector $\underline{s}$ and same noise vector $\underline{v}$ and only differ in the values of λ (say λ_1 and λ_2), then the estimate corresponding to the higher SNR (lower λ) will be the more "reliable" one in the sense that the estimate of the periodic component will be closer to the true periodic component for that case. For a given realization of the noise $\underline{v}$, how much improvement does the model afford? The answer is not determinstic and is data-dependent but intuitively it can be said that the SNR of the periodic estimate will be *better* than that of the original noisy signal. The reasoning for this is as follows: in the case of the processed signal, the noise has been averaged using a pitch-based matrix. Therefore, under the (safe) assumption that the noise does not temporally correlate at the same temporal rate as the speech signal, the averaging operation should reduce the power of the noise in the estimate, i.e., $||\mathbf{P}\underline{v}||^2 < ||\underline{v}||^2$ while keeping the power of the periodic component about the same (since the speech signal correlates at the pitch period) - thus increasing the signal to noise ratio.

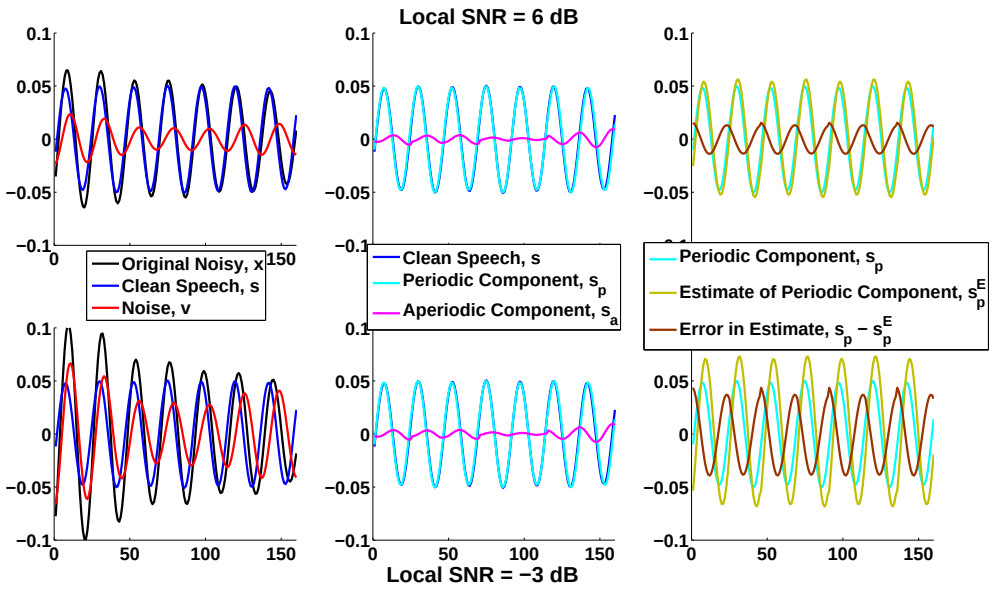

Figure 4.1: Comparison of the speech enhancement process for two different SNRs. The top row contains figures for the case of SNR = 6 dB, while the bottom contains figures for the case of SNR = 3 dB. The information in each plot is indicated in the legend.

Fig. 4.1 illustrates these observations. The top panel displays information for a signal with local SNR 6 dB, while the bottom panel displays information for a signal with local SNR -3 dB. The estimates for the periodic components in both cases are obtained from temporal averaging of the black plot, while the actual true periodic components are obtained by the temporal averaging of the blue plots. The resulting signals are shown in green and cyan respectively, with the error between these two signals shown in brown. It can be seen firstly that the shape of the brown line is preserved in the case of the different SNRs and one version is simply a scaled version of the other, confirming that the error between $\underline{s}$ and $\underline{s}_p$ is simply proportional to the strength of the noises which were added to the speech signals (and thus that the relative scaling of the estimation errors is the same as the relative scaling of the noises). Secondly, it can be seen that in the case of SNR = 6 dB, the estimation error (brown line) and the noise (red line) levels are quite similar to each other, and that the SNR between the speech signal (blue, $\underline{s}$) to noise (red lines, $\underline{v}$), actually 6 dB, is similar to the SNR between the true periodic component (cyan, $\underline{s}_p$) and the error in estimation of the periodic component (brown, $\underline{s} - \underline{s}_p = \lambda \mathbf{P}\underline{v}$) which is 6.3 dB. In the case of SNR = -3 dB, the SNR between the speech signal and noise is -3 dB, while the SNR between the true periodic component and the error component is much higher, actually equalling 0.3 dB (compare the levels of the brown and cyan lines in the lower panel). This illustrates the fact that the proposed segregation system *does* provide us with better SNR by averaging out the noise. As mentioned above, this method works because the speech signal is highly correlated at the pitch period, while the noise is not necessarily correlated at that pitch - therefore averaging will inevitably reduce its relative influence on speech.

For different realizations of the noise vector $\underline{v}$ conclusions are more difficult to arrive at. This is due to the factor $\mathbf{P}\underline{v}$. Since this factor is an averaging operation at a certain pitch period, the resultant signal will depend very much on the temporal behavior of $\underline{v}$. A little thought on comparing the operation $\mathbf{P}\underline{v}$ and $\mathbf{P}\underline{v}_{permute}$, where $\underline{v}_{permute}$ is a permuted version of the elements of $\underline{v}$ indicates very clearly that the averaging operation is data-dependent. As such, for a signal $\underline{x}$

consisting of the same speech signal $\underline{s}$ with the same SNR λ but two different noise realizations $\underline{v}_1$ and $\underline{v}_2$, the algorithm would yield two different estimates $\hat{\underline{s}_{p}}_1$ and $\hat{\underline{s}_{p}}_2$ for the periodic component of the speech signal, one of which may be more reliable (closer to the actual periodic component) than the other.

4.2.1 Reliability of Estimates: Dealing with Real-World Speech Signals

We will now try to understand how reliable the estimates are using our proposed algorithm, and whether we can modify our estimates from the algorithm to be perceptually more relevant and preferable than the raw estimates given by the algorithm. In particular, we will take note of the observation that since speech signals are temporally smooth in the voiced regions, the energy of the periodic component $\underline{s}$ is expected to change slowly with time and also across frequency channels. On the other hand, most kinds of noise typically have no inherent structure in then, and as such the energy of the periodic component of the noise vector $\underline{v}_p$ could vary rapidly over time and frequency. The estimates of the speech signal will therefore also exhibit rapid fluctuations spectro-temporally while clean speech would not exhibit such tendencies. In fact, the greater the level of noise or more rapid its fluctuations, the greater its effect on the estimates and thus on the reconstructed speech signal and the farther our speech estimates are from the truth. In reality, the scale of the periodic estimate is dependent on the level of the noise which renders the enhancement algorithm sensitive to the noise power. Therefore, if there were a mapping function to inform us of the true periodic power of the clean speech signal in each TFU, then the estimated periodic signal could be scaled appropriately so that at least in power (if not in exact temporal structure), the estimated signal would be close to the ideal clean signal.

We would thus ideally like to have an algorithm which scales the estimates such that (a) in case of low noise (or high SNR), the estimates are known to be reliable and used as predicted by the model, and (b) in case of high noise (low SNR), the estimates are scaled down so that they come to the expected level of the original speech signal. In effect, this kind of a mapping would match the powers of the estimated and true periodic components of the speech signal. That ensures that in low SNR conditions, while the estimate would be comparable to the level of the noise signal, the scaled estimate after mapping would be comparable to the level of the speech signal albeit with some noise content. Essentially this would reduce the local effect of the noise on the global speech signal, and could thus perceptually enhance the quality of speech since the adjacent spectro-temporal regions would then be comparable to the present TFU and thus provide enough masking to drown the noise. The enhancement algorithm described in Chapter 3 would therefore attempt to preserve the temporal structure of the speech signal, while the mapping function alluded to here would help to preserve the relative amplitudes of the signal in various spectro-temporal regions, and also simultaneously help to modify the estimates according to their reliability. Thus, we expect to benefit a lot from a function $f(.)$ which when operated on the estimate $\hat{\underline{s}_{p_\lambda}}$ would yield the true periodic power of the signal $\underline{s}_p$, i.e., $f(\hat{\underline{s}_{p_\lambda}}) = ||\underline{s}_p||^2$.

Similarly, if we consider the residual signal $\hat{\underline{s}_{a_\lambda}}$ which is composed of both the aperiodic component of the speech ($\underline{s}_a$) as well as of the noise ($\underline{v}_a$), we see that a similar observation implies that the residue is closer to the true aperiodic component of the speech signal when the noise power is small, and is much higher than the true aperiodic component when the noise power is high. Therefore, a scaling function that maps the aperiodic estimate to its true value could similarly improve the overall speech signal estimate by preserving spectro-temporal smoothness and at the same time weighting the estimate by some measure of its reliability.

If we therefore consider a sequence of operations as suggested below:

- Estimate $\hat{\underline{s}_{p_\lambda}}$ from $\underline{x}_\lambda$ using the enhancement algorithm - call it the estimate of the periodic component
- Estimate $\hat{\underline{s}_{a_\lambda}}$ from $\underline{x}_\lambda$ and $\hat{\underline{s}_{p_\lambda}}$ - call it the estimate of the aperiodic component

- Using a mapping function $f(\underline{\hat{s}}_{p_\lambda}) \rightarrow ||\underline{s}_p||^2$, scale the periodic estimate so that its power is now equal to the power of the true periodic component
- Using a mapping function $g(\underline{\hat{s}}_{a_\lambda}) \rightarrow ||\underline{s}_a||^2$, scale the aperiodic estimate so that its power is now equal to the power of the true aperiodic component
- Add the modified periodic and aperiodic components and call it the speech signal estimate

then we can hope to obtain a speech signal estimate which contains both the periodic and aperiodic components of the clean speech signal, and weighted in an appropriate way so that the noise content in these signals is reduced as much as possible. If done right, this speech signal estimate would then contain less noise than the original noisy speech signal, and would therefore be more preferable perceptually. With well-defined functions, this process would ensure that (a) reliable estimates are made perceptually more significant by scaling with a value close to unity, and (b) unreliable estimates are made perceptually less significant by scaling with a very small value.

Let us first introduce some symbols (we will drop the subscript λ for simplicity, but assume its presence throughout the discussion) and also make a few observations:

- Noisy Signal Power, $E_t = ||\underline{x}||^2$
- Periodic Speech Power, $E_{\underline{s}_p} = ||\underline{s}_p||^2$
- Aperiodic Speech Power, $E_{\underline{s}_a} = ||\underline{s}_a||^2$
- Periodic Noise Power, $E_{\underline{v}_p} = \lambda^2||\underline{v}_p||^2$
- Aperiodic Noise Power, $E_{\underline{v}_a} = \lambda^2||\underline{v}_a||^2$
- Periodic Estimated Power, $E_{\underline{\hat{s}}_p} = ||\underline{\hat{s}}_p||^2$
- Aperiodic Estimated Power, $E_{\underline{\hat{s}}_a} = ||\underline{\hat{s}}_a||^2$
- Hadamard Signal, $\underline{\hat{s}}_{pa} = \underline{\hat{s}}_p \circ \underline{\hat{s}}_a$, i.e., the signal obtained by pointwise multiplication of the estimated periodic and aperiodic components
- Hadamard Estimated Power, $E_{\underline{\hat{s}}_{pa}} = ||\underline{\hat{s}}_{pa}||^2$

It may be observed that $E_{\underline{s}_p}$, $E_{\underline{s}_a}$, $E_{\underline{\hat{s}}_p}$ and $E_{\underline{\hat{s}}_a}$ are related to each other:

$$E_{\underline{\hat{s}}_p} = ||\underline{\hat{s}}_p||^2 = ||\underline{s}_p||^2 + \lambda^2||\underline{v}_p||^2 + 2\lambda < \underline{s}_p, \underline{v}_p >= E_{\underline{s}_p} + E_{\underline{v}_p} + 2\lambda < \underline{s}_p, \underline{v}_p >$$

$$E_{\underline{\hat{s}}_a} = ||\underline{\hat{s}}_a||^2 = ||\underline{s}_a||^2 + \lambda^2||\underline{v}_a||^2 + 2\lambda < \underline{s}_a, \underline{v}_a >= E_{\underline{s}_a} + E_{\underline{v}_a} + 2\lambda < \underline{s}_a, \underline{v}_a >$$

This pair of equations illustrates the discussion in this section: the power of the estimated periodic component is greater than the sum of the true periodic component, and depending on the value of the SNR $\frac{1}{\lambda^2}$, the two power values could be significantly different from each other. In context of that discussion, we would like to find an appropriate scaling that would bring the power of the periodic estimate from $E_{\underline{\hat{s}}_p}$ to $E_{\underline{s}_p}$, and similarly that of the aperiodic estimate from $E_{\underline{\hat{s}}_a}$ to $E_{\underline{s}_a}$.

The relation between $E_{\underline{s}_p}$ and $E_{\underline{\hat{s}}_p}$ is obvious from the equation, as is the relation between $E_{\underline{s}_a}$ and $E_{\underline{\hat{s}}_a}$. The relation between $E_{\underline{\hat{s}}_a}$ and $E_{\underline{\hat{s}}_p}$ stems implicitly from the fact that $E_{\underline{s}_a}$ and $E_{\underline{s}_a}$ are related to each other since one of these signals is the **P**-space complement of the other. Thus, one can re-write the above equations as:

$$\begin{array}{rcl} E_{\underline{\hat{s}}_p} & = & \phi_1(E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}) \\ E_{\underline{\hat{s}}_a} & = & \phi_2(E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}) \end{array} \tag{4.2}$$

Given the knowledge of a clean speech signal $\underline{s}$ and the *exact* noise (i.e. its temporal nature) which is added to the signal $\lambda\underline{v}$, the powers of the estimates from the enhancement algorithm can thus be

calculated from the above functions. However, in our case, we are looking at the inverse problem - given the estimates and their powers, we would like to identify the true values of the periodic and aperiodic components of the signal and noise powers. The observed variables in our case are the periodic estimate $\underline{\hat{s}}_{p_\lambda}$ and the aperiodic estimate $\underline{\hat{s}}_{a_\lambda}$. It is these elements which we want to modify, and the modification we want is a function of those very elements which we want to modify *to* (and which are not available to us). We would ideally like to find:

$$(E_{\underline{\tilde{s}}_p}, E_{\underline{\tilde{v}}_p}, E_{\underline{\tilde{s}}_a}, E_{\underline{\tilde{v}}_a}) = \varphi(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a})$$

as an inverse of the set of equations in 4.2.

The proposed idea here is: once these mapped power values are known, the estimates obtained from the algorithm will then be modified as the following:

$$\begin{aligned} \underline{\tilde{s}}_p &= \sqrt{\frac{E_{\underline{\tilde{s}}_p}}{E_{\underline{\hat{s}}_p}}}\underline{\hat{s}}_p \\ \underline{\tilde{s}}_a &= \sqrt{\frac{E_{\underline{\tilde{s}}_a}}{E_{\underline{\hat{s}}_a}}}\underline{\hat{s}}_a \end{aligned}$$

This operation will ensure that the resultant estimates have the same power as the true components of the speech signal. Thus, for unreliable estimates (which have $E_{\underline{\tilde{s}}_p} \gg E_{\underline{\hat{s}}_p}$) the scale factor will be very small and thus the estimate of the algorithm will be scaled to a very small range - a range which is comparable to that of the true speech component. For reliable estimates (which have $E_{\underline{\tilde{s}}_p} \simeq E_{\underline{\hat{s}}_p}$) the scale factor will be close to unity and thus the estimate is passed almost with no modification. Furthermore, by ensuring that the power of the estimates is equal to the true power, we ensure that on a global spectro-temporal scale, the power of the estimates are similar to what should naturally be expected for speech. Thus, we are able to preserve the perceptual properties of speech with this kind of a modification/scaling of the estimates.

