2017

Early adductive reasoning for blind signal separation

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

EARLY ABDUCTIVE REASONING FOR BLIND SIGNAL SEPARATION

by

PINAR ÖZDEMIR

B.S., Gaziantep University, 2007
M.S., Boston University, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2017

Approved by

First Reader _____

S. Hamid Nawab, Ph.D.
Professor of Electrical and Computer Engineering
Professor of Biomedical Engineering

Second Reader _____

W. Clem Karl, Ph.D.
Professor and Chair of Electrical and Computer Engineering
Professor of Systems Engineering
Professor of Biomedical Engineering

Third Reader _____

Jeffrey B. Carruthers, Ph.D.
Associate Professor of Electrical and Computer Engineering

Fourth Reader _____

Osama Alshaykh, Ph.D.
Assistant Research Professor of Electrical and Computer Engineering

Acknowledgements

I would like to express my sincere appreciation to Professor S. Hamid Nawab for his invaluable support and advice during my research. He has taught me the philosophy and method of research, and has helped me shape my research career. He also taught me how to write and present technical materials for which I am forever grateful to him. I would especially like to thank him for his dedication to making sure that I am fully prepared for a professional carrier.

I would also like to thank Professor W. Clem Karl, Professor Jeffrey Carruthers, and Professor Osama Alshaykh for their participation as readers of this thesis. Their valuable comments have helped me refine the material presented in this document to make it better written. I would also like to acknowledge Professor Vivek Goyal for chairing my oral defense.

My sincere thanks also go to Ken Sutton of Yobe Inc. for providing me an opportunity to join as an intern, and who giving me access to the Yobe database and research facilities. I would also like to thank Shibani Abhyankar, Sami Shahin, and Alexander Stooss for their support and helpfulness.

I owe a debt of gratitude to my lab-mates and friends during my time at BU, for their endless support and encouragement over these years. Specifically, I would like to thank my current and former lab-mates: Wenyang Zhang, Siddhant Sharma, Narasimha Chakravarty, Bryan Cole, Seif Abu Bakr. I also would like to thank the staff at Boston

University Electrical and Computer Engineering Department, especially Sabrina Renee Salvati.

I would also like to say thank you to my parents, Ummahan Ozdemir and Halil Ibrahim Ozdemir, who encouraged me and calmed me down whenever I felt out of breath in this journey. Without their support, this work could never have been achieved. I would like to thank my husband, Osman Doner, who joined my life 2 years ago and supported me during my journey. Lastly, I would like to thank all my friends and family for their support and encouragement.

EARLY ABDUCTIVE REASONING FOR BLIND SIGNAL SEPARATION

PINAR ÖZDEMIR

Boston University, College of Engineering, 2017

Major Professor: S. Hamid Nawab, Ph.D. Professor of Electrical and Computer Engineering

ABSTRACT

We demonstrate that explicit and systematic incorporation of abductive reasoning capabilities into algorithms for *blind* signal separation can yield significant performance improvements. Our formulated mechanisms apply to the output data of signal processing modules in order to conjecture the structure of time-frequency interactions between the signal components that are to be separated. The conjectured interactions are used to drive subsequent signal separation processes that are as a result *less blind* to the interacting signal components and, therefore, more effective. We refer to this type of process as *early* abductive reasoning (EAR); the "early" refers to the fact that in contrast to classical Artificial Intelligence paradigms, the reasoning process here is utilized *before* the signal processing transformations are completed.

We have used our EAR approach to formulate a *practical* algorithm that is more effective in realistically noisy conditions than reference algorithms that are representative of the current state of the art in two-speaker pitch tracking. Our algorithm uses the Blackboard architecture from Artificial Intelligence to control EAR and advanced signal processing modules. The algorithm has been implemented in MATLAB and successfully tested on a database of 570 mixture signals representing simultaneous speakers in a variety of real-world, noisy environments. With 0 dB Target-to-Masking Ratio (TMR)

and no noise, the Gross Error Rate (GER) for our algorithm is 5% in comparison to the best GER performance of 11% among the reference algorithms. In diffuse noisy environments (such as street or restaurant environments), we find that our algorithm on the average outperforms the best reference algorithm by 9.4%. With directional noise, our algorithm also outperforms the best reference algorithm by 29%. The extracted pitch tracks from our algorithm were also used to carry out comb filtering for separating the harmonics of the two speakers from each other and from the other sound sources in the environment. The separated signals were evaluated subjectively by a set of 20 listeners to be of reasonable quality.