Fig. 4.2 shows an example of a speech signal processed by the proposed algorithm. As was mentioned in Chapter 1, the process described in Chapter 3 and herein are applied for each TFU of the analysis. When all the TFUs are processed, the resulting estimates are combined across frequency by using a reconstruction filterbank, and then across time by the overlap-add method. The figure shows the result after processing the entire noisy speech signal thus. The top panel shows the noisy speech signal at an SNR of 3 dB. The second panel shows the spectrogram of only the voiced portions of the clean speech signal (if must be noted that according to the model, this voiced portion itself will have a periodic and aperiodic component). This voiced portion of the clean speech signal is what we would ideally like to achieve from our algorithm. The third panel shows the output of the algorithm after being processed by the system described in Chapter 3 but without any scaling as described here. Instead, the regions where the residual is stronger than the estimate have been attenuated by 20 dB to remove excessive noise. The fourth panel shows the output of the algorithm if the estimates of each TFU were scaled as decribed in this section. For this example, the true power values $(E_{\underline{s}_p}, E_{\underline{s}_a})$ were used to scale the estimates. We will later see the output when the power values $(E_{\underline{\tilde{s}}_p}, E_{\underline{\tilde{s}}_a})$ are estimated from $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a})$ as discussed in section 4.3. An important observation to make here is that the estimate with no scaling shows many spectro-temporal regions of local spikes in energy, and the overall spectro-temporal energy map is not smooth. Furthermore, it does not match well with the spectro-temporal energy map of the clean signal. Both these problems are alleviated in the case of the estimate after scaling, as shown in the fourth panel. In particular, the energy map is very close to that of the clean signal, especially in the low-frequency regions. This allows for great improvement in the percpetual quality of the reconstructed signal, all the way down to -6 dB SNR. Thus, the method of scaling the estimates according to a reliability function is indeed highly recommended.

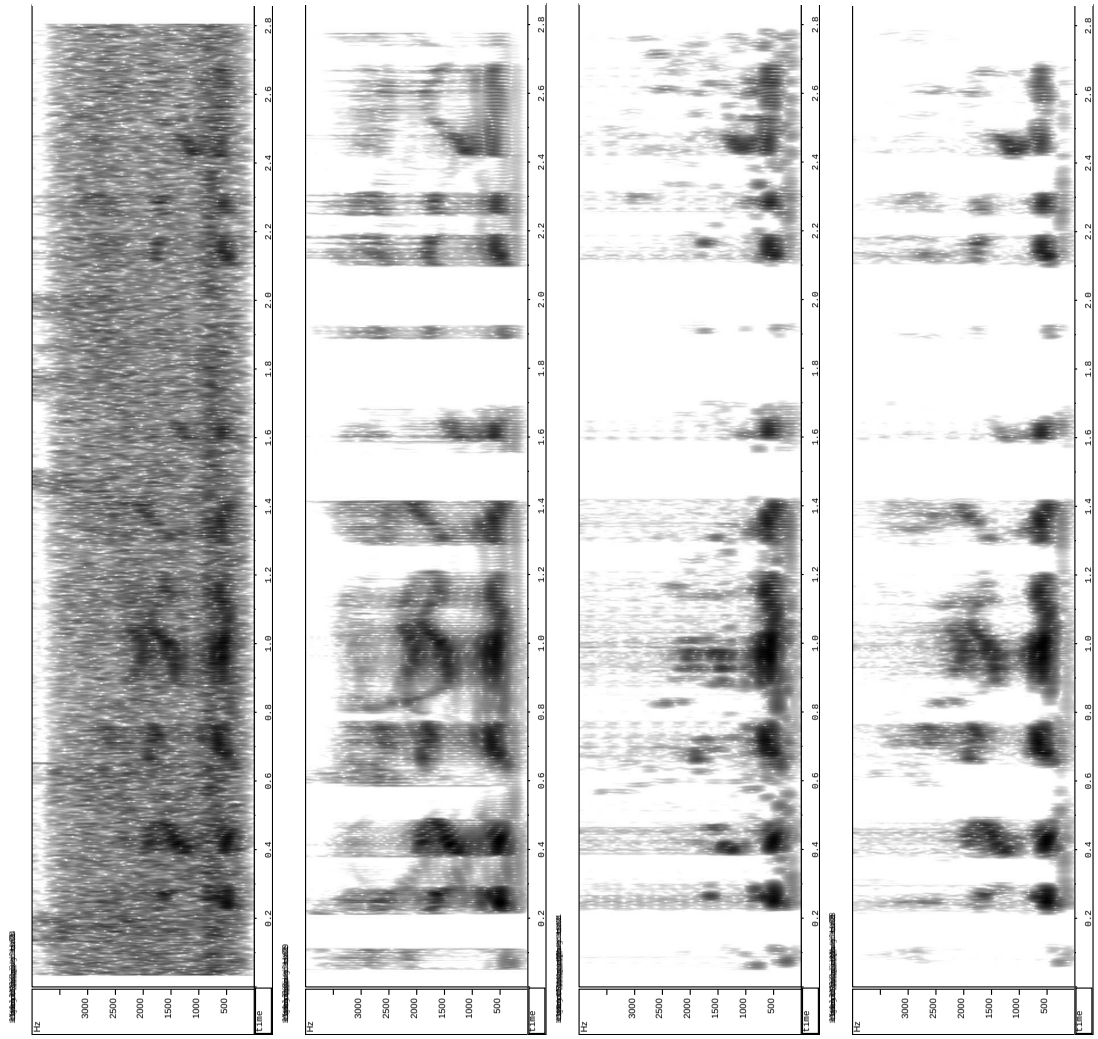

Figure 4.2: Noisy sample speech signal at 3 dB SNR, and its processed versions. (First Panel) Noisy speech signal, (Second Panel) Clean speech signal (only voiced portions), (Third Panel) Enhanced version of the speech signal after processing by the proposed algorithm but without any scaling, (Fourth Panel) Enhanced version if the individual TFUs were scaled according to match the true periodic power in the speech signal

The question now remains about identifying the mapping function $\varphi(.)$ which will take us from the space of estimated powers to the space of true powers. As pointed out, this function is non-determinstic due to the nature of the problem. This can be traced back to the fact highlighted in the last paragraph of 4.2 - due to the nature of the operation $\underline{v}_p = \mathbf{P}_{\mathbf{v}}\underline{v}$ being a data-dependent, non-determinstic one, the set of equations 4.2 is actually a non-determinstic function which varies with the noise samples. For different realizations of the vector $\underline{v}$ with same statistical properties (including power), keeping all the speech components unchanged, the estimates and their powers would still be different since they would be functions of the realization of noise.

Finally, it is also easy to imagine that there could be several combinations of true powers ($\{E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}\}$) which could yield the same estimated powers ($E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}$). As a simple example, let us recollect that the periodic estimate is simply the sum of the periodic component of the speech and the periodic component of the noise. If the periodic speech component is multiplied by 2 and the periodic noise component reduced by an appropriate amount, the estimated periodic power would still be the same value. Is it not clear how this would affect the aperiodic estimates as well (i.e., will the aperiodic powers be distinct in that case), but it would be incorrect to assume the mapping to be unique when the answer is unclear.

Summarizing, the following are the major issues to be taken note of regarding the process of scaling the estimates:

- during run-time (i.e., while performing enhancement), only the estimated powers are known, not the true powers to which we need to scale them,
- the problem is complicated by the fact that different realizations of noise can cause the same true noise power to yield different estimated noise powers, i.e., the mapping $\phi : \{E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}\} \rightarrow \{E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}\}$ is a one-to-many mapping,
- furthermore, the mapping $\varphi : \{E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}\} \rightarrow \{E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}\}$ could also be potentially a one-to-many mapping

4.3 Learning Important Parameters for Recovery of the Periodic and Aperiodic Components in Voiced Speech

Due to the complex, non-determinstic relationship between the estimated power (observed) variables ($E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}$) and the true power (unobserved) variables ($E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}$), we might consider resorting to machine learning techiques to find the mapping between the two sets of variables. Before we start, we realize that we have only 2 parameters in the observed space, from which we want to estimate 4 different parameters in the unobserved space - due to the already complicated nature of the problem, we should expect that such a mapping from 2-D to 4-D space may not be very well-learnt by any machine learning algorithm and that we would do better to add more dimensions to the observed space. Since the only observations we have are the vectors $\underline{x}$, $\underline{\hat{s}}_p$ and $\underline{\hat{s}}_a$, we can only use some combination of these to increase the dimensionality of the observed data. As such, we report to the use of the Hadamard product between the two estimates $\underline{\hat{s}}_p$ and $\underline{\hat{s}}_a$, denoting it by $\underline{\hat{s}}_{pa}$ and in particular, use its energy $E_{\underline{\hat{s}}_{pa}}$ as the third parameter in our observed space. Thus, we want to learn a function $\varphi : \{E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}}\} \rightarrow \{E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}\}$.

In order to learn the nature of such a function, we will first need several examples of the behavior of the inverse function $\phi(.)$, so that we can identify what values of the true parameters lead to which estimated values. As such, we first need to create a large database of training examples from which we can then learn the behavior of $\phi(.)$ and $\varphi(.)$. We do this by taking several clean speech signals (spanning both genders and a large population including 600 speech files), and adding several noise types to each of these signals (here, 25 different noise types including

car, wind, helicopter, subway, music, siren etc. were included) at various SNRs (-9, -6, -3, 0, 3, 6, 9 and 100 dB). For each of these different combinations of {speech, noise type, SNR} we find the values of the 7 parameters of interest, namely, $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}}, E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a})$. Thus, this *training phase* gives us a considerably large database describing the joint behavior of these paramters under various conditions.

During the *learning phase*, the mapping which takes us from the 3-D space of known paramters to the 4-D space of unknown paramters is learnt using the training data generated above. In particular, we have available with us the 3-D data of the observed variables and the corresponding 4-D data of the unobserved variables for these training samples. Using this data, we can learn the mapping through one of several possible approaches. Let us call the mapping that we will thus learn as $\tilde{\varphi}(.)$.

During the *testing phase or runtime*, we will have available with us the triplet of observed variables, and we also have available with us the function $\tilde{\varphi}(.)$ which we have learnt from the training data. Thus, we use this function in concert with the triplet of observations and find the mapped values for the unobserved variables, namely the true periodic and aperiodic speech and noise powers. Once these values are available from the function, we then know the scaling factors which must be applied to each of the estimates - that takes us from the raw enhanced speech to a perceptually more preferable one.

An important issue to be highlighted here is the dependence of the function $\tilde{\varphi}(.)$ on the speech signal itself. As has been described, the set of unobserved parameters (true energies) includes the noise estimates, which are functions of the temporal structure of the noise as well as the pitch period T_0 at which they have been averaged. Furthermore, the speech estimates are also functions of T_0 since the projection matrix $\mathbf{P_V}$ implicitly depends on T_0. As such, the function which maps the set of observed data to the set of unobserved data should implicitly depend on T_0 as well, since both the observed and unobserved data depend on it. In addition, different frequency channels exhibit different characteristics for the periodic and aperiodic components. For example, the low frequency channels usually exhibit strong periodicity from speech and therefore are more robust to noise even at low SNRs - as such the periodic and aperiodic estimates in those frequencies may be closer to the true values and the scaling required may be close to unity. On the other hand, since the high frequency regions typically have less speech energy, they may be more susceptible to noise and thus the enhancement may result in less reliable estimates - requiring a low scale factor which would attenuate the estimates. This reasoning implies that the function $\tilde{\varphi}(.)$ should also be a function of the frequency channel on which it is operating. Thus, accurately speaking, we need several mapping functions $\tilde{\varphi}_{C_0,T_0}(.)$ (where C_0 is the frequency channel and T_0 is the pitch period of the speech signal) to map the observations to the true values. During the training phase, this is accounted for, and the data is partitioned according to the pitch period and the frequency of operation. Each partition has its own respective mapping function. Similarly, during runtime, the knowledge of the pitch period and the channel of operation helps decide which mapping function to use, and the true power values are estimated using this selected mapping function. In the rest of this chapter, since the same discussion applies to all the different mapping functions (except, possibly, the exact values of various parameters) we will drop the subscripts $_{C_0,T_0}$ for the sake brevity, and include it only when there is scope for confusion. However, it must be remembered at all times that these mapping functions are indeed different for different pitch periods and different channels.

There are several ways the mapping $\tilde{\varphi}(.)$ could be learnt from the training data. For example, we could rely on a statistical approach [3], where the probability distribution function (PDF) of the unobserved variables conditioned on the observed variables, i.e., $pdf(E_{\underline{s}_p}, E_{\underline{v}_p}, E_{\underline{s}_a}, E_{\underline{v}_a}) \,|E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})$ is learnt from the data. During runtime, if we values of the observed variables as $(E_{\underline{\hat{s}}_p} = a, E_{\underline{\hat{s}}_a} = b, E_{\underline{\hat{s}}_{pa}} = c)$, then the optimal estimate for the set of unknown paramters (in the MSE sense) is given by the Maximum Likelihood Estimate of the parameters, i.e., the set of parameters which maximizes this conditional PDF. More details can be found in standard textbooks on Estimation Theory (c.f. [36]) or Machine Learning [3]. However, the parameter set exists in a 4-D space and finding the set of optimal parameters which maximizes the PDF involves a 4-D search which is computationally prohibitive. Though techniques exist for computational feasibility using dynamic

programming based approaches [2], these are usually not applicable for real-time applications. Furthermore, it was found upon data analysis that the conditional PDF alluded to here is highly multimodal, and therefore the MLE process, if solved iteratively (as is normally the case in order to avoid the expensive 4-D search) could yield local maxima and thus incorrect solutions for the required unobserved values. As such, due to the computational cost and possibility of local maxima, we do not rely on the statistical approach.

Functional regression methods are an appealing alternative to the statistical methods. A Regression function is a mapping that yields the set of unobserved values when given the observed values as inputs - during the training process, the form and parameters of the regression function are learnt. Typically, the use of regression functions involves selecting an appropriate form of the regression function, specifying its order and number of free parameters, etc. Thus, regression often requires some prior knowledge of the problem being solved as well as some information about the behavior of the data [3]. Several linear and non-linear approaches to regression have become popular in the literature. Two of the most popular methods are support-vector regression (SVR) [14], and artificial neural-network (ANN) based regression [19]. In both cases, when the training data size is large, the number of parameters becomes large and the training process gets highly time-consuming. However, once the regression function parameters are learnt during the training phase, the runtime performance is very fast for both methods since runtime only involves evaluating the function at the given data points. In this thesis, since the training data was very large and SVR-based methods became computationally prohibitive, we relied on ANNs as the prefered regression tool to learn the required mapping.

ANNs provide a principled approach to learning non-linear mappings from a given input space to an output space. The capability of learning non-linear mappings make them especially relevant for our purpose, since we require transformation from a 3-D to a 4-D space - a non-linear operation. A neural network consists of an input layer of certain number of nodes, called its size (which in our case is of size 3 since the input is 3-D), an output layer (which is of size 4 in our case) and a set of intermediate layers which could be of any number and any size, depending on performance. Each layer contains a node, which is an element connected to every other element in the layer preceding it and whose function is to take a weighted sum of all its inputs, add a bias to it, transform it using a non-linear mapping, and feed it to the layer following it. The parameters of the network include this set of weights and biases for each node. The neural network learns the values of these parameters during a training phase that involves the minimization of a cost function. The training algorithm, called the back-propagation algorithm, is an alternate form of a family of dynamic programming algorithms. As in the case of the MLE, the cost function could contain several local extrema and the training process might stop at one local extremum yielding bad sets of weights and biases. However, there are methods to overcome this difficulty, including methods like bootstrapping (c.f. [15]). The limitation with ANNs is that the selection of its architecture (i.e., number of layers, number of nodes in each layer, the non-linearity associated with each node) is is not a straightforward process with clear answers. In this thesis, upon analysis of the data and trial with various different architectures and their associated cost functions, it was found that a neural network with 3 layers yielded reasonable regression results. However, it was found that if the 4 unobserved dimensions were treated as independent variables, and if 4 different individual ANNs were trained, one each to learn the regression for one of the dimensions, the error was much lower than the case of training for all dimensions at once. Therefore, the true power values were predicted from the estimated power values by using an individual neural network for each of these power values. The exact configuration of the neural network for each dimension was: 3 (Input) X 5 (TanSig) X 10 (TanSig) X 6 (Linear) X 1 (Output), where the non-linearity of each layer is given in (.).

The neural network was fed training data from the samples described above, and trained to yield the 4-D unobserved variables given the 3-D observed variables. The overall transformation resulting in the input data passing through 3 layers of non-linear operations can be viewed as the mapping function $\tilde{\varphi}(.)$ that we are looking for, consisting internally of four independent mapping functions $\tilde{\varphi^{(i)}}(.)$. As discussed above, each pitch period and each frequency channel had its own unique mapping function $\tilde{\varphi}(.)$ from this neural network approach.

4.3.1 Practical Issues: Training the Neural Network for Better Regression

In this section, we will briefly cover the procedure that was used to learn the parameters of the neural networks, and some practical issues that had to be dealt with. Figure 4.3 shows the distribution of the observed energy variables $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})$ as evaluated by the procedure described in the previous section. Not surprisingly, the data is roughly distributed in clusters that are dependent on the SNR vaues of the signals they came from. However, the clustering is not very marked, and there is significant overlap between the data points arriving from different SNRs.

From this ditribution, it is intuitive to realize that training a single neural network to learn the mapping function is a less efficient process compared to an alternative: cluster the data into individual cluster using some clustering procedure, and learn a mapping function for each of these clusters. Since the clusters are more representative of the points they contain than the union of all points is, therefore the mappings which we will learn would be more suitable to the points within each cluster than a single mapping that tries to account for all the points in the dataset. Therefore, the training procedure is modified as follows:

1. cluster the data into K different clusters using some clustering scheme, calling the data from the k^{th} cluster as, say, $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})_k$.
2. for each of the clusters k, train an individual neural network or learn the mapping function $\tilde{\varphi(.)}_k$ using the data *only* within that cluster, i.e., the data which is labeled as $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})_k$.

During runtime, for a given test set $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})_{test}$, the procedure to map this point to the appropriate set of observed variables can be done as follows:

1. identify the cluster j into which the data $(E_{\underline{\hat{s}}_p}, E_{\underline{\hat{s}}_a}, E_{\underline{\hat{s}}_{pa}})_{test}$ falls
2. use the mapping function corresponding to that cluster, i.e., $\tilde{\varphi_j}(.)$ to evaluate the set of true (unobserved) parameters

In our system, we use a method of soft clustering by using Gaussian Mixture Models (GMMs) [3]. A Gaussian Mixture Model is first fit to the training data. The order of the GMM is iteratively increased until the relative weight of at least one Gaussian falls below 0.1 (this then is assumed to be the point which marks the beginning of overtraining). The GMM which has all components with relative weights > 0.1 is chosen as the clustering tool. Once the GMM is chosen, all those data points which have a probability of < 0.05 under that distribution are deemed as outliers and eliminated from the training phase. During the training phase, each valid (non-outlier) data point is assigned to a particular Gaussian in the GMM depending on which one was most likely to produce that point (i.e., the Gaussian which maximizes the likelihood of that data point). Once all the data are clustered thus, the mapping functions are then learnt for each of the individual Gaussians. As is clear, this results in the hard clustering of the training data into one of the clusters. This kind of hard clustering could be less efficient and possibly even detrimental to the training process, but a soft clustering solution causes a dramatic increase in computational complexity [3] and was thus avoided. During runtime, though, we try to avoid the hard clustering of the input data and adopt a soft clustering approach. For the data point d_j, we first calculate the probability of it belonging to each of the K clusters, i.e., $p(d_j \in G_k) = P_k p(d_j|G_k)$ where P_k is the relative weight of the k^{th} Gaussian in the GMM. Next, the input data is sent to each of the mapping functions each of which return a mapped value, $\varphi_k\tilde{(}d_j)$. These mapped values $\varphi_k\tilde{(}d_j)$ are all then averaged together with relative weights $p(d_j \in G_k)$ to yield a final mapped value. This kind of soft clustering gives us a smoother, more robust mapped value, which takes into account the influence of other data on the clustering process and on the GMM etc. Finally, it should be noted that in addition to these preprocessing steps including outlier-removal and clustering-before-training, the data used to train each of the individual neural networks was also pre-processed using standard techniques like Z-normalization before being used for training.

As had been mentioned, the dimensionality of the unobserved space was 4. Therefore, there are 4 different mapping functions we want to learn (since we have assumed independence of each dimension from every other). As such, the procedure reported in this section is repeated for each of the dimensions for which we intend to learn the mapping function.

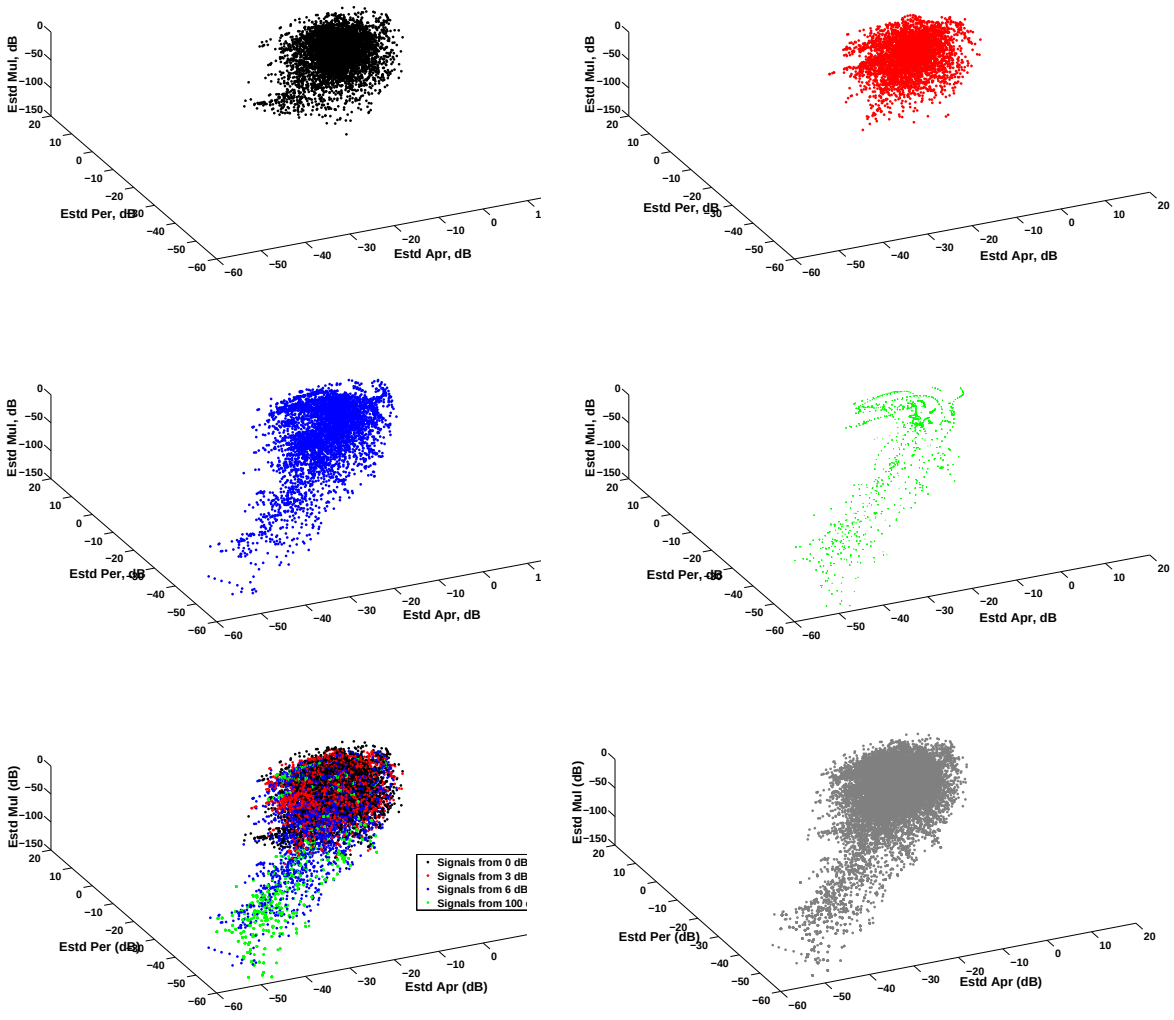

Figure 4.3: The joint distribution of the Estimated Periodic, Aperiodic and Hadamard Powers for different SNRs (first four panels). The fifth panel shows the data from the first four panels in a single plot, and the last panel shows the data as it is seen by the training process for the mapping function

4.3.2 Performance of the ANNs for Regression of Parameters

We have thus far described the method to estimate the True Periodic component $E_{\underline{s}_p}$, True Aperiodic component $E_{\underline{s}_a}$, and True Noise component $E_{\underline{v}_p} + E_{\underline{v}_a}$, from the Estimated Periodic and Estimated Aperiodic components through the use of certain mapping (regression) functions which predict the value of these True components using those of the Estimated components. Since the mapping functions do not capture temporal properties like smoothness, derivative etc., it is natural to expect that the estimates provided by the mapping functions will be noisy in nature and not temporally well structured. We will notice that it is indeed the case, and therefore during runtime, these estimates are temporally smoothed.

Figures 4.4, 4.5 and 4.6 show the predicted values of the true powers ($E_{\underline{s}_p}$, $E_{\underline{s}_a}$, $E_{\underline{v}_p} + E_{\underline{v}_a}$) using the neural networks as trained by the methodology described previously. For the sake of brevity, we look at the total noise power instead of its individual periodic and aperiodic components. It can be seen from these three figures that the estimates yielded by the proposed algorithm are very close to the true ones, at least after smoothing. In particular the true periodic power component, which through our empirical observations appears to be the most critical component for good perceptual quality of reconstruction, is very well predicted by the proposed method.

It may be noted that this set of values corresponds to the speech signal shown in Figure 4.2. Therefore, we can now explore and see how well these *mapped* true periodic, aperiodic and noise energies will reconstruct the clean speech from the noisy one, and how closely the reconstruction matches the one obtained through by using the true values of these parameters. That will give us a measure of the sensitivity of the reconstruction process to the error in estimates of these power values obtained using the mapping functions.

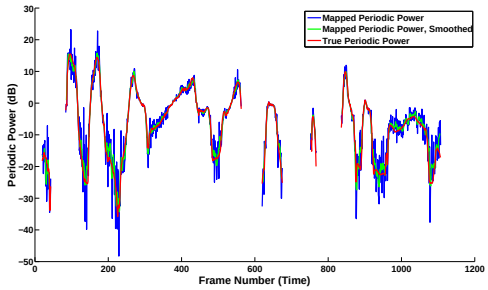

Figure 4.4: Comparison of the true periodic power, and the estimate of the periodic power after mapping using the neural networks, for a speech signal at SNR = 0 dB. The smoothed version of the estimated power is also plotted, and is the actual set of values used for speech enhancement during runtime

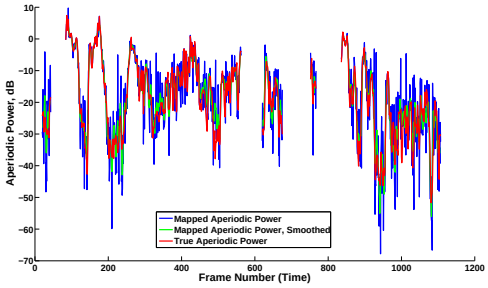

Figure 4.5: Comparison of the true aperiodic power, and the estimate of the aperiodic power after mapping using the neural networks, for a speech signal at SNR = 0 dB. The smoothed version of the estimated power is also plotted, and is the actual set of values used for speech enhancement during runtime

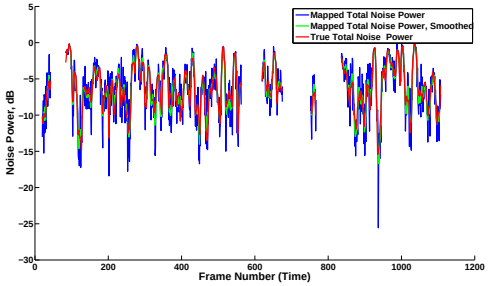

Figure 4.6: Comparison of the true noise power, and the estimate of the noise power after mapping using the neural networks, for a speech signal at SNR = 0 dB. The smoothed version of the estimated power is also plotted, and is the actual set of values used for speech enhancement during runtime

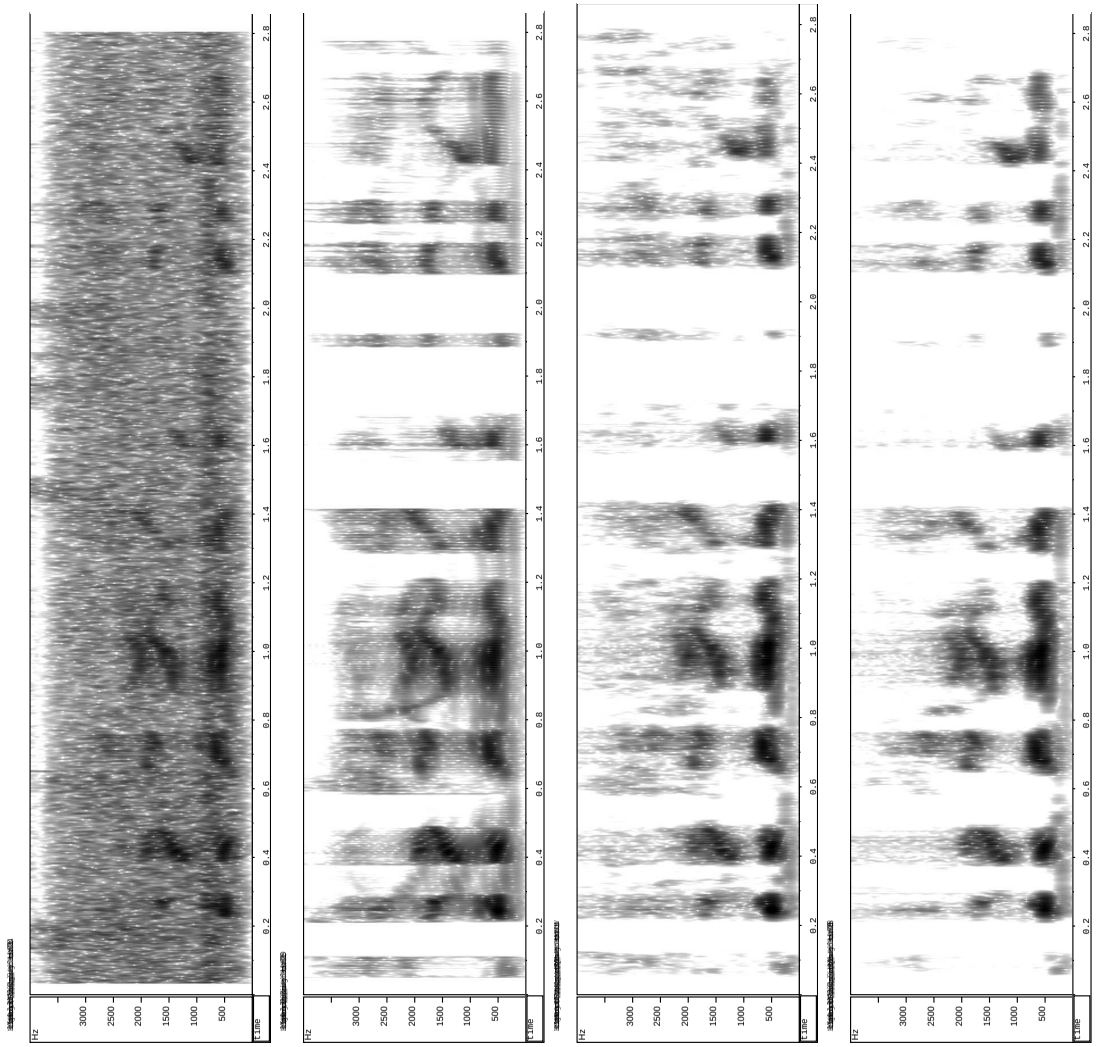

Figure 4.7: Noisy sample speech signal at 3 dB SNR, and its processed versions. (First Panel) Noisy speech signal, (Second Panel) Clean speech signal (only voiced portions), (Third Panel) Enhanced version of the speech signal after processing by the proposed algorithm and after scaling the various components according to the estimates of their true powers as obtained through the neural network mapping, (Fourth Panel) Enhanced version if the individual TFUs were scaled according to match the true periodic power in the speech signal

Figure 4.7 shows the spectrograms of the same speech signal as in Figure 4.2, but with the third panel now replaced by the reconstruction obtained after mapping the periodic estimates. It can be seen clearly that the reconstructed speech signal (third panel) now has a closer spectro-temporal profile to the clean speech signal (second panel) than did the unscaled estimate from Figure 4.2. Furthermore, the reconstruction using the true energies (last panel) is also well represented by the speech obtained by reconstruction using the mapped values (third panel). The reconstruction also shows spectro-temporal smoothness leading to a more perceptually acceptable speech estimate, as was expected from the mapping process. Finally, it should be pointed out that from the Figures 4.4, 4.5 and 4.6, it was expected that the reconstruction may be significantly affected due to the noisy nature of the predicted parameters, but we now see clearly that the smoothing operation has ensured that the departure from the ideal estimates is not much, and is perceptually (as well as visually!) acceptable.