Table of Contents

1 Chapter 1: Introduction 1

1.1 Multi-Speaker Pitch-Tracking Problem 5

1.2 Evaluation of Multi-Speaker Pitch Tracking 8

1.2.1 Error-Based Evaluation Methodology 10

1.2.2 Enhancement-Based Evaluation Methodology 12

1.3 Contributions of Thesis 14

1.4 Thesis Outline 15

2 Chapter 2: Pitch Tracking Background 17

2.1 Single-Speaker Pitch Tracking 18

2.2 Multi-Speaker Pitch Tracking 22

2.3 Chapter Summary 25

3 Chapter 3: Multi-Speaker Pitch Tracking Challenges 26

3.1 Multi- Speaker Pitch Tracking Techniques 28

3.1.1 2-D Average Magnitude Difference (2-D AMDF) 28

3.1.2 Multi-Pitch Tracker (MP TRACKER) 34

3.2 State of the Art Multi-Speaker Pitch Tracking Methods and Their Limitations and Challenges 41

3.3 Speech Enhancement Techniques 50

3.3.1 Coherence-Based Filtering 50

3.3.2 Phase Error-Based Filtering 54

3.3.3 Minimum Variance Distortionless Response (MVDR) 57
3.3.4 Cross-Correlation Subtraction 63
3.3.5 Harmonic Product Spectrum (HPS) 65
3.4 Chapter Summary 69
4 Chapter 4: Early Abductive Reasoning Approach 71
4.1 Appropriateness of Blackboard Architecture 72
4.2 Abductive Reasoning Process 74
4.3 Early Abductive Reasoning Process 78
4.4 Abductive Reasoning (EAR) Modules 81
4.4.1 Discrepancy Detection Module 81
4.4.2 Discrepancy Diagnosis Module 88
4.4.3 Reprocessing Planning and Reprocessing Module 93
4.5 Chapter Summary 97
5 Chapter 5: Evaluation of EAR- Based Algorithm on Speech Mixtures 98
5.1 Introduction 98
5.1.1 Database 98
5.1.2 Error-Based Evaluation Methodology 101
5.1.3 Enhancement-Based Evaluation Methodology 108
5.2 Chapter Summary 114
6 Chapter 6: Conclusion and Future Work 115
6.1 Conclusion 115

6.2 Future Directions 117

References 120

Curriculum Vitae 130

List of Tables

Table 1.1. Scale of signal distortion (SIG) 13

Table 1.2. Scale of background intrusiveness (BAK)..... 14

Table 4.1. Distortion Indicators and Descriptions 84

Table 4.2. Lookup table for identifying the discrepancies..... 86

Table 4.3. Distortion Operators 89

Table 4.4. Abductive reasoning Progress 92

Table 5.1. Comparison of the performance of the proposed algorithm with that of 2-D AMDF and MP Tracker regarding the criteria GER and SE when TMR is 0dB. 102

Table 5.2. Scale of signal distortion (SIG) 109

List of Figures

Figure 2.1: An example of a pitch tracking with a recording of a single speaker clean speech. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 2.5 seconds long. The plot in the middle corresponds to the periodogram of the original signal using a transform (Fast Fourier Transform). The x-axis in the middle plot represents the time in seconds, and the y-axis corresponds to the frequency. The plot at the bottom represents the pitch track of the audio signal. The pitch track of the speech signal is found using autocorrelation method. 20

Figure 3.1: Multi-pitch determination performance of the 2-D AMDF algorithm. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. 32

Figure 3.2: Multi-pitch determination performance of the 2-D AMDF algorithm in the presence of diffuse noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz..... 33

Figure 3.3: Multi-pitch determination performance of the 2-D AMDF algorithm in the presence of directional noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz..... 34

Figure 3.4: Multi-pitch determination performance of the MP Tracker algorithm in the presence of diffuse noise as background noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying MP Tracker results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. 39

Figure 3.5: Multi-pitch determination performance of the MP Tracker algorithm. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying MP Tracker results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. 40

Figure 3.6: Explanation of the performance of an ideal multi-pitch tracker. First Panel: Waveform of a mixed speech signal containing speech from two male speakers (speaker A and speaker B). The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. Middle Panel: Periodogram of mixture signal containing speech from speaker's A and B where the x-axis represents

the time in seconds, and the y-axis corresponds to the frequency. Third Panel: Pitch track of mixture signal found by (Boersma & Weenink), with red lines representing pitch track of speaker A and blue lines representing pitch track of speaker B..... 42

Figure 3.7: An example of applying 2-D AMDF and MP Tracker to a 2.5-second audio recording where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The ratio of their total energies is roughly 0dB. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted as-is. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. 44

Figure 3.8: Performance of multi-pitch tracking algorithms when the energy difference between two speakers is more than 15dB. The audio recording is 3-second long where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted as-is. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A,

while blue dots represent the pitch track of speaker B. The pitch of the speaker A is varying between 80Hz to 135Hz during the recording while the pitch of the speaker B is between 110Hz and 150Hz during the audio recording. 46

Figure 3.10: Performance of the algorithms in the presence of directional noise synthesized with a 3-second recorded mixture speech signal where two male speakers were talking simultaneously. The difference in total energy between the mixture signal and directional noise is roughly 0dB. The plot on the top represents the ground truth of the speakers estimated by (Boersma & Weenink). The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. 49

Figure 3.11: (a) Histogram of coherence of diffuse dominated T-F units centered at 2 kHz. (b) Histogram of coherence of directional dominated T-F units centered at 2 kHz. The axes are the coherence values and its probability. 53