We will finally end this section by looking at the distribution of the data we want to predict (i.e., the true energy components) and distribution of the data as predicted by our neural-network based regression technique. We will also look at the distribution of the relative error between the two. These are shown in Figure 4.8. It can be seen that the predicted data (red plots) is statistically distributed in the same region as the original data to be predicted (blue plots), and that the PDF of the estimated data more or less matches that of the true data, which suggests that the neural networks are doing a decent job at predicting the unobserved data. The distribution of the relative error between the true and predicted values, i.e., ratio of the difference between true and predicted, to the true value, is shown in black lines. We can see that the variance of prediction is slightly < 1 for all three plots, which actually means that the predicted data is very rarely away from the true data by more than a factor of 2. While this is still unacceptable, especially in the dB scale, for most applications, we have already seen that after smoothing of the predicted values, the enhancement offered by the proposed algorithm is perceptually good and seemingly adequate.

4.3.3 Add-Back of the Unvoiced Regions from the Noisy Speech Mixture

Recovery of unvoiced regions from noisy speech signals is one of most difficult, and currently yet unsolved, problems. The hurdle lies in the difficulty of characterizing unvoiced sounds in a systematic and consistent fashion, given the large variety in the types of unvoiced sounds as well as the variations in their pronunciation in conversational speech. Coupled with the fact that since they are typically weak sounds, they are also highly susceptible to the influence of noise and even more from competing voiced speech - extraction of unvoiced speech from noisy mixtures is a tough nut to crack. In our earlier work, we tried to exploit the perceptual properties of the human auditory system by selectively emphasizing and de-emphasizing certain regions of the speech mixture, so as to recover reasonable and acceptable estimates of the unvoiced phonemes. We found from our subjective listening tests that the resulting signals were perceptually more prefered than the unprocessed ones, and improved speech intelligibility significantly (please refer to Section 5.1.5 for details). We will now briefly discuss the approach taken to extract these perceptually significant regions and add them back to the voiced estimates we had extracted using the proposed algorithm.

Consider a speech mixture $x[n]$ consisting of speech from speakers A and B, calling them $s_A[n]$ and $s_B[n]$ respectively. Let the contributions of these speech signals to the mixture be α and β respectively. Given our proposed least-squares segregation model that can recover voiced regions from the mixture, we thus obtain estimates $\hat{s_A}[n]$ and $\hat{s_B}[n]$, as well as $\hat{\alpha}$ and $\hat{\beta}$. These estimates can then be used to obtain what we call the Conjugate Residuals of the signals from A and B. In particular, defining the conjugate residual of speaker A as $\tilde{s_A}[n] = (x[n] - \hat{\beta})/\hat{\alpha}$ (i.e., the residual obtained from estimate of speaker B) and defining the conjugate residual of B similarly in terms of estimate of speaker A, we notice that (1) the estimate $\hat{s_A}[n]$ (ideally) contains the voiced portion of the speaker A due to the pitch-based segregation, and (2) the estimate $\tilde{s_A}[n]$ (ideally) contains the voiced *as well as* unvoiced portions of speaker A along with the unvoiced portion of speaker

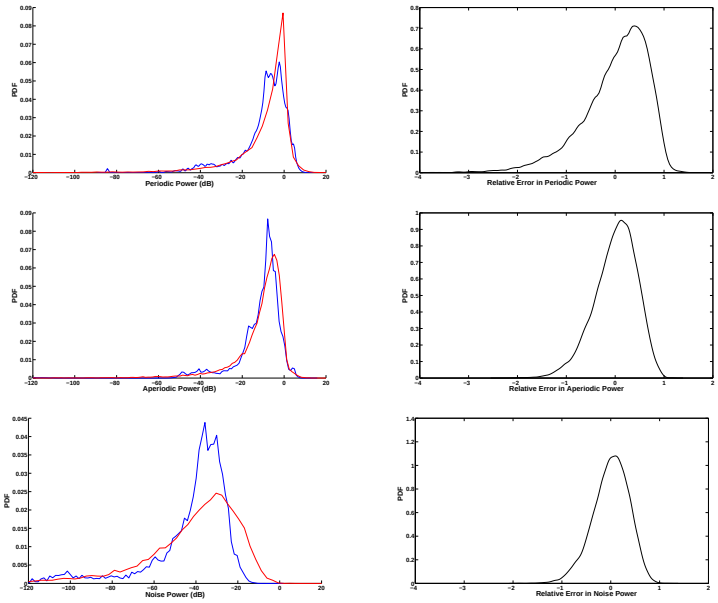

Figure 4.8: (Left Column) Distribution of the True Periodic, Aperiodic and Noise Powers (blue plots) and of their estimates as provided by the mapping functions from the neural networks (red plots), (Right Column) Distribution of the relative error between the true and estimated values, (Top Row) Plots for the periodic power (Middle Row) Plots for the aperiodic power, (Bottom Row) Plots for the noise power

B. Thus, we ask ourselves if there is an intelligent method to combine these two estimates, so that the unvoiced portions of the target speaker might be recovered from the conjugate residual of the target, if not the direct estimate.

In order to combine the two estimates, we first note that in practical situations, the residual is seldom free of the harmonics of both speakers, and will always contain a residual voiced component even if the segregation model has recovered a majority of the voiced portion. As such, the conjugate residual $\tilde{s_A}[n]$ would often also contain some of the voiced speech signal from B. As such, care must first be taken to remove the effects of this voicing leakage, since any such leakage can adversely affect the perceptual clarity of the final target stream output. The conjugate residual signal, as well as the estimate of the signal, were first split into high-frequency and low-frequency regions by a simple set of filters. For the high frequency regions, the following set of signals were added together: estimate from the high-frequency region, conjugate residual extracted from the regions where both speakers were detected to be unvoiced, multiplied by a constant c_{UUH} and conjugate residual extracted from the regions where target speaker alone was unvoiced, multiplied by a constant c_{UVH}. In the low frequency regions, the following sets of signals were added together: estimate from the low-frequency region, and conjugate residual extracted from the regions where both speakers were detected to be unvoiced, multiplied by a constant c_{UUL}. The parameters c_{UUH}, c_{UVH} and c_{UUL} were chosen empirically and to obtain optimal perceptual results, they had to be varied for different TMRs. The effect of this total manipulation was to reinforce the estimates $\hat{s_A}[n]$, and at the same time provide additional consonant information in the form of the conjugate residual $\tilde{s_A}[n]$.

This method of recovering unvoiced regions has proved to be vey effective, as seen from the results of perceptual tests: both normal hearing listeners and hearing aid users have already demonstrated to us that this approach of extracting unvoiced speech can affect a significant improvement in speech intelligiility in competing talker conditions. To the best of our knowledge, this is the first such approach to the recovery of the unvoiced regions, and from our experience over several different databases and environments has proven to be a very useful method of obtaining

the consonant information.

The choice of the above-mentioned parameters was arbitrary in the case of the perceptual tests, and was manually set by the user to take on a set of values that yields the best perceptual experience. In the next section, we will explore the problem further and in particular attempt to automate the method of estimating the required coefficients which determine how to "add back" the mixture signal. We will continue with the case of speech enhancement, where there is only one speaker whose unvoiced speech needs to be extracted, and then generalize for the 2-speaker case.

4.3.4 Automatically Estimating the Required Amount of Add-Back: Recovery of the Unvoiced Regions Using the Parameters Learned from the Voiced Regions

In this section, we will explore how to intelligently add back the mixture (noisy) speech signal to the extracted voiced portions so as to recover the unvoiced speech components which were not estimated by the proposed model. Intuition from Section 4.3.3 indicates that since the unvoiced information is expected to exist close to the voiced-unvoiced boundaries in speech, it is sufficient to add back only those regions which are within a temporal margin of such boundaries. The only question remaining then is the decision of how much of the mixture signal to add back. We rely on the information provided by the ANN-based regression system to estimate this energy. We first note that the aperiodic content in the speech signal is known to us due to the ANN-based regression, but is available only in the voiced regions. The aperiodic energy of the unvoiced regions is unavailable. However, since speech is usually continuous in nature and these energies usually do not exhibit rapid discontinuities (exceptions are strong plosives), the aperiodic energy estimates from the neighboring voiced regions can be used to predict the aperiodic energies in the current unvoiced region.

In order to impose practical utility of the algorithm, we search for a causal version of this method which only uses the energy obtained from the previous frames to predict the aperiodic energy in the current frame. Thus, the aperiodic energy estimates (equivalently, the add-back amounts) are known only for unvoiced regions succeeding voiced regions, but not for those preceding voiced regions. There are several alternatives to using the aperiodic energies from the previous frames, and the method we use here is the exponential decay of the aperiodic energy estimates from the last known voiced frame. Figure 4.9 illustrates the true and estimated aperiodic energy for the signal within a particular channel of a noisy speech input. Figure 4.10 illustrates the modified aperiodic energy, which is obtained by predicting the information in the unvoiced regions from the past known voiced regions. As can be seen, whenever an unvoiced region is encountered the red line decreases exponentially from its last known point and continues to decrease until the next voiced region, where the aperiodic energy is again computed from the voiced frame. This track of aperiodic energy is used to determine how much of the noisy signal to add back: the signal is multiplied with a scaling factor in such a way that the energy of the add-back signal follows the track shown by the red plot in Figure 4.10.

The motivation for choosing this particular track is as follows: as has been argued, since speech seldom shows rapid energy discontinuities the aperiodic energy is expected to show a continuous variation from the immediately past voiced region. Also, since the prediction must be causal (i.e., cannot wait until the beginning of the next voiced region to predict the energy of the current unvoiced region) the algorithm can only look backwards and not forwards - therefore, one cannot use methods like interpolation to fill in these unknown values. The perdicted aperiodic energy itself must not increase with time, since that would increase the probability of adding back noise and not speech - therefore the predicted aperiodic energy curve should be non-increasing. We choose the exponential decay since that allows for all these requirements and in addition, is computationally efficient to implement.

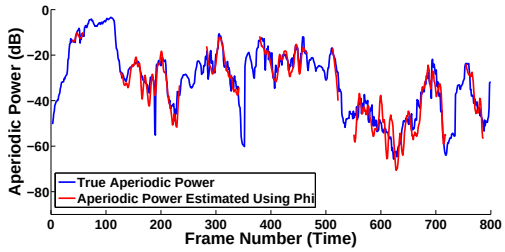

Figure 4.9: Comparison of the true aperiodic power, and the estimate of the aperiodic power after mapping using the neural networks, for the signal within a channel in a speech signal at SNR = 0 dB. The smoothed version of the estimated power alone is plotted, and is the actual set of values used for speech enhancement during runtime.

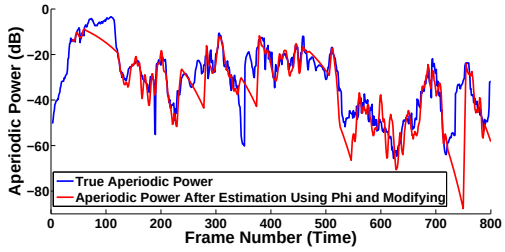

Figure 4.10: Causal prediction of the aperiodic power in unvoiced regions, using the (estimated) aperiodic energy information from the voiced regions. Exponentially decaying values from the last known voiced frame are used here as the prediction curve, but one can in general use any non-increasing curve for good perceptual effects.

Once the aperiodic energy plot is predicted (either from the ANNs for the voiced regions, or through exponential decay for the unvoiced regions), this is used to determine how much of the noisy speech signal to add back to the estimated voiced portions in order to recover the unvoiced portions of speech. It may be noted that since there are multiple channels with the same operation but data-dependent, the algorithm is automatically able to recover unvoiced regions effectively.

All results reported in this thesis are based on this automatic method of recovering unvoiced regions, unless otherwise noted.

4.4 The Case of Speech Segregation

Thus far, we have discussed in the context of speech enhancement the technique for extracting the periodic components of the speech signal, using the powers of these estimated components with their aperiodic counterparts to predict the true values these components should be taking, and then scaling the estimates accordingly so to match those predicted values. In both cases, the estimation of the periodic components is done in a similar fashion as a least-squares solution for a set of equations. However, the process of scaling these estimates in terms of their reliability as well as estimating the aperiodic and noise components is a much different task. For the case of speech segregation, the proposed solution is much more complicated, because of the larger number of variables involved as well as the non-linearity of the segregation operation. In particular, for the case of speech enhancement, the estimates were obtained as linear functions of the mixture signal. However, for the case of speech segregation, the estimates were obtained by picking specific elements off the estimand vector γ - a non-linear operation - and then use these selected coefficients for estimation of the individual speaker. The observed variables in the case of speech segregation

would be the model estimates of both speakers (in parallel to the estimate of the periodic component of the target speaker in enhancement) as well as the residue of the model (in parallel to the estimate of the aperiodic component of the target speaker). The variables which we need to estimate for appropriate scaling would be the true periodic and aperiodic powers of both speakers, as well as the noise power.

Finally, the trickiest problem would be to find the actual true values of these estimands. In the case of speech enhancement, this was a simple task since the training data was obtained from the clean and noisy speech signals using the same model (i.e., $\underline{s_p}$ is estimated from $\underline{s}$ in exacly the same fashion as $\underline{\hat{s_p}}$ is obtained as $\underline{x}$). However, in the case of speech segregation, these operations are not parallel to each other since estimation of each speaker is obtained *from the mixture* as a non-linear operation while that of the speaker *from its clean version* is a linear operation. Therefore, this pair of data is not exactly compatible with each other. However, all the same at this point, we do use this set of data for training the mapping function.

In particular, if we define the following variables:

- $\underline{x}$: the mixture signal which is being analyzed by the algorithm
- $\underline{s_1}$: the speech signal corresponding to the speaker 1, which we want to estimate
- $\underline{s}_{1_p}$: the periodic component of the speech signal as obtained from its clean speech version
- $\underline{s}_{1_a}$: the aperiodic component of the speech signal as obtained from its clean speech version
- $\underline{s_2}$: the speech signal corresponding to the speaker 2, which we want to estimate
- $\underline{s}_{2_p}$: the periodic component of the speech signal as obtained from its clean speech version
- $\underline{s}_{2_a}$: the aperiodic component of the speech signal as obtained from its clean speech version
- $\underline{s_v}$: the noise component added to the clean speech version to obtain the mixture $\underline{x}$
- $\underline{\hat{s_1}}$: the estimated periodic signal which corresponds to the speaker 1, as yielded by the segregation algorithm
- $\underline{\hat{s_2}}$: the estimated periodic signal which corresponds to the speaker 2, as yielded by the segregation algorithm
- $\underline{\hat{s_r}}$: the residual signal as yielded by the segregation algorithm, which is equal to $\underline{x} - (\underline{\hat{s_1}} + \underline{\hat{s_2}})$

then we will need to learn a mapping $\varphi : \{E_{\underline{\hat{s_1}}}, E_{\underline{\hat{s_2}}}, E_{\underline{\hat{s_r}}}, E_{\underline{s_1 \hat{,} s_r}}, E_{\underline{s_2 \hat{,} s_r}}, E_{\underline{s_1 \hat{,} s_2}}\} \rightarrow \{E_{\underline{s}_{1_p}}, E_{\underline{s}_{1_p}}, E_{\underline{s}_{2_p}}, E_{\underline{s}_{2_a}}, E_{\underline{s}_v}\}$ in order to obtain the scaling functions as developed in the case of speech enhancement.

We again refer to the neural network to help us learn such a mapping. In an exactly analogous fashion to the case of speech enhancement the mapping function is learnt for the case of speech segregation, and used to scale the estimates appropriately. Similar to the estimation of the unvoiced regions using the previous voiced estimates in the case of Speech Enhancement, here too the unvoiced speech estimates are made using the aperiodic energy estimates in the voiced regions. From our experiments on various tasks evaluating the speech segregation algorithm, this method was found to yield perceptually good segregation results that were also well suitable for various objective tests - the performance is discussed in Chapter 5.

4.5 Speaker Assignment

The basic block diagram of the final speech segregation system as visualized in this thesis is outlined in Fig. 4.11. The last stage of the algorithm is the Sequential Grouping stage, which decides which estimated speech signal goes to which source. It must be noted that until this point,

the algorithm has been discussed in terms of how it separates the speech signals or speech from noise, but no allusion has been made to the process of how the separated signals are assigned to the appropriate source. In particular, over multiple frames, the algorithm yields two signal estimates corresponding to the two speakers in the mixture. However, these estiamtes need to be linked over time so that it is know they come from the same source. This problem of linking or assigning speech to the appropriate speaker is discussed in brief here.

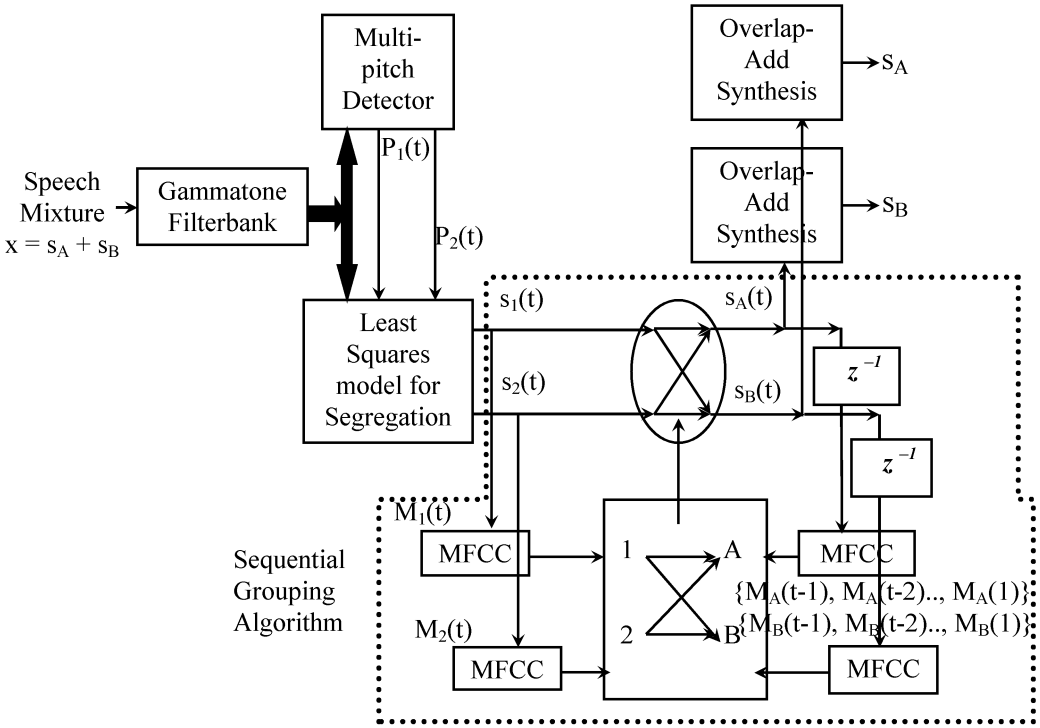

Figure 4.11: Block diagram of the proposed speech segregation system

We follow [53] and rely on using the spectral envelope in the form of Mel-Frequency Cepstral Coefficients (MFCCs) for performing sequential grouping. The intra-segmental grouping is achieved by comparing the distances between consecutive MFCC vectors, while the inter-segmental grouping is achieved by creating online distribution models of the MFCC vectors and using them to make grouping decisions. In addition, we improve our acoustic cue segregation by improving the separation of the different kinds of unvoiced sounds.

4.5.1 Intra-Segment Stream Formation

Even though the multi-pitch detector yields the numerical pitch estimates of the two speakers, and the segregation algorithm yields the two constituent speech signals for a given frame, these are not yet assigned to any speaker. In particular, for two speakers A and B, and two segregated signals $s_1(t)$ and $s_2(t)$, the question of which segregated signal $s_i(t)$ should be assigned to A will be answered in this section and the next. In the proposed algorithm, the well-known MFCCs are used as the features for the speaker assignment problem. The MFCCs of each of the segregated streams $s_1(t)$ and $s_2(t)$ are evaluated to get the features $M_1(t)$ and $M_2(t)$ respectively. These features are then used in combination with the features from the speech of the two speakers A

and B in the previous frames, i.e., with [$M_A(t-N)$, $M_A(t-(N-1))$, ..., $M_A(t-2)$, $M_A(t-1)$] and [$M_B(t-N)$, $M_B(t-(N-1))$, ..., $M_B(t-2)$, $M_B(t-1)$]. There are different approaches adopted for using this information in two different scenarios. The order of the MFCCs used in the proposed system is 13, including the energy coefficient but not including the difference coefficients.

In regions of voiced speech, (say A being voiced) due to the nature of the continuity of acoustic features over time, consecutive MFCC features (say $M_A(t-1)$ and $M_A(t)$) are expected to be very close to each other (where the notion of closeness is defined by some distance measure). As such, during the segregation process, at time frame t, the MFCCs of the segregated streams, namely $M_1(t)$ and $M_2(t)$, are compared with the MFCCs of the known streams one time step in the past, i.e., $M_A(t-1)$ and $M_B(t-1)$. Defining $D_{1A} = ||M_1(t) - M_A(t-1)||$, $D_{2B} = ||M_2(t) - M_B(t-1)||$, $D_{1B} = ||M_1(t) - M_B(t-1)||$ and $D_{2A} = ||M_2(t) - M_A(t-1)||$, if $(D_{2A} + D_{1B}) - (D_{1A} + D_{2B}) > \gamma$ (a threshold), then the assignment of the speech streams is $1 \rightarrow A$ and $2 \rightarrow B$, i.e., the speech signal obtained from stream 1 is assigned to speaker A and the one obtained from stream 2 is assigned to speaker B. On the other hand, if $(D_{1A} + D_{2B}) - (D_{2A} + D_{1B}) > \gamma$, then the assignment of the speech streams is $2 \rightarrow A$ and $1 \rightarrow B$. If the difference $|(D_{1A} + D_{2B}) - (D_{2A} + D_{1B})| < \gamma$, then this implies that both the distance measures are approximately in the same range, and therefore a decision based on this measure is unreliable; such cases will be treated in the next section. While this approach was initially devised for the case of both speakers showing voiced speech, it was also found to work well even when one or both of the speakers showed unvoiced speech, as long as the distance measure exceeded the minimum threshold γ. Regions of speech wherein the MFCC distances are sufficient to enable good grouping can be called as a single segment since the acoustic features (MFCCs) show a smooth continuity in this region, and grouping within such segments may be considered as a form of Intra-Segmental grouping (which we achieve using MFCC distance measure in our case).

4.5.2 Inter-Segment Stream Formation

However, there do occur some regions in the mixture speech signal, wherein the MFCC distances referred to above are not reliable enough to make a decision of speaker assignment. In such circumstances the decision of speaker assignment using a simple MFCC-distance measure may not be reliable, and such regions divide the mixture speech signal into multiple segments, with these regions acting as boundaries between them. A different strategy needs to be adopted to group segments that are disconnected from each other by such low-reliability frames. An obvious answer to the question of grouping in such cases would be to build statistical models of the features already identified for each specific speaker, and then use those models to make assignment decisions at a later stage. Thus, during the assignment of the speech to the speaker A or B using the MFCC distance measure, the reliable regions are used to build online models of the distributions of the MFCCs under each speaker (in contrast to prior models wherein the models are created prior to the segregation process itself). In particular, for all features [$M_A(t-N)$, $M_A(t-(N-1))$, ..., $M_A(t-2)$, $M_A(t-1)$] which are found reliable enough to make the intra-segmental decision, a model of the distribution of these MFCCs under the speaker A is built. In the proposed system, a Gaussian Mixture Model (GMM) is used to model the MFCC data, with the parameters of the GMM being learnt by the Expectation Maximization algorithm. During the speaker assignment decision, the speaker models are used to assign the MFCC vector to the appropriate speaker: If $p(M_1(t)|A) > p(M_1(t)|B)$ and $p(M_2(t)|B) > p(M_2(t)|A)$, then $1 \rightarrow A$ and $2 \rightarrow B$. If $p(M_1(t)|B) > p(M_1(t)|A)$ and $p(M_2(t)|A) > p(M_2(t)|B)$, then $2 \rightarrow A$ and $1 \rightarrow B$. If neither of these conditions is met, a conflict arises with the single frame at time t, and the decision is taken by voting over multiple frames.

This approach appears to have the disadvantage that since models are built online, they may be weak and unreliable in the beginning of the input speech file and become more reliable as more data arrives. However, due to the nature of speech, it so happens that situations wherein both speakers are unreliable (e.g., both speakers are simultaneously silent) occur very rarely and

even if they do, it is quite likely that enough MFCC data was extracted by that time to create reliable models. After assigning the segregated speech streams to the appropriate speakers, the overlap-add (OLA) method is used to reconstruct the total speech signals, as described in [47].

The performance of both the proposed intra- and inter- segmental grouping procedures is discussed in the thesis work of Mahadevan [31]. Currently, there exist no reliable methods of performing speaker assignment based on very little prior training data, as is often the practical case in many real-world environments. As such, there needs to be further work done in order to find an appropriate procedure to assign the extracted speech streams. This will be part of the future work in this thesis.

4.6 Chapter Summary

In this chapter, we have looked at several important aspects of the speech extraction problem. We have looked at improving the reliability of the estimates obtained by the segregation model, by scaling the various estimates with appropriate scaling factors. We found that this scaling strategy greatly improves the quality of the segregated speech signal. We have also looked at the estimation of aperiodic energy in both the voiced and unvoiced regions, and have especially looked at the context of extracting aperiodic energy from noisy speech. We discussed these issues in both the contexts of speech enhancement and speech segregation. Finally, we briefly discussed the problem of assigning the extracted speech to the appropriate source speaker.

Chapter 5

EVALUATION OF THE PROPOSED ALGORITHM ON SPEECH MIXTURES AND NOISY SPEECH

5.1 Introduction

In this chapter, we will evaluate the proposed speech segregation algorithm on a number of different tasks using several different criteria. It is of interest to note that while the speech separation and enhancement literature is vast and research has been ongoing for over 20 years, there is still no agreed set of standard criteria used to compare the performance of these segregation or enhancement methods with each other. Even today, the most accurate method of evaluating speech quality is through subjective listening tests. While much effort has been placed on developing objective measures to predict speech quality with high correlation to subjective results, unfortunately no objective measure currently exists that correlates as high as desired with the perceptual quality scores reported by human listeners. Some of the objective measures that have been used by various researchers include the mean square error, spectral distance measures, LPC measures, Itakura-Saito distance, log-likelihood ratio, segmental SNR etc [24]. Currently, the most popular measure used for the objective evaluation of speech signals is the Perceptual Evaluation of Speech Quality (PESQ) score, which has been shown to have a correlation of over 0.8 with human perceptual scoring [24]. It is also fair to compare the performance of various algorithms on applications such as ASR or SID, where the utility of the algorithm can be evaluaed for that particular application. As such, performance on ASR can also be considered as an objective measure for evaluation. In this thesis, we will report the performance of the algorithm on both the enhancement and segregation tasks, using some of these objective measures. We will also compare the proposed algorithm to other algorithms in the literature for some of these objective measures. Furthermore, we will report the performance of the algorithm on the segregation task in the case of perceptual tests, for both normal hearing listeners and hearing aid users.

The first set of experiments attempts to identify the percentage of speech energy retained by the proposed segregation algorithm, in comparison with a popular contemporary speech segregation algorithm [23], for the speech segregation task, and some popular speech enhancement algorithms for the enhancement task. The second set of experiments describes the improvement in the quality of the target signal from the mixture in terms of the Signal-to-Noise Ratio (SNR) for both the speech segregation and enhancement tasks. The third set of experiments describes the performance of the algorithm using an objective measure called the PESQ score. Next, we evaluate our algorithm on an Automatic Speech Recognition (ASR) task involving both the challenges of speech segregation as well as speech enhancement. The last set of experiments evaluates the quality of segregation using a perceptual test. In all the different modes of evaluating the algorithm, we find that the proposed algorithm yields speech signals which are better or more suitable than the original speech mixtures they were extracted from, for the task in question. Furthermore, we will see that the proposed speech extraction algorithm (which is a generic approach to both segregation and enhancement) often compares well with or outperforms the state-of-the-art algorithms on the tasks of evaluation.

We will first describe the databases used in the different experiments.

5.1.1 Databases

The first database which the performance is compared on is the Cooke database [5], which contains 10 voiced utterances in the presence of 3 different masker speech signals and 7 different noise signals. The task is to recover the speech of the voiced speaker, which was called the Target. The target speaker to be recovered is always a male speaker. The following are the different noise types: 1-kHz pure tone (N0), white noise (N1), noise bursts (N2), babble noise (N3), rock music (N4), siren (N5), telephone trill (N6), female masker speech (N7), male masker speech (N8) and female masker speech (N9). The masker speech signals (N7-N9) are all composed of both voiced and unvoiced speech. This database was chosen in order to compare the performance with other algorithms which reported segregation results on this database [23, 52]. It may be noted that this database involves both speech the enhancement (N0-N6) and speech segregation (N7-N9) tasks. This database will be refered to as the "Cooke database" in the rest of this thesis.

The second database consists of synthetic mixtures created by adding together speech signals from the TIMIT database. Three classes of mixtures were created: different gender (FM), same gender (male, MM) and same gender (female, FF). For each class, 200 pairs of sentences with lengths closest to each other were identified. In the case of the FM database, half the dataset had the male as the target speaker and half the dataset had the female as the target speaker. Care was taken during this process that no speaker or no utterance was the same in any pair. Each pair of signals was relatively normalized so that the ratio of their energies was 0 dB, i.e., both signals were equally strong. These were then added together in different target to masker ratios (TMRs), ranging from -9 dB to 9 dB in steps of 3 dB. This procedure gave a total of 600 mixture signals (200 for each class) for each of the seven TMRs. This database was used for the test of percentage of speech energy retained, and will be refered to as the "TIMIT mixture database" in the rest of this thesis.

The third database consists of synthetic noisy speech created by adding speech signals from the TIMIT database to various noise samples. 600 speech files were used. 17 different typs of noises were used in the creation of this database: (1) white noise, (2) pink noise, (3) babble noise, (4) restaurant noise, (5) subway noise, (6) car noise, (7) wind noise, (8) street noise, (9) airport noise, (10) siren noise, (11) engine noise from a motorboat, (12) helicopter noise, (13) tank noise, (14) rock music, (15) piano music, (16) noise from a chainsaw and (17) vibration noise. In a similar procedure as above for the TIMIT mixture database, each of the noise files was added to each of the speech files in different signal to noise ratios (SNRs), ranging from -9 dB to 9 dB in steps of 3 dB. We will refer to this database as the "TIMIT noisy database". It must also be noted that this was the database which was used to learn the mapping functions of the neural network refered to in Chapter 4 for the speech enhancement task, and the TIMIT mixture database was used to learn the mappings for the speech segregation task. This database is different from the Cooke database in the sense that the former had only voiced speech in the target signal, while this database has unvoiced speech as well in the target.