Figure 3.12: (a) Histogram of coherence of diffuse dominated T-F units centered at 2 kHz. (b) Histogram of coherence of diffuse dominated T-F units centered at 200 Hz. The axes are the coherence values and its probability. 54

Figure 3.13: Comparison of speech dominated and directional noise dominated TF units, and noise source and speech source are placed -90° and 90° respectively. Phase error values are centered around zero for speech dominated TF units, and its value is far from zero (around $\pm\pi$) for noise dominated TF units. 56

Figure 3.14: An example of applying MVDR beamformer to a recording that involves two speakers. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 25 seconds long, and it could be divided into three regions. The first region, starting from 0.5 seconds to 8 seconds, is where the first speaker is speaking alone. The second region, 9 seconds to 15 seconds, is where the second speaker is speaking by himself. The third region, 16 to 25 second, is where the two speakers talking simultaneously. The plot in the middle corresponds to the periodogram of the original signal while the plot at the bottom represents the periodogram of the result of applying the MVDR beamformer to suppress the second speaker. 61

Figure 3.16: Overview of HPS algorithm taken from www. ccrma.stanford.edu 66

Figure 3.17: Spectrogram and the result of HPS analysis for clean speech. 67

Figure 3.18: Spectrogram and the result of HPS analysis for speech contaminated with diffuse 68

Figure 3.19: Spectrogram and the result of HPS analysis for speech contaminated with directional noise. 69

Figure 4.1: Generic Organizational Framework of EAR approach. 75

Figure 4.2: Abductive reasoning process 76

Figure 4.3: Depiction of the flowchart of Early Abductive Reasoning approach. 79

Figure 4.4: Procedure for discrepancy detection 83

Figure 4.5: Examples of discrepancies 85

Figure 4.6: Types of discrepancies. (a) Missing prediction in an edge (b) Missing prediction with converging one pitch track (c) Missing prediction with converging two pitch tracks (d) Fluctuations in a pitch track (e) Fluctuations in two pitch tracks (f) Incorrect speaker assignment (g) Missing consistency in a pitch track (h) Missing consistency in two pitch tracks (j) Missing prediction at two edges (k) Missing prediction (l) Contamination with diffuse noise source (m) Contamination with directional noise source..... 87

Figure 4.7: Possible scenarios..... 91

Figure 4.8: An example of EAR-based System..... 96

Figure 5.1: Omni-directional dual microphones used for recording our data..... 99

Figure 5.2: Placement of the two omnidirectional microphones and sound sources..... 100

Figure 5.3: GER versus TMR for male-male mixture speech signals. The results are averaged over 90 mixture speech signals..... 103

Figure 5.4: GER versus TMR for female-female mixture speech signals. The results are averaged over 90 mixture speech signals..... 104

Figure 5.5: GER versus TMR for male-female mixture speech signals. The results are averaged over 90 mixture speech signals..... 104

Figure 5.6: SE versus TMR for averaged over 270 mixture speech signals..... 105

Figure 5.7: GER and SE versus SNR for averaged over 270 mixture speech signals in a directional noise environment..... 107

Figure 5.9: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals for TMR level of 0dB..... 112

Figure 5.10: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with restaurant noise for SNR level of 0dB. 112

Figure 5.11: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with restaurant noise for SNR level of 5dB. 113

Figure 5.12: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with music noise for SNR level of 0dB..... 113

Figure 5.13: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with music noise for SNR level of 5dB..... 114

List of Abbreviations

ACF…….....	Autocorrelation Function
AI…….....	Artificial Intelligence
AMDF…….....	Average Magnitude Difference Function
BAK……...	Background Intrusiveness
BB…….....	Blackboard
CPSD..……..	Cross power spectral density
EAR…….....	Early Abductive Reasoning
ERB…….....	Equivalent Rectangular Bandwidth
FFT..…….....	Fast Fourier Transform
GER…….....	Gross Error Rate
HPS…….....	Harmonic Product Spectrum
MVDR……	Minimum Variance Distortionless Response
NN…….....	Neural Networks
OVRL……..	Overall Mean Opinion Score
PE..…….....	Phase Error
SE…….....	Separation Error
SIG…….....	Signal Distortion
SNR…….....	Signal-to-Noise Ratio
SVM……...	Support Vector Machines
TMR……...	Target to Masker Ratio

Chapter 1: Introduction

In this thesis, we demonstrate how early abductive reasoning (EAR) can be practically incorporated into signal processing methods to address an important signal separation problem *more effectively than previously known methods*. Abductive reasoning (Josephson and Josephson, 1996) is a reasoning process in which incomplete or partial evidence is used to conjecture causal explanations for what gave rise to that evidence. In many applications, the signal processing is designed to produce signal representations that ensure the computational practicality and accuracy of subsequent abductive reasoning by humans or machines. However, there is a line of research (Cole, 2011), (Cole et al., 2010), (Cole et al., 2011), (De Luca et al., 2006) suggesting that signal processing applications that require the generation of dynamically changing explanations for the input signal data may potentially benefit from the incorporation of abductive reasoning capabilities to drive a data-adaptive process for selecting the most appropriate signal processing transformation at any given time. In the context of our research, we refer to this type of process as early abductive reasoning; the "early" here refers to the fact that the reasoning process is utilized before the signal processing transformation is completed.