The Speech Separation Challenge (SSC) database [7] is a popular database used for testing ASR performance of speech segregation and speech enhancement algorithms. The database consists of a target speaker in the presence of a masker speaker or background noise. The target speaker's utterance consists of a color, a letter from the English alphabet, and a number from 0 to 9. In the case of interfering talker, which is the problem this set of experiments will focus on, both the speakers are speaking an utterance consisting of the same sequence of words, i.e., adhering to the same syntax as above - in this case, the target is the speaker who utters the color "white". The task for the ASR system is to recognize the letter and the number the target speaker has uttered. The standard ASR system provided for the task is the HTK (HMM ToolKit) system using the standard set of MFCCs. In order to test the applicability of the system in different scenarios, the target and masker signals are added in different Target-to-Masker Ratios (TMRs), ranging from

-9 dB to 6 dB in steps of 3 dB. The SSC database was used for evaluating the performance of the segregated speech signals on the ASR task since several different segregation and enhancement approaches have participated in the challenge, and thus it gives a good benchmark for evaluation of the algorithm.

The fifth database consists of synthetic mixtures created using the IEEE database [35]. Three classes of mixtures were created in this case too, with gender combinations as described above and for TMRs ranging from -6 dB to 6 dB in steps of 3 dB. In addition, noisy speech samples were created by adding different kinds of noises (car, subway and babble noise) to the clean speech signals under different SNRs, ranging from -6 dB to 6 dB in steps of 3 dB. Each condition had 60 utterances. The database was used for evaluating the objective and subjective scores of the signals processed by the proposed speech extraction algorithm, and for evaluating the SNR of the noisy signals. This database will be refered to as the "IEEE database" in the rest of this thesis.

5.1.2 Experiment 1: Percentage of Speech Retained

Since many algorithms in the literature report their evaluations on the percentage of speech retained after segregation, we will use that criterion as a benchmark for comparison. The criteria used are the estimated time-frequency masks of speech as introduced in Chapter 1. We recollect that the Ideal Time-Frequency Mask (Dominated), called ITF_{DOM} is defined as the sets of all TFUs where the target speech dominates over the interference (speech or non-speech), and was introduced as the reference mask to which the performance of various algorithms are compared. The test is how closely the non-silent regions of the reconstructed signals match to the ITF_{DOM}. By motivating that the ITF_{DOM} was not sufficient to describe the perceptual quality of the reconstructed speech signals, we introduced a new mask, call the Complete Ideal Time Frequency Mask, ITF_{COM} - this was the set of all TFUs where the target speech signal is non-silent.

To evaluate the overall performance of the algorithm in separating speakers, we will compare the estimate of the ITF_{DOM} (and that of the ITF_{COM}) of the speech mixture as obtained by the proposed segregation algorithm. The Estimated Complete TF mask (ETF_{COM}) of a speaker is obtained by finding all the non-silent regions of the *reconstructed* speech of that speaker. Experiments with the proposed algorithm show that most of the speech-present region of the target speaker are well captured by the algorithm, and very little region is missed. Furthermore the Estimated Dominated TF mask (ETF_{DOM}), which is obtained by finding all regions in the reconstructed stream A which are stronger than corresponding regions in B, also well matches the Ideal Dominated TF mask (ITF_{DOM}). The performance of the algorithm is equally good for both the target and masker speakers, with few regions missed or falsely detected. The most interesting observation is that since we are trying to estimate both the constituent signals simultaneously, the regions where one speaker dominates over the other are also well-preserved in our estimates. Thus, the match between the ITF_{DOM} and the ETF_{DOM} is very close, indicating that in terms of performance metrics adopted by other approaches, our proposed algorithm would be expected to perform comparable to state-of-the-art algorithms. We will demonstrate that this is indeed the case in the following sections.

In order to first establish the segregation capacity of the algorithm, we compare the performance of the proposed algorithm to that of the algorithm presented in [23] on the same task. The database used for comparison is the Cooke database described above. The metric of comparison is the Percentage of Energy Loss (P_{ELD}) and the Percentage of Noise Residue (P_{NRD}), defined as follows with respect to the ITF_{DOM} of target:

$$P_{ELD} = \frac{\text{Total Energy of TFUs with } ITF_{DOM} = 1 \text{ and } ETF_{DOM} = 0}{\text{Total Energy of TFUs with } ITF_{DOM} = 1} \tag{5.1}$$

$$P_{NRD} = \frac{\text{Total Energy of TFUs with } ITF_{DOM} = 0 \text{ and } ETF_{DOM} = 1}{\text{Total Energy of TFUs with } ITF_{DOM} = 1} \tag{5.2}$$

Masker Type	P_{ELD} [23]	P_{NRD} [23]	P_{ELD} [52]	P_{NRD} [52]	P_{ELD} Proposed	P_{NRD} Proposed	SNR Proposed
n7	3.02	1.83	8.61	4.23	1.70	1.14	18.01
n8	1.71	1.34	7.27	0.48	2.44	0.16	20.20
n9	10.11	17.27	15.81	33.03	2.58	6.43	13.16

Table 5.1: Comparison of the segregation performance of the proposed algorithm with that of [23] in terms of the critera defined in [23] as well as the SNR of reconstruction

i.e., P_{ELD} is the relative energy present in the ITF_{DOM} but missing from the ETF_{DOM} and P_{NRD} is the relative energy absent in ITF_{DOM} but detected as present in ETF_{DOM}. Note that these parameters represent the "missed speech" and the "leakage error" concepts that were alluded to in the beginning of this chapter. Ideally, both these numbers must be as low as possible.

However, as argued earlier, since the ITF_{DOM} does not provide complete information about the goodness of reconstruction, and it is the ITF_{COM} that we should try to estimate correctly, not just the ITF_{DOM}. Therefore, we also present the values P_{ELC} and P_{NRC} which are obtained by considering the ITF_{COM} mask instead of the ITF_{DOM} and are defined as follows:

$$P_{ELC} = \frac{\text{Total Energy of TFUs with } ITF_{COM} = 1 \text{ and } ETF_{COM} = 0}{\text{Total Energy of TFUs with } ITF_{COM} = 1} \tag{5.3}$$

$$P_{NRC} = \frac{\text{Total Energy of TFUs with } ITF_{COM} = 0 \text{ and } ETF_{COM} = 1}{\text{Total Energy of TFUs with } ITF_{COM} = 1} \tag{5.4}$$

Table 5.1 demonstrates the performance of the segregation algorithm on the Cooke database using these proposed measures, along with comparable numbers for the algorithm in [23].

It can be seen that the values of P_{ELD} and P_{NRD} are in general lower for the proposed algorithm than for comparable state-of-the-art segregation algorithms. Furthermore, the P_{ELC} & P_{NRC} values were found to be surprisingly low (as will be evidenced by Fig. 5.2), indicating that most of the speech-present regions are well preserved in the reconstruction, and the missed regions or falsely identified regions are very low energy TFUs. However, it is possible that certain TFUs in the reconstructed signals may be non-silent (and thus give low P_{ELC}) but might carry the signal content of the other speaker. Thus, in order to confirm that the reconstructed signals are indeed of good quality, we also look at the SNRs of the reconstructed signals and other perceptual measures (both objective and subjective), and it becomes clear that the quality of reconstruction by the proposed algorithm is good.

Figure 5.1 shows the performance of the algorithm in comparison with other algorithms which reported results on the same database. It must be noted that since this database contains both competing speech signals as well as noise interference, this evaluation in a sense compares the overall speech extraction performance of different algorithms. Although the SNR values after enhancement are quite high for most algorithms, it must be remembered that the speech data in this database consists completely of voiced target speech, which makes the task relatively easy. In particular, the proposed algorithm outperforms all the other algorithms in different noisy conditions. Interestingly, the algorithm also outperforms the Ideal Binary Mask in situations where

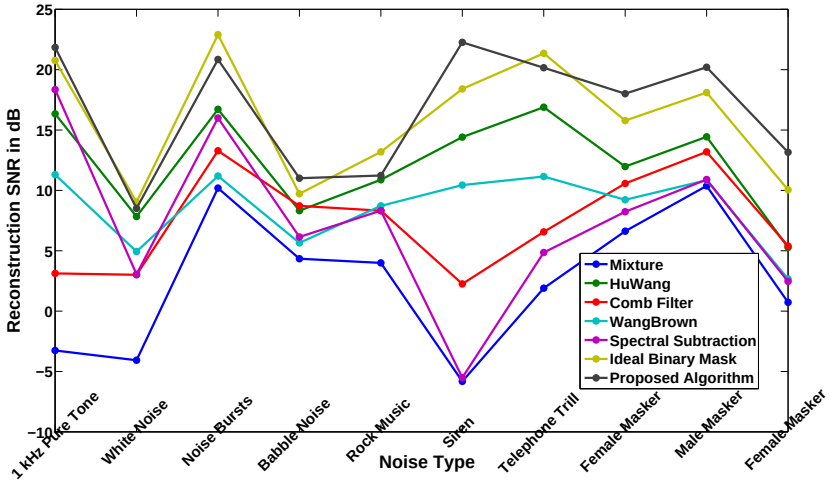

Figure 5.1: Performance of the segregation algorithm on the Cooke database in comparison with other segregation and enhancement algorithms

the noise/interference is of a periodic nature. This could possibly be attributed to the fact that the proposed algorithm applies a harmonic model to tease apart the two periodic signals and allot energy to *both* the participating sources, while the other algorithms *and* the binary mask assign energy to only one of the sources. As such, the reconstruction provided by the proposed algorithm should be perceptually better than the others, since there are fewer spectro-temporal holes in the reconstruction.

The algorithm is also evaluated on the larger database consisting of synthesized mixtures from the TIMIT database, where the mixtures are combined in different TMRs. It must be noted here that the pitch estimates for this speech segregation task were automatically extracted using the algorithm described in Chapter 2. The pitch was then assigned to the appropriate speaker according to the method described in Section 4.5. However, in regions where the pitch assignment was made incorrectly, the speaker assignment was then corrected based on the true pitch values obtained from the clean signals. For this experiment, the average percentage of incorrect assignment was approximately 40% for TMR $\pm$ 9 dB, 22% for TMR $\pm$ 6 dB, 9% for TMR $\pm$ 3 dB and 4.5% for TMR 0 dB. Fig. 5.2 shows the performance of the proposed segregation algorithm on the dataset. As can be seen, the Error Loss and Noise Residue figures follow a similar trend for the three gender sets. The values of P_{ELC} & P_{NRC} are significantly lower than that of P_{ELD} & P_{NRD} (compare the scales of the two plots). This happens due to two factors - the numerators of both parameters reduce since the algorithm allots energy on a "shared" basis and therefore less units with significant energy are missed or falsely detected. At the same time, the denominator increases since the total number of non-silent elements is much larger (ITF_{COM}) than the energy-dominating units (ITF_{DOM}).

From the top two plots, a few observations can be made. The low values of P_{ELC} indicate that the overall number of TFUs that were missed by the algorithm is very small, since compared to the energy of all the non-silent frames their contribution is less. On the other hand, the values of the P_{ELD} indicate that those TFUs which were indeed missed were mostly the "dominant" TFUs rather than the shared TFUs, as they seemed to significantly affect the P_{ELD} but not the P_{ELC}. From the bottom plots, we see that the noise residue increases with increasing TMR for the P_{NRD} but reduces for the P_{NRC}. This might indicate that the residue energy might be significant if only the ITF_{DOM} was used for reconstruction, but since we are estimating the ITF_{COM} the perceptual effect of the noise residue might be very less.

In order to obtain more information about these hypotheses, we will next look at the SNRs of the recovered signals.

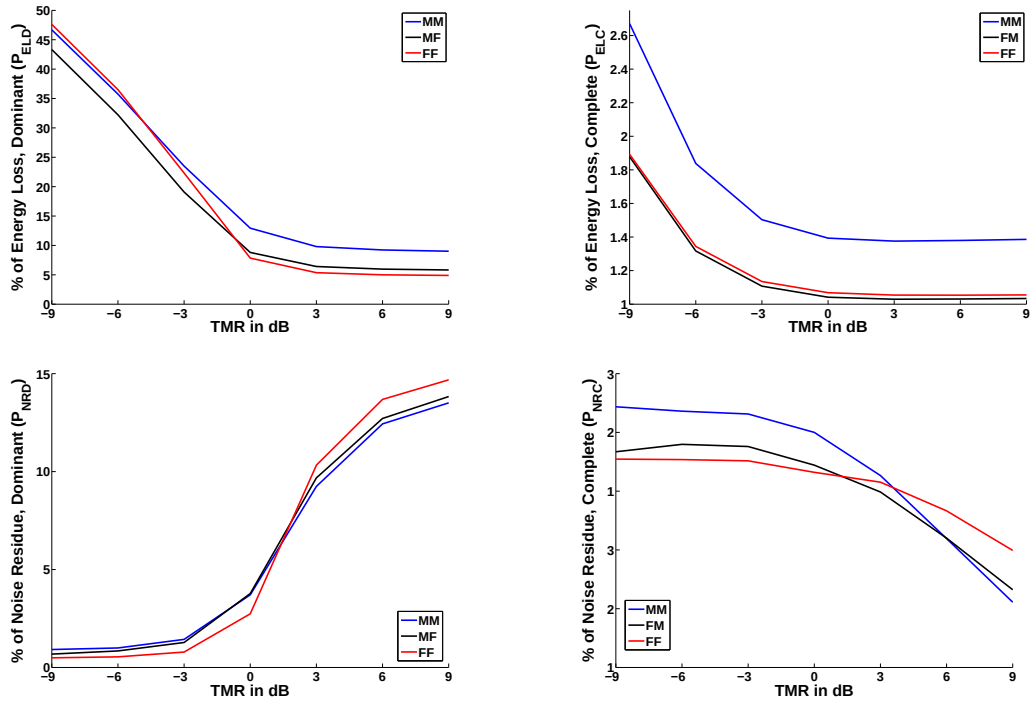

Figure 5.2: Performance of the segregation algorithm on the TIMIT database. The percentages of energy loss and noise residue are shown for both the ITF_{DOM} and ITF_{COM} at different TMRs.

5.1.3 Experiment 2: Signal to Noise Ratio (SNR) of Reconstructed Speech

It may so happen that the TFUs estimated as containing energy due to speaker A may truly contain energy from A, but the actual signal estimated may be incorrect. As an example, consider a certain TFU which contains 10 dB of *voiced* energy and dominates over the same TFU of the speaker B, which contains 4 dB of energy. If the proposed speech extraction algorithm estimates that the energy of A is greater than that of B in that TFU, it means that the ITF estimation for that TFU is correct. If it estimates that the target speech energy is 9.1 dB and the masker speech energy is 2.9 dB, the algorithm has also done a good job at extracting the relative contributions of the two speakers in the mixture. However, if the algorithm estimates that the actual signal of speaker A is an unvoiced signal, then the estimates we obtain are useless for speech reconstruction. Thus, it is not just important to simply estimate the ITF_{DOM} correctly; it is also necessary to estimate the actual signals accurately, especially their temporal structure. In order to evaluate the accuracy of reconstruction, we therefore look at the SNRs of the reconstructed target signals, defined as the ratio between the energy of the target signal to the energy of the error in reconstruction:

$$SNR_{target} = \frac{\sum_n (s_{target}[n])^2}{\sum_n (s_{target}[n] - \hat{s}_{target}[n])^2} \tag{5.5}$$

Ideally, this figure must be as high as possible. Note that here the SNR is calculated for the entire signal over all TFUs, irrespective of whether they are dominant or not, since we want to get as good a reconstruction of the global original signal as possible.

The fact that the non-silent TFUs detected are not erroneous decisions and indeed belong to the correct speaker is confirmed by the SNR plots of the recovered signals, shown in Fig. 5.3. The reconstruction SNR is shown for both the speech-in-noise and speech-in-speech scenarios. It can be seen from the figure from both figures that even at very low SNRs or TMRs (-9 dB), the SNR of the recovered signal is greater than 0 dB. This implies that even in regions where the target speech is very weak, the harmonic model is able to extract some speech and yields a signal that is

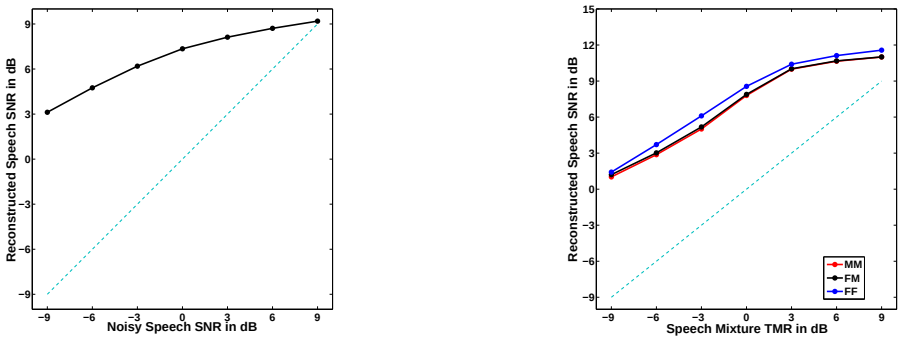

Figure 5.3: Performance of the speech extraction algorithm on the TIMIT noisy speech database in terms of SNR (left) and the TIMIT speech mixture database (right). It may be seen that even at very low SNRs and TMRs (-9 dB), the SNR of the reconstructed signal is greater than 0 dB for all population sets. In particular, the improvement is very high in the left regions of both axes, which represent weak target-to-interference regions.

closer to the clean speech than the original unprocessed one, yielding speech that is "more usable than the original unrpocessed signal". An analysis of the speech signals showed that at TMR $<$ -3 dB, the target speech is almost not perceptible; yet usually in the reconstructed stream more than half of the target speaker is well-recovered and also audible/perceptible. This is a qualitative method of confirming that the SNR is indeed increased by the proposed algorithm and that we are given a "more usable" version of the target speech. This observation demonstrates that even at very low TMRs, the streams of the participating can be pulled apart and good quality segregation is indeed possible.

5.1.4 Experiment 3: Perceptual Evaluation of Speech Quality (PESQ) of Reconstructed Speech

While the SNR is a good measure to evaluate how well a reconstructed signal matches with the original one, it does not take into account the perceptual attributes of a speech signal. In particular, the human auditory system emphasizes the lower frequencies more than the high frequencies during speech perception [4, 34]. Similarly, the auditory system focuses more on the formant regions of the speech than the formant valleys [34]. Thus, if a certain reconstructed speech had low SNR than another alternate reconstruction. but gave more importance to and preserved the low-frequency regions (or in the alternative to the formant regions) it might still be perceptually more preferable than the other signal with a higher SNR. Thus, it would be instructive to evaluate the proposed algorithm on an objective measure that would account for the perceptually preferable attributes of the speech signal. As mentioned in the introduction to this chapter, such measures are not yet standardized and literature reports various different measures, and we will stick to the PESQ scores here for evaluation.

The PESQ model [39] is an ITU-T recommendation that is used to test the quality of speech transmitted through a channel with variable distortion and delay characteristics. The model has been validated by comparing its predictions of perceived speech quality to actual reported values by humans, and the correlation between the predicted and true values has shown to be the best among existent measures of speech quality evaluation. In addition to its close predictions of human perception of the input speech signal, the PESQ measure is also robust to the effects

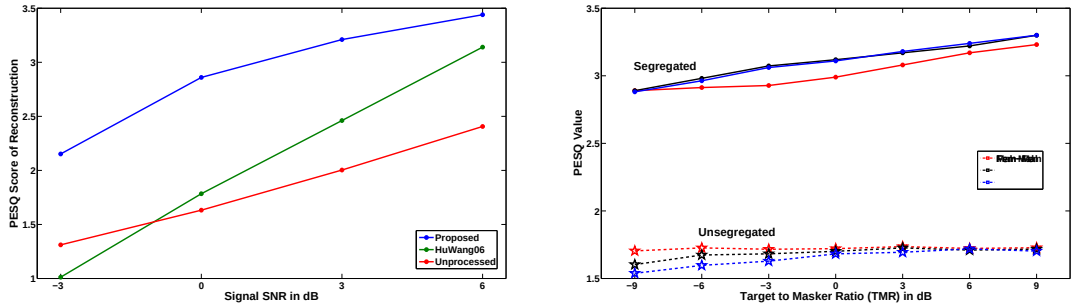

Figure 5.4: PESQ scores of speech signals processed by the proposed segregation algorithm. Both speech enhancement (left) and speech segregation (right) results are shown, averaged on different types of noise.

of channel filtering, distortion, delays, packet/data loss, background noise etc. and gives good performance across a variety of channel, codec and network conditions. As such, it is a good choice to evaluate the segregation algorithm, since it gives us a good quantitative estimate of how well the reconstructed speech signals sound like, i.e., how a prospective subject would judge the quality of the reconstructed signals. The PESQ measure takes into account the non-linear loudness mapping, Bark scale transformation and variable gain transmission in the human auditory system. The PESQ measure has been used in the past for evaluation of speech enhancement algorithms. It can take on values between 1.0 (bad signal) and 4.5 (no distortion; perfect reconstruction) in typical settings. In extreme conditions, the value can also go below 1. We will use this measure for evaluating the proposed segregation algorithm on both the speech segregation and speech enhancement tasks. It may be noted here that there is no standard database or task on which authors of various algorithms have presented PESQ scores. We will therefore restrict ourselves to presenting the performance of the proposed algorithm in this section without comparisons for the speech segregation task. For the speech enhancement task, we will compare our performance with the code for the segregation algorithm in [23] which was used as-is for enhancement. In addition to the SNR-based comparison of section 5.1.2, the comparison of the ASR performance discussed in section 5.1.2 should give a fair idea of the performance of the proposed algorithm in comparison with other segregation or enhancement algorithms in the literature in terms of objective criteria.

In this experiment, we will use the IEEE database - a different database from the one which was used to train the neural network for finding the mapping functions. This will thus help in judging the generalizability of the proposed algorithm. The PESQ values of the noisy signals are compared with the PESQ values of the enhanced versions. The plots of these two quantities with increasing SNR are shown in Fig. 5.4. It can be seen that the algorithm yields perceptually more favorable speech signals than the original noisy ones. Furthermore, the improvement in the PESQ value is consistent irrespective of the SNR of the signal, i.e., it does not deteriorate with falling SNR. Finally, we see that the PESQ scores of the proposed algorithm are consistently better than the scores for the algorithm proposed in [23] across all SNRs.

We also test the PESQ performance of the segregation algorithm on a database of speech mixtures from the IEEE database. The aim is to extract the target speaker from the mixture in all TMR conditions. The PESQ values are calculated in the same method as described above. Fig. 5.4 shows the PESQ values compared for different TMR conditions in all gender combinations, with the basic PESQ values of the mixtures also shown. It can be seen that the PESQ values are increased significantly for all TMRs, even when the target is significantly weaker than the masker. In reality, for the TMR = -6 dB case, the target is almost inaudible in the mixture when these samples are played but the segregated signals show a great semblance to the original signal even when extracted from such low TMR situations. This is supported by the PESQ numbers shown in the figure. Furthermore, the increase in performance is consistent for all TMRs and all gender combinations.

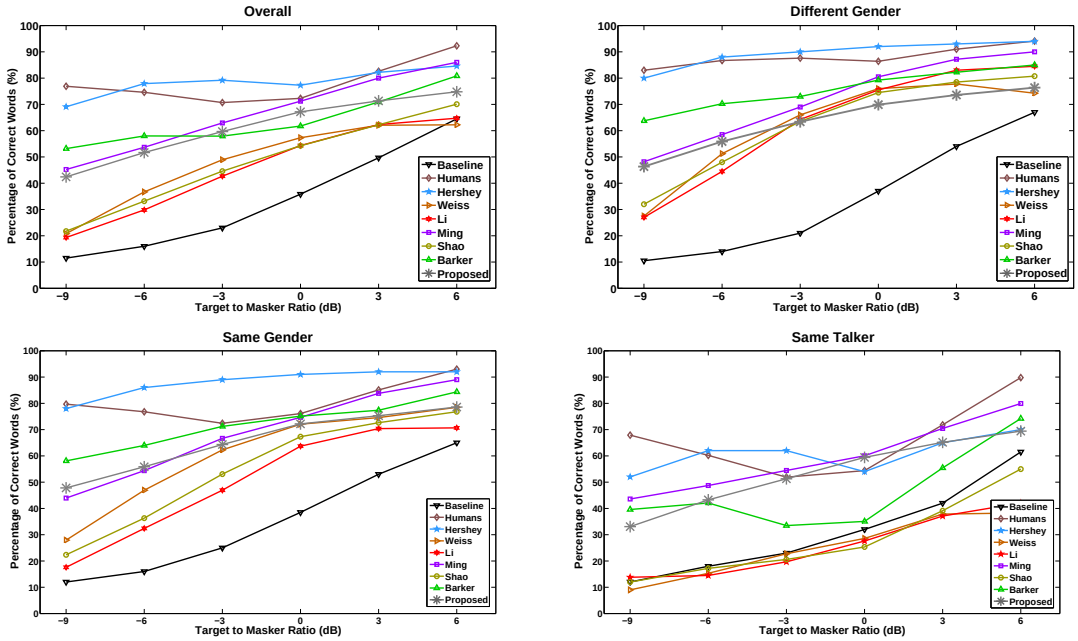

Figure 5.5: ASR performance of the proposed algorithm on the SSC task. The x-axis represents the Target-to-Masker Ratio (TMR) in dB, and the y-axis shows the word recognition rate in percentage.

5.1.5 Experiment 4: Performance on an Automatic Speech Recognition (ASR)

Task

We evaluate the segregation performance on an ASR task, called the Speech Separation Challenge (SSC), on the SSC database described above. A number of segregation and enhancement algorithms have participated in the SSC using different approaches. Most of these approaches can be classified as supervised algorithms, in the sense that these algorithms use prior speaker or speech models constructed from training data to assist in either segregation or recognition. The other set of algorithms that perform segregation on-the-fly can be called unsupervised segregation schemes. Since the proposed algorithm does not rely on any prior speaker models and estimates the segregated streams without using any prior data, we are ideally comparable to the performance in ASR with algorithms that fall under the second category. The algorithm by Shao et al [42] is the other algorithm that is purely bottom-up or independent of any prior speech models or speaker models. The algorithm by Barker et al [1] can be considered as a hybrid of supervised and unsupervised approaches, since it first identifies the dominant spectro-temporal regions just as the unsupervised methods do, and then fills the spectro-temporal holes by relying on Missing Feature Theory and by relying on task-specific grammar models. The algorithms of Hershey et al [21], Weiss et al [51], Li et al [27] and Ming et al [33] all rely on prior models of either the speech data, or the speaker data, and in some cases also the task-specific grammar, to perform ASR. As such, their systems are greatly tuned for performance with the SSC task, but may not be as generic as the unsupervised ones. Furthermore, most of these model-based approaches (with the exception of Weiss et al [51]) are actually designed only for the speech recognition task, and are not intended to reconstruct the speech signal in the acoustic domain - thus not much can be said about their performance for perceptual tasks. It may further be noted that the algorithms by Barker et al [1], Weiss et al [51], Shao et al [42] and Li et al [27] also rely on the concept of the dominated TF masks during segregation.

Fig. 5.5 shows the results of ASR performance of the proposed algorithm in comparison to other reported algorithms. As can be seen from the plot, in comparison to Shao et al [42] which is "comparable" to the proposed algorithm in terms of the supervised versus unsupervised paradigm, the proposed algorithm shows recognition rates that are significantly better. In particular, for TMRs less than 0 dB, the proposed algorithm has a performance better than algorithms which rely on the dominated TF masks. This is consistent with our assertion that for TMRs < 0 dB, the other algorithms would not be very efficient in extracting the target stream since the target would be weaker and there would be less dominant TFUs which the other algorithms would be able to exploit. On the other hand, since our proposed algorithm tries to share the energy between both participating speakers, it shows better performance when the target is weaker than the masker. Thus, our algorithm is more robust to adverse conditions and gives the best performance among all reported algorithms that do not use pre-trained models for segregation and work in an unsupervised fashion. In particular, it may be noted here that the algorithm performs a good job at separating the energies apart, since the performance is better than those algorithms which assign energy to the dominant speaker. In general, in comparison with other algorithms, the performance of the proposed algorithm is among the middle-ranked ones. This is shown in Fig. 5.5.

It may be noted at this point that the proposed algorithm relies on a very primitive speaker assignment technique that does not rely on any prior speaker trained models. As such, the pitch assignment of this algorithm is not optimal. In spite of that, the ASR results are comparable to most of the ones presented for the SSC challenge, including algorithms which rely on speaker models for the assignment of speech. It is therefore reasonable to expect that if the speaker assignment block of the proposed algorithm were improved, the ASR results would be much better and be comparable to the superior ones presented in the separation challenge. Indeed, some preliminary tests indicated that the major problem in the reconstruction of the target speech was the pitch assignment once the signals were segregated. We found that the proposed segregation system yielded segregated signals of *very good* perceptual quality when pitch assignments are made accurately (i.e., manually corrected to go to the right source speaker). Therefore, a better functioning speaker assignment block promises to yield even better ASR results than what is presented here. Since we do not intend to rely on prior speaker models, and would like to arrive at an alternate method of speaker assignment which would be able to work on fly (which is a very difficult task due to the algorithm's lack of sufficient knowledge for speaker assignment), designing such an algorithm would be part of the future work.

Further, it may be noted that the supervised algorithms use the speech from the training data not just in training the ASR backend, but also in the segregation process. This luxury may not be always available in all practical scenarios. On the other hand, the proposed algorithm does not assume any prior information about the available speech data. The algorithm ranks 4 among all reported algorithms, including the supervised and unsupervised class. It also outranks two algorithms from the supervised class, for both the low-TMRs and high-TMRs. Finally, this performance is consistent across TMRs, and is not sensitive to the TMR.

5.1.6 Experiment 5: Perceptual Subjective Evaluation of Recovered Speech

The speech signals obtained by processing through a segregation algorithm typically serves one of two purposes - increasing perceptual clarity of some algorithms that perform other speech processing steps (for example, cochlear implant applications) or improving the performance of some artificial intelligence systems that rely on speech processing applications as a pre-processing step for other applications (like Automatic Speech Recognition). The proposed segregation system must ultimately find use in typical speech processing applications, like automatic speech recognition (ASR) or speaker identification.

As such, we will evaluate the performance of the segregation algorithm on a subjective speech recognition task. The database used for our subjective evaluation is the IEEE database refered to above, with emphasis on the speech segregation task. The speech mixtures were also

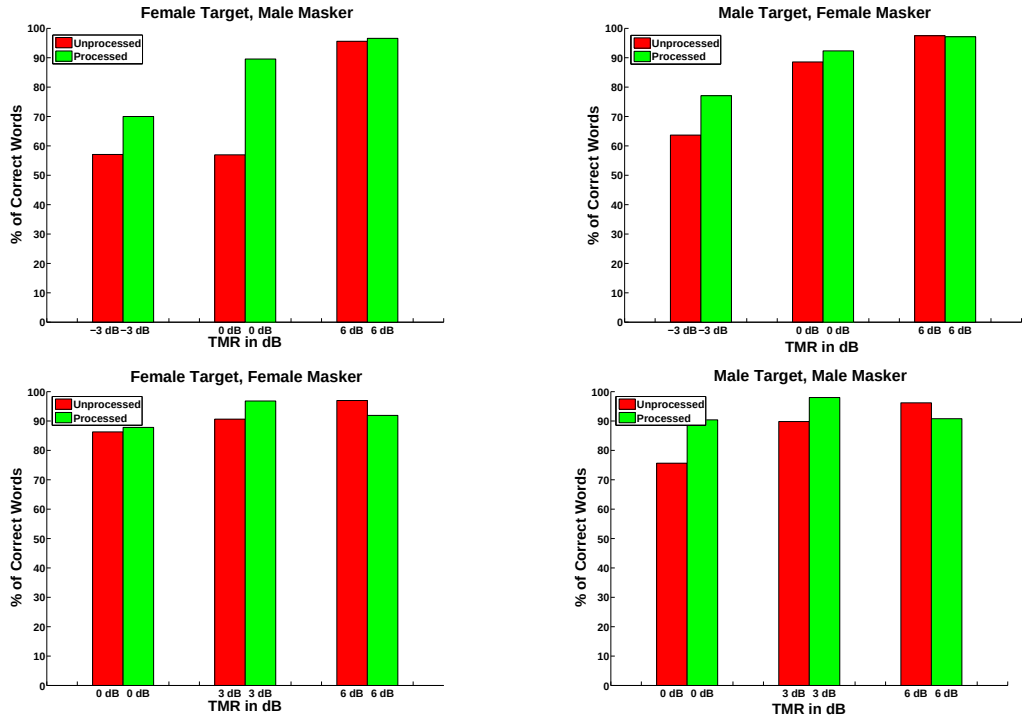

Figure 5.6: Perceptual evaluation of the segregation algorithm on the IEEE database. The percentage of correct words reported by the subjects is shown on the y-axis, with the TMR on the x-axis.

processed by the proposed segregation algorithm to recover the target speech signals from the mixtures. Twenty-four subjects, all native American English speakers with no hearing disabilities, were invited to participate in the experiment. Four different datasets were investigated: (Male Target, Female Masker), (Female Target, Male Masker), (Female Target, Female Masker) and (Male Target, Male Masker) - requiring six subjects for each dataset. The task for the subject was to recognize the target speech signal in each case. The target reports were scored based on the percentage of correct words recognized. The subjects were presented with a total of 48 sentences to recognize, of which the first 12 were meant for training the subject to the task, and not scored during evaluation. The total dataset contained 12 sentences for each TMR, 6 of them being unprocessed speech mixtures and the other 6 being the processed speech obtained by the proposed segregation algorithm. During the subjective evaluation, care was taken to ensure that for a given dataset, half the subjects were given those speech mixtures for which the other half were given the corresponding processed speech so as to phonetically balance the resulting reports.