In this dissertation, we demonstrate that explicit and systematic incorporation of early abductive reasoning capabilities into algorithms for *blind* signal separation (Schwarz et al., 2012), (Even et al., 2008), (Mukai et al., 2006) can yield significant

performance improvements over the current state of the art. The early abductive reasoning mechanisms formulated and studied in this dissertation are applicable to the output data of signal processing modules in order to conjecture the structure of time-frequency (Even et al., 2008), (Mukai et al., 2006), (Oppenheim and Nawab, 1997) interactions between the signal components that are to be separated. The conjectured interactions may then be used to drive subsequent signal separation processes that are as a result *less blind* to the interacting signal components and, therefore, more effective.

For the purposes of demonstrating the practical use of early abductive reasoning in this thesis, we decided to concentrate on an application domain that involves multi-speaker pitch tracking in everyday noisy environments. We selected this domain because the interactions of speech with other speech signals and everyday sounds are best viewed in the time-frequency domain using tools such as the short-time Fourier transform (Nawab and Quatieri, 1988). Consequently, there is the potential of an artificial intelligence program to use abductive reasoning to "uncover" the interactions between signal components in the time-frequency domain despite the unpredictable dynamics of the environment. These unpredictable dynamics arise because the artificial intelligence program has no *a-priori* information of when each speaker is talking or what he/she is saying and, consequently, how the time-frequency components of their voices may be interacting with each other and with the time-frequency components of other sounds that may be in the background.

The EAR-based algorithm we have developed as a result of the research conducted for this dissertation is superior to previously published algorithms (Ba et al.,

2012), (Ziolko et al., 2009). The algorithm development and evaluation for previously published algorithms took place for signals corresponding to two speakers' speaking simultaneously in noise-free environments. In contrast, our EAR-based algorithm has been developed to track the pitch of individuals in the presence of unstructured audio environments such as restaurants, cafeterias, and street corners. To observe the pitch tracking performance of the previously reported algorithms in noisy environments, we implemented these algorithms and tested them over a database of two-speaker signals contaminated with noise. The results show that the EAR-based algorithm *significantly* outperforms previous multi-speaker pitch algorithms in the presence of noise activity in the audio environment.

It should be noted that while other investigators have previously investigated two-speaker pitch tracking in noisy environments, their investigations have been limited in one of two ways. Their algorithms have either been developed for single-microphone data (as opposed to dual-microphone data in our case) or they have worked only with restricted classes of noise such as white or pink noise. Representative of the one-microphone category is the work of Wu et al. (Wu et al., 2003) and Jin et al. (Jin and Wang, 2010). They used HMM-Based Multi-pitch tracking in realistic noisy environments. While their algorithms are able to separate a single pitch track from unstructured noise, they are only able to separate two speech tracks from each other only if no noise is present. Wu et al., (Wu et al., 2003) used pitch period statistics from selected channels for Hidden Markov Model (HMM) in order to generate pitch tracks. Jin et al. (Jin and Wang, 2010) used correlogram and cross-channel correlation features for

HMM-based pitch tracking. Wohlmayr et al. (Wohlmayr et al., 2011) and Lin et al. (Lin et al., 2014) have developed a multi-pitch algorithm for single-microphone data, which is synthetically mixed two speech signals (no noise). Wohlmayr et al., (Wohlmayr et al., 2011) used probabilistic models using factorial HMM (FHMM) in obtaining pitch tracks. Lin et al., (Lin et al., 2014) used correlogram and continuous correlation features followed by Deep Belief Network (DBNs)/HMM based pitch track estimation. Abhijit et al. (Abhijith et al., 2014) have developed a multi-pitch tracking algorithm for single-microphone data using time-varying Gaussian Mixture Model, which is able to compute two pitch tracks from a mixture of two speech signals. Representative of the multi-microphone category is the work of Gerlach et al. (Gerlach et al., 2014) in which they developed an algorithm for joint estimation of pitch and direction of arrival. However, the algorithm was evaluated only with additive pink noise.

We can see that previous multi-pitch tracking algorithms have focused on developing an algorithm for single-microphone data rather than dual-microphone data. However, their performance is reliable in the case of a mixture of two speech signals or a mixture of one speech signal and the unstructured noise signal. On the other hand, multi-speaker pitch track algorithms developed for dual microphone data are able to separate two speech tracks from each other only if white or pink noise is present. We have developed a new algorithm that estimates individual pitch tracks of two simultaneously speaking speakers in the presence of unstructured noise by using dual-microphone data.

This chapter begins in Sec. 1.1 with a background on the multi-speaker pitch-tracking problem and continues with an overview of previous state-of-the-art algorithms

that have been developed to address it. Next, in Sec. 1.2, we describe our overall database construction for two-speaker speech signals and two evaluation methodologies that we have used to test multi-speaker pitch-tracking solutions. The contributions of this thesis are outlined in Sec. 1.3, and a general overview of the remaining chapters follows in Sec. 1.4.

1.1 Multi-Speaker Pitch-Tracking Problem

Pitch tracking is a fundamental problem in speech signal processing. A reliable pitch-tracking algorithm is critical for applications such as speech enhancement (Loizou, 2007), (Ming et al., 2010), (Hu, 2008), speaker recognition (Maurya and Aggarwal, 2016), (Jokic et al., 2015) and speaker identification (Chakroun et al., 2015), (Tazi, 2016), especially in noisy environments. However, due to the difficulty of working with noisy files and interference of the other speakers, designing a reliable pitch-tracking algorithm is very challenging. Most of the existing pitch-tracking algorithms (Ding et al., 2006), (Radfar et al., 2011), (Vishnubhotla and Epsy-Wilson, 2008), (de Cheveigné and Kawahara, 2002) are limited to clean (or modestly noisy) single-speaker speech.

Many pitch-tracking algorithms have been specifically designed for detecting a single pitch track with voiced/unvoiced decisions in noisy speech. The majority of these algorithms have been tested on clean speech and speech mixed with different levels of white noise or on synthetic noisy speech data in a laboratory environment (Ba et al., 2012). One of the widely-used time domain methods for single-speaker pitch detection is based on the autocorrelation function (ACF) (de Cheveigné and Kawahara, 2002). A number of algorithms (McLeod and Wyvill, 2005), (Hess, 1983), (Schroeder, 1968) have

been developed based on this approach. Average Magnitude Difference Function (AMDF) (Ross et al., 1974) is a variation of ACF, which calculates a formed difference signal between the delayed signal and the original one. Reference (de Cheveigné and Kawahara, 2002) uses a novel difference function similar to autocorrelation to search for the period. It further refines the detection result using some post-processing methods.

In the frequency domain, the pitch is often found by using methods (de Cheveigne, 1993), (Ding et al., 2006) that searching for harmonic peaks in the power spectrum (Oppenheim and Nawab, 1997). A third approach to single-speaker pitch detection is the cepstrum method (Rabiner and Schafer, 2010), (Noll, 1967). The cepstrum is found by computing the inverse Fourier transform of the log-magnitude Fourier spectrum, which captures the period in the speech harmonics, and thus shows a peak corresponding to the period in frequency.

A pitch tracker should perform robustly in a variety of acoustic environments. However, for backgrounds containing harmonic structures such as background music or voiced speech, more than one pitch is present in various time frames. For that reason, a multi-speaker pitch tracker is required that can yield multiple pitches at each frame and can separate background noise from the speech.

Current multi-speaker pitch-tracking algorithms do not produce promising results in unstructured environments where the speech may be dynamically contaminated by the unpredictable appearance of multiple noise sources at different times. For instance, while tracking the pitch of two people talking to each other in a restaurant, the pitch tracking process may have to deal with other people in the restaurant speaking loudly and also at

the same time there might be music and other sound-generating activities in the background. These kinds of environments make the task of multi-speaker pitch estimation and tracking difficult.

Vishnubhotla and Epsy-Wilson (Vishnubhotla and Epsy-Wilson, 2008) have addressed the multi-speaker pitch extraction problem in certain situations. They have proposed, implemented, and tested a 2-D AMDF algorithm for mono-channel speech separation. Bokhoven and Van (Bokhoven & Van, 1991) and Chazan et al. (Chazan et al., 1993) have proposed algorithms for detecting up to two pitch periods for single-microphone signal speech separation by suppressing the harmonic frequencies of one of the speakers to obtain the pitch frequencies of the other speaker. In a recent model proposed by Radfar et al. (Radfar et al., 2011), the pitch frequencies are estimated by introducing a novel spectral distortion optimization that takes into account the sinusoidal modeling of the speech signal. However, it should be noted that all of these multi-speaker pitch-trackers were designed for and tested on mixtures of speech-only signals with no background noise. The problem of multi-speaker pitch tracking in realistic noisy environments has largely not been addressed.

In our research, we aimed to develop and implement a computational approach that uses a combination of existing speech enhancement, speech separation, and pitch detection techniques in conjunction with *early abductive reasoning* to obtain multiple pitch tracks in unstructured noisy environments. Since early abductive reasoning may be viewed as a data-adaptive process for selecting the most appropriate signal processing for blind signal separation, we decided to utilize well-known Artificial Intelligence (Lesser et

al., 1995), (Mani, 1998) techniques for implementing early abductive reasoning. Specifically, we decided to utilize the so-called Blackboard architecture (Nawab et al., 1992), (Lesser et al., 1995) from Artificial Intelligence (AI) to implement the required abductive reasoning processes.

We have used the Blackboard architecture to organize and implement the integration of the early abductive reasoning approach with existing traditional signal processing techniques. Noise suppression techniques such as coherence based filtering (Abdipour et al., 2014), minimum variance distortionless response (MVDR) (Pan et al., 2014), cross-correlation subtraction (Ziolko et al., 2009), and phase based filtering (Abdipour et al., 2014) were used to separate the desired speech signal at some level from the background noise. For the actual pitch tracking, we used techniques such as the 2-D average magnitude difference function (AMDF) (Vishnubhotla and Epsy-Wilson, 2008), and the harmonic product spectrum (HPS) (Ding et al., 2006) and others as necessary. Our objective was to combine all of these techniques together and combine them with the use of early abductive reasoning to build a system that can work in an unstructured audio environment.