The results of the subjective tests are shown in Fig. 5.6. The percentage of correct words is shown on the y-axis. It can be seen that for all four datasets, the perceptual scores are better for the segregated speech targets than for the unprocessed speech mixtures. In particular, at low TMRs, the proposed algorithm yields perceptually more preferrable speech signals, which justifies the use of the proposed algorithm for speech segregation.

5.2 Chapter Summary

In this chapter, we have described the performance of the speech segregation and enhancement algorithms on various different objective tasks and the segregation algorithm on a subjective

task. The proposed speech extraction model is capable of segregating overlapping voiced speech, as well as voiced speech from unvoiced speech. The utility of the segregated signals resulting from the algorithm has been demonstrated on various performance evaluation criteria, showing improved performance compared to the state of the art. We will next discuss some directions for future research.

Chapter 6

THESIS SUMMARY AND FUTURE WORK

6.1 Thesis Summary

In this thesis, we have developed and described a speech extraction algorithm that is meant to recover speech signals from either noisy speech or speech mixtures. The algorithm has been developed with the aim of extracting target speech from any kind of interference, and is a feature-based, bottom-up approach that does not rely on prior statistically trained speech or speaker models. As such, the algorithm should be able to deal with any kind of interference, and separate out the participating streams in the speech mixture. It does not rely on any assumptions of the interfering speech or noise signals, nor does it rely on any prior speech or speaker data for good performance - this guarantees its applicability for a wide variety of applications. Currently, the algorithm has been designed to segregate a maximum of two speakers.

The algorithm includes a multi-pitch detector which identifies the pitch estimates of one or both speakers in the speech mixture. The multi-pitch detector is based on a 2-D AMDF function which evaluates the periodicity content of the input signal along a two-dimensional function - the two-dimensional point which optimizes this function yields the pair of pitch estimates. The voiced regions are then extracted based on a least squares model which is set up based on the pitch estimates identified from the earlier block. It is shown that these estimates, in the case of speech enhancement, are equivalent to a simple temporal averaging operation. With this interpretation, we then extend the ehnhancement algorithm to model the regions of the speech which have not been modeled by the least-squares model. By identifying and relying on certain features of the modeled (and extracted) portions of the speech signal, we try to extract additional information regarding the aperiodic contributions as well as the noise contributions in the voiced and unvoiced regions. Simultaneously we also modify our knowledge of the estimates of the voiced regions based on their reliability. We then put together all these estimates of the periodic and aperiodic regions, as well as the information from the noise regions, to yield the final speech reconstructions. We later extend this approach in an analogous fashion to the speech segregation case. Therein, the model yields estimates of the periodic components in the voiced regions. We then use these estimates to extract information of the aperiodic components and noise components of both the target and masker speech signals. A speaker assignment stage delegates the extracted components to their corresponding sources based on certain features extracted from these extracted estimates. Finally, all those components have been hypothesized to have come from the same source (speaker) are combined together to yield the final speech reconstructions.

We then evaluate the segregation algorithm on various objective tasks, and a subjective test. Using various criteria, both purely theoretical (like the SNR) as well as perceptually motivated (like the PESQ score), we show the utility of the algorithm for dealing with various kinds of interferences. We also illustrate the performance of the algorithm on an automatic speech recognition task. Perceptual tests reveal to us that the proposed segregation algorithm improves the intelligibility of speech after the noisy mixture signals have been procssed by the algorithm. We thus demonstrate that a single speech extraction algorithm can be used under various noisy conditions to yield target speech signals which are useful for various different tasks tha and which perform well according to several evaluation metrics.

6.2 Future Directions

There are still some open and interesting questions to address as extensions of the work accomplished in this thesis:

- ***Increased Robustness to Noise***: Noise is always a challenging problem in speech processing applications, and there is always a constant need to improve robustness of speech extraction systems irrespective of their current performance. Especially in the case of cellular communication, the variety of background interferences (both speech and noise) and adverse conditions (very low TMRs or SNRs) raise increasingly difficult challenges of preserving perceptual quality while eliminating the background. The current algorithm shows good promise until moderate TMRs and SNRs, but needs more work in very low TMRs and SNRs ($<$ 3dB).In particular, most pitch trackers fail to achieve good estimates of the voiced regions in such adverse conditions, and since the proposed algorithm heavily relies on voicing detection, its performance is expected to go down in such scenarios. Possible approaches to handle such situations include combining the proposed algorithm with noise-suppression algorithms like spectral subtraction so as to improve the SNR for pitch detection and to obtain better speech enhancement or segregation.

- ***Better Recovery of Unvoiced Regions***: The algorithm proposed here exploits the properties of human perception to partially recover the unvoiced regions of the target speech signal, by adding back information about aperiodic regions immediately following periodic (voiced) regions. However, there can be a significant loss of information when the aperiodic regions preceding voiced regions are missed (which is expected to occur in the context of the algorithm propsoed in this thesis). As such, there is a need to better recover the unvoiced regions. In the unsupervised setting, this causal estimation of unvoiced speech is an extremely challenging problem, especially in the presence of stationary noise which greatly resembles unvoiced speech. Solutions to this problem may require exploring other methods, possibly relying on models which characterize unvoiced speech.

- ***Improving the Speaker Assignment Strategy***: The assignment of the extracted speech signals to the appropriate speakers is an important component of the speech segregation algorithm. Even if an algorithm estimates the components of the mixture signal accurately but assigns them to the wrong source over time, the reconstructed speech streams can sound perceptually unacceptable. In fact, it is the belief of the author that this is the primary reason why the ASR performance in section 5.1.5 was not up to the mark, though the proposed segregation system yielded segregated signals of very good perceptual quality when pitch assignments are made accurately. As discussed in section 4.5, the method of speaker assignment is currently based on using the MFCC features of the extracted speech signals to assign them to the appropriate source. However, the experiments by Mahadaven [31] have shown that the approach followed here shows reliable assignment only in the voiced regions (i.e., intra-segment grouping), but not as reliably across voiced regions (i.e., inter-segment grouping). An important task for the improvement of the algorithm would therefore be to enhance the performance of this block. While there are model-based algorithms which can achieve good performance for this functionality by using speaker data (e.g., see [7] for an overview), we would like to do this without reliance on prior speaker data. We are currently looking at using additional features of the extracted speech signals to supplement the MFCCs. A promising idea seems to be the usage of the coefficients of the neural networks which were used to estimate the unvoiced regions: the temporal behavior of these coefficients showed interesting properties in a preliminary study, and will be a subject of more detailed analysis.

- ***Extension of the algorithm to multiple speakers***: As mentioned at the outset of this thesis, the methods and results presented here are applicable for a maximum of two simultaneous speakers, but are generalizable to a larger number of speakers. In particular, the multi-pitch algorithm can be extended by considering a 3-D AMDF to estimate three simultaneous pitch periods, and our preliminary studies have already shown promising results.

However, in practice, the calculation of the 3-D AMDF can involve the usage of a much longer analysis window than the stationarity assumption can afford. Therefore we will first have to explore the consequences of loss of the stationarity assumption, and the performance degradation as this analysis window length is changed. Similar observations apply for the segregation block itself, since there are three sets of harmonics to be estimated and therefore a larger number of coefficients - thus requiring a larger window to solve the set of least squares equations. The three speaker case would also bring in additional problems regarding learning the scaling function for reliability, estimating the aperiodic and noise power etc. Finally, the speaker assignment problem in that case would become even more tougher to solve. As such, in principle, the algorithm can be extended to multiple speakers theoretically but this has several practical ramifications which need to be explored. That will also be a direction of exploration following from this thesis.

Bibliography

[1] Jon Barker, Ning Ma, André Coy, and Martin Cooke. Speech fragment decoding techniques for simultaneous speaker identification and speech recognition. *Comput. Speech Lang.*, 24:94–111, January 2010.

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.

[4] A. S. Bregman. *Auditory Scene Analysis*. MIT Press: Cambridge, MA, 1990.

[5] Martin Cooke. *Modelling auditory processing and organisation*. Cambridge University Press, New York, NY, USA, 1993.

[6] Martin Cooke and Guy J. Brown. Separating simultaneous sound sources: issues, challenges and models. pages 295–312, 1994.

[7] Martin Cooke, John R. Hershey, and Steven J. Rennie. Monaural speech separation and recognition challenge. *Comput. Speech Lang.*, 24(1):1–15, 2010.

[8] M. Davy and S. J. Godsill. Bayesian harmonic models for musical signal analysis. In *in Bayesian Statistics 7*. Oxford University Press, 2002.

[9] Alain de Cheveigné. Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model of auditory processing. *The Journal of the Acoustical Society of America*, 93(6):3271–3290, 1993.

[10] Alain de Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[11] Stefano De Marchi and Robert Schaback. Stability of kernel-based interpolation. *Advances in Computational Mathematics*, 32:155–161, 2010. 10.1007/s10444-008-9093-4.

[12] O. Deshmukh, C.Y. Espy-Wilson, A. Salomon, and J. Singh. Use of temporal information: Detection of periodicity, aperiodicity, and pitch in speech. *Speech and Audio Processing, IEEE Transactions on*, 13(5):776 – 786, sep. 2005.

[13] Om D. Deshmukh, Carol Y. Espy-Wilson, and Laurel H. Carney. Speech enhancement using the modified phase-opponency model. *The Journal of the Acoustical Society of America*, 121(6):3886–3898, 2007.

[14] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. In *NIPS*, pages 155–161, 1996.

[15] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, London, 1993.

[16] Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(6):1109 – 1121, dec. 1984.

[17] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):443–445, January 2003.

[18] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, and Victor Zue.

[19] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition, July 1998.

[20] Simon Haykin. *Adaptive Filter Theory (4th Edition)*. Prentice Hall, September 2001.

[21] John R. Hershey, Steven J. Rennie, Peder A. Olsen, and Trausti T. Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Computer Speech & Language*, 24(1):45 – 66, 2010. Speech Separation and Recognition Challenge.

[22] Wolfgang Hess. *Pitch Determination of Speech Signals: Algorithms and Devices*. Springer Series in Information Sciences, 1983.

[23] Guoning Hu and Deliang Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Trans. Neural Networks*, 15:1135–1150, 2004.

[24] Yi Hu, Philipos C. Loizou, and Senior Member. Evaluation of objective quality measures for speech enhancement. *IEEE Trans. Audio Speech Language Processing*, pages 229–238, 2008.

[25] Hirokazu Kameoka, Takuya Nishimoto, and Shigeki Sagayama. A multipitch analyzer based on harmonic temporal structured clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(3):982 –994, mar. 2007.

[26] Ning Li. Contribution of acoustic landmarks to speech recognition in noise, 2009.

[27] Peng Li, Yong Guan, Shijin Wang, Bo Xu, and Wenju Liu. Monaural speech separation based on maxvq and casa for robust speech recognition. *Comput. Speech Lang.*, 24:30–44, January 2010.

[28] Yipeng Li and DeLiang Wang. On the optimality of ideal binary time-frequency masks. *Speech Communication*, 51(3):230 – 239, 2009.

[29] Philip Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[30] Philipos C. Loizou. Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum. In *IEEE Trans. Speech Audio Proc.*, pages 857–869, 2005.

[31] Vijay Mahadevan. Sequential grouping in co-channel speech, 2011.

[32] Ray Meddis and Michael J. Hewitt. Virtual pitch and phase sensitivity of a computer model of the auditory periphery. i: Pitch identification. *The Journal of the Acoustical Society of America*, 89(6):2866–2882, 1991.

[33] Ji Ming, Timothy J. Hazen, and James R. Glass. Combining missing-feature theory, speech enhancement, and speaker-dependent/-independent modeling for speech separation. *Comput. Speech Lang.*, 24:67–76, January 2010.

[34] Brian C. J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, fifth edition, April 2003.

[35] W. Pachl, G. Urbanek, and E. Rothauser. Preference evaluation of a large set of vocoded speech signals. *Audio and Electroacoustics, IEEE Transactions on*, 19(3):216 – 224, sep. 1971.

[36] H. Vincent Poor. *An introduction to signal detection and estimation (2nd ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1994.

[37] T.F. Quatieri and R.G. Danisewicz. An approach to co-channel talker interference suppression using a sinusoidal model for speech. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(1):56 –69, jan. 1990.

[38] T.F. Quatieri and R.G. Danisewicz. An approach to co-channel talker interference suppression using a sinusoidal model for speech. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(1):56 –69, January 1990.

[39] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *Proceedings of the Acoustics, Speech, and Signal Processing, 200. on IEEE International Conference - Volume 02*, pages 749–752, Washington, DC, USA, 2001. IEEE Computer Society.

[40] Ali H. Sayed. *Adaptive Filters*. Wiley-IEEE Press, 2008.

[41] M. R. Schroeder. Period histogram and product spectrum: New methods for fundamental-frequency measurement. *The Journal of the Acoustical Society of America*, 43(4):829–834, 1968.

[42] Yang Shao, Soundararajan Srinivasan, Zhaozhang Jin, and DeLiang Wang. A computational auditory scene analysis system for speech segregation and robust speech recognition. *Comput. Speech Lang.*, 24(1):77–93, 2010.

[43] G. S. Stickney, F.-G. Zeng, R. Litovsky, and P. Assmann. Cochlear implant speech recognition with speech maskers. *Acoustical Society of America Journal*, 116:1081–1091, August 2004.

[44] David Talkin. A robust algorithm for pitch tracking (rapt). 1995.

[45] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory: Radar-Sonar Signal Processing and Gaussian Signals in Noise*. Krieger Publishing Co., Inc., Melbourne, FL, USA, 1992.

[46] S. Vishnubhotla and C.Y. Espy-Wilson. An algorithm for multi-pitch tracking in co-channel speech. In *Proc. of the Intl. Conf. on Spoken Language Processing (Interspeech 2008)*, Sep. 2008.

[47] S. Vishnubhotla and C.Y. Espy-Wilson. An algorithm for speech segregation of co-channel speech. In *Acoust., Speech & Signal Pro. 2009, IEEE Intl. Conf. on*, pages 109–112, April 2009.

[48] S. Vishnubhotla and T. Pruthi. An efficient implementation of a speech enhancement algorithm, in preparation.

[49] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[50] R. J. Weiss and D. P. W. Ellis. Monaural Speech Separation Using Source-Adapted Models. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 114–117, New Paltz, USA, October 2007.

[51] Ron J. Weiss and Daniel P. W. Ellis. Speech separation using speaker-adapted eigenvoice speech models. *Comput. Speech Lang.*, 24:16–29, January 2010.

[52] Mingyang Wu, DeLiang Wang, and G.J. Brown. A multipitch tracking algorithm for noisy speech. *Speech and Audio Processing, IEEE Transactions on*, 11(3):229 – 241, may. 2003.

[53] M. A. Zissman and D. C. Seward. *Two-Talker Pitch Tracking for Co-Channel Talker Interference Suppression*. Tech. Rep. Lincoln Lab., MIT, Cambridge, Apr. 1992.