1.2 Evaluation of Multi-Speaker Pitch Tracking

An important aspect of our research is the evaluation of how well our EAR-based algorithm performs. We have done this through two separate methodologies:

- Error-based Evaluation Methodology
- Enhancement-based Evaluation Methodology.

In both evaluation methodologies, we utilized a database of 570 mixture signals created from 20 clean speech files (10 male and 10 female) from the TIMIT database (Garofolo et al., 1993) and directional and diffuse noise files recorded in various restaurant and street scenes. To create dual-speaker speech files, we placed two speech sources in the various combinations of the directions corresponding to 0°, $\pm 30°, \pm 45°, \pm 75°$, and $\pm 90°$, where 0° is perpendicular to the line which combines the two microphones. For each direction, the two signals received at the microphones placed approximately 5cm apart were saved as the corpus of two-speaker files. Figure 1.1 depicts the microphones set up for the recording scenarios.

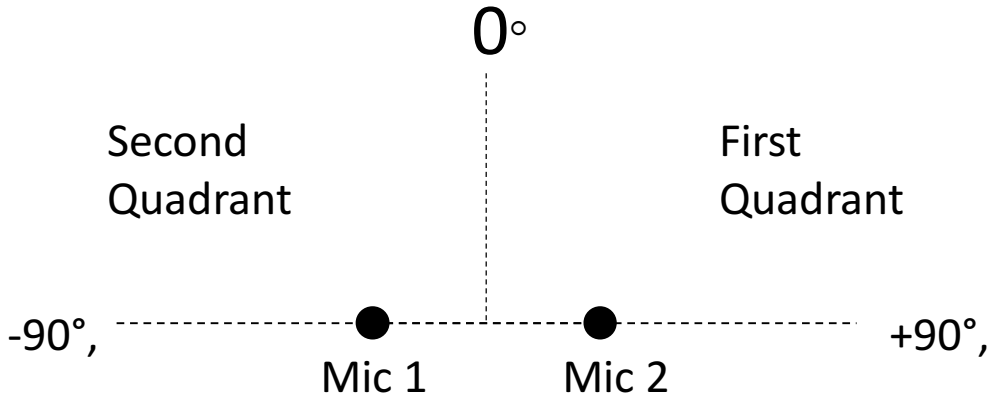


Figure 1.1: Placement of the two omnidirectional microphones and sound sources.

Similarly, we placed directional music noise sources in the directions stipulated above to make the corpus of directional noise files. In addition, to make a corpus of the diffuse noise files, we made recordings in restaurant and street scenes. Finally, we synthesized the simultaneous two speaker files with directional noise files and diffuse

noise files. It should be noted that we synthesized simultaneous two speaker files with the noise files where sources are placed in different directions.

1.2.1 Error-Based Evaluation Methodology

One of the most difficult problems in comparing and evaluating the performance of multi-speaker pitch trackers is choosing a meaningful performance criterion. Voiced sounds are a combination of a fundamental frequency with a set of harmonics that occur at the integer multiples of that fundamental frequency. A human can interpret the pitch (or fundamental frequency), even if it is absent in the sound. We may quantify pitch as a fundamental frequency, but the fundamental frequency is a physical value while the pitch is an auditory percept of a sound. The pitch is a subjective attribute of a sound that humans perceive based on the fundamental frequency of the sound and changes in that frequency. Pitch carries conversational cues in human speech. These cues also play a role in allowing us to identify a speaker consistently. For most purposes, it can be assumed that the pitch and fundamental frequency of speech sounds correspond to each other. The definition of the pitch by Terhardt (Terhardt, 1979) provides a good way to combine the temporal properties of the stimulus with its perceived pitch. He states that "*The extraction of the fundamental frequency is in some respect equivalent to the extraction of the virtual pitch. In a strict sense, however, the frequency which corresponds to virtual pitch, and the fundamental frequency defined as the largest common divisor of the partials) are in general not identical. Hence in the analysis of auditory signals such as speech and music actually the extraction of the fundamental frequency is not the real aim but rather extraction of the frequency which corresponds to the virtual pitch*".

Although there are some distinctions between pitch and fundamental frequency, they were considered equivalent for the purpose of the objective evaluation criteria used in our error-based evaluation method. In this evaluation method, two criteria, namely gross error rate (GER) defined in (Rabiner et al., 1976), and separation error (SE), defined in (Radfar et al., 2011), were used. The GER and SE are defined as follows:

$$GER = \frac{\#\ pitch\ values\ that\ vary\ by\ more\ than\ 10\ Hz\ from\ true\ pitch}{\#\ pitch\ values} * 100 \tag{1.1}$$

$$SE = \frac{\#\ pitch\ values\ are\ mistakenly\ declared\ for\ a\ speaker}{\#\ pitch\ values} * 100 \tag{1.2}$$

The ground truth pitch tracks of a speech mixture were obtained by computing pitch trajectories of the individual speech signals (without noise) using Praat (Boersma & Weenink, 2016). We compared the performance of our system to that of two recent multi-speaker pitch-tracking techniques: (1) 2D-AMDF technique proposed by Vishnubhotla and Epsy-Wilson (Vishnubhotla and Epsy-Wilson, 2008) and (2) the one proposed by Radfar et al. (Radfar et al., 2011) called MP Tracker. We evaluated these three methods using GER and SE to determine the efficacy of our EAR-based algorithm with respect to the other two methods.

Since the speech signal is pseudo-periodic, gross errors can arise when there is a strong first harmonic, which results in its amplitude becoming significant or greater than that of the fundamental harmonic. This can lead to what are known as "doubling" errors

because this leads to a significant second peak in each period, which time-domain algorithms sometimes confuse with the main peak but humans can perceive the true pitch of the sound. For this reason, objective assessment criteria may not be enough to be dependable for evaluation of multi-speaker pitch tracking.

1.2.2 Enhancement-Based Evaluation Methodology

One very basic question arose from this earlier investigation. This is the question as to how, and in what manner, the results of the error analysis used in the objective evaluation of the pitch detectors are related to perceptual criteria of quality in a subjective evaluation of the pitch detectors. Such a subjective evaluation of pitch detectors can be obtained by assessing the quality of speech after harmonic filtering using pitch tracks obtained from multi-speaker pitch trackers. Speech is a non-stationary signal. That is to say, its characteristics change rapidly as a function of time. For instance, the usage of improper analysis window size for the T-F domain will result in inadequate algorithm performance despite low GER and SE rates. On the other hand, the human auditory system is more sensitive to changes in absolute frequency at low frequencies, and the pitch tracking algorithms may not be able to respond rapidly enough to changes in the speech signal. For this reason, we also decided to use a subjective criterion in order to assess the multi-speaker pitch tracker algorithms as well as the objective error-based criteria.

In the enhancement-based evaluation method, we used each extracted pitch track to enhance the speech of the corresponding speaker. The enhanced speech was then evaluated through listening tests to compare the quality of the enhanced speech to the speech in the original recordings. The enhancement was done by using the harmonic

filtering technique (Jin et al., 2009), (Jackson and Shadle, 2000). The basic idea is to suppress the frequency components of the noise signal that belong to the interference speech while preserving the fundamental frequency and its harmonics of the target speech. Enhanced speech was then evaluated through a subjective test to compare the quality of the enhanced speech to the speech in the original recordings by a set of 20 listeners.

Subjective tests were designed according to ITU-T Recommendation P.835 methodology (P.835, 2003) intended to evaluate the speech quality along three components: signal distortion, noise suppression, and overall quality. This method (P.835, 2003) instructs the listener to rate the speech along three different axes on the scales described below:

1) The speech signal alone is rated using a five-point scale of signal distortion (SIG)

5 - Very natural, no degradation
4- Fairly natural, little degradation
3-Somewhat natural, somewhat degraded
2-Fairly unnatural, fairly degraded
1- Very unnatural, very degraded

Table 1.1. Scale of signal distortion (SIG)

2) The background noise suppression alone is rated using a five-point scale of background intrusiveness (BAK)

5 - Not noticeable
4- Somewhat noticeable
3- Noticeable but not intrusive
2- Fairly conspicuous, somewhat intrusive
1- Very conspicuous, very intrusive

Table 1.2. Scale of background intrusiveness (BAK)

(3) The overall quality of the audio experience is rated using the scale of the Mean Opinion Score (OVRL) – [1 = bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent].

1.3 Contributions of Thesis

The results of the research presented in this thesis demonstrate the power of combining early abductive reasoning and advanced signal processing methods to address an important multi-speaker pitch-tracking problem. We have successfully integrated multi-speaker pitch tracking algorithms into early abductive reasoning based system; this integration allowed us to overcome the complexities of multi-speaker pitch tracking problem that arise from unstructured signal environment. One of the main contributions of this thesis is the development and establishment of an algorithm for multi-speaker pitch tracking in noisy environment. This is important because previously published feature-based multi-speaker pitch tracking algorithms are not as robust to noisy environments as our proposed algorithm. In contrast to model-based multi-speaker pitch

tracking algorithms, our proposed algorithm is independent of *a priori* information about speakers and environment.

Another contribution of this thesis is the use of Blackboard (BB) architecture from Artificial Intelligence (AI) to incorporate Early Abductive Reasoning (EAR) into multi-speaker pitch tracking. Blackboard system is used to decompose of EMG signals and recognize of movement disorders that involve real-world signal environments. The earliest applications of Blackboard systems to the analysis of sound signals (Lesser et al., 1995) and music signals (Mani and Nawab, 1999) were limited to synthetic signals because at that time technology was not mature enough to deal with the complexities of real-world conditions. Beyond the contributions of this thesis mentioned above, this is the first use of the Blackboard-based system tested on the real-world audio signal.

In addition, we have implemented a new algorithm that shows a significant performance improvement in terms of quantitative and qualitative evaluation in comparison to current multi-speaker pitch tracking algorithms. Experiments show that the previous algorithms yield high error rates in some regions, especially where some of their design constraints are violated. The errors of this type are called discrepancies and are described in detail in Chapter 4. Our EAR-based approach successfully handles these discrepancies and yields higher performance in comparison to the previous algorithms.

1.4 Thesis Outline

We begin by first giving a description of pitch and pitch tracking. We continue with formulating single-speaker and multi-speaker pitch tracking problem in general form and

we describe the other closely related works in the literature in Chapter 2. In Chapter 3, the challenges and constraints of current multi-speaker pitch tracking methods are discussed and explained in detail. Chapter 4 presents our improved algorithm using the early abductive reasoning approach. Finally, the chapter briefly explores the speaker assignment problem. Chapter 5 describes the evaluation of our proposed approach on several tasks including objective and subjective tests, as well as comparison with other algorithms proposed in the literature. The thesis is concluded in Chapter 6 with a discussion of potential directions of future work.

Chapter 2: Pitch Tracking Background

The pitch of a sound is crucial in many contexts such as phonetics (Baer, 1979), speech separation (Wiem et al., 2016), and speech coding (O'Shaughnessy, 2000). In one of the speech coding algorithms called voice vocoders, the analyzer is used to estimate and transmit the pitch values that represent the original signal. Speech is synthesized using these pitch values. The quality of the synthesized speech signal highly depends on the accurate estimation of the pitch values (O'Shaughnessy, 2000). Pitch estimation is also used in speech separation techniques in order to segregate the source from the mixture signal using spectral information such as trajectory and harmonic structure of target source (Lee et al.,, 2008). Pitch tracking is useful for musical analysis as well as for speech analysis. For example, pitch extraction can be used for melody extraction. In reference (Rao et al., 2008), pitch tracking is used to search for a song from a database of thousands of songs by only using melody. In addition, pitch tracking can be used as a helper to make visual feedback tools for musical performers.

In this thesis, our focus is on pitch tracking for speech. Although pitch tracking is used in many research areas, it is not a trivial task. Many pitch-tracking algorithms have been developed, and many of them work well in a specific context because it has been difficult to develop a pitch tracker that works well in all contexts. For example, a pitch tracker developed for a particular application, such as musical note detection or speech analysis, may depend on the domain of the data. A pitch tracker for music analysis is less accurate when applied to speech analysis. Furthermore, a tracker for clean speech does

not perform well on speech contaminated with noise. The result is that there are many pitch tracking algorithms currently on research, but few that are appropriate to more than one context.

2.1 Single-Speaker Pitch Tracking

The simplest pitch-tracking problem is when there is only one speaker and no background noise. The topic of pitch tracking for single-speaker speech has been well explored, and there are several developed pitch tracking methods (Ding et al., 2006), (Hess, 1983), (de Cheveigné and Kawahara, 2002) in the literature based on mathematical principles. Single-speaker pitch trackers can be divided into two general categories: Time-domain and frequency-domain. All pitch trackers have their advantages and disadvantages. In general, time domain methods are usually computationally simple compared to frequency domain methods. All single-speaker pitch-tracking methods rely on processing small portions of a signal to produce pitch values. This process is called windowing. The window used is typically 20–50 ms duration. Although shorter windows give higher time resolution, they compromise resolution in the frequency domain. Figure 2.1 represents an example of pitch tracking performed on a non-noisy recording of single-speaker speech. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 2.5 seconds long. The plot in the middle corresponds to the periodogram of the original signal using the short-time Fourier Transform. The x-axis in the middle plot represents the time in seconds, and the

y-axis corresponds to the frequency. The plot at the bottom represents the pitch track of the audio signal. In this example, the pitch track of the speech signal is found using the autocorrelation method described in (Hess, 1983). The periodogram represents the energy content of the speech as a function of frequency and time. The horizontal stripes show the harmonic content of the speech. As seen from the figure, the harmonics of the periodogram are correlated with the pitch track of the speaker and they bend as the pitch changes. This figure shows the evidence that the pitch is available in the periodogram and it carries the information about the frequency content of the speech.

Some time-domain pitch tracking methods search how often the waveform fully repeats itself by using features such as zero crossing rate, peak rate etc. The main idea for these methods is that if a signal is periodic, then there are time-repeating events that can be counted, and the number of these events that happen in a second is inversely related to the fundamental frequency or pitch. If there is a specific time event that is known to exist once per period in the waveform, it may be identified with time-domain pitch tracking methods.

There are some positive aspects of time-event rate detection algorithms. These methods are exceedingly simple to understand and implement, and they take very little computing power to execute. Speech can largely be considered a combination of frequency and amplitude modulated sinusoidal waves and thus is best viewed as a quasi-periodic signal. This means that it is hard to pin down periodic events in the speech signal that can be used to reliably arrive at pitch estimates over short periods of time.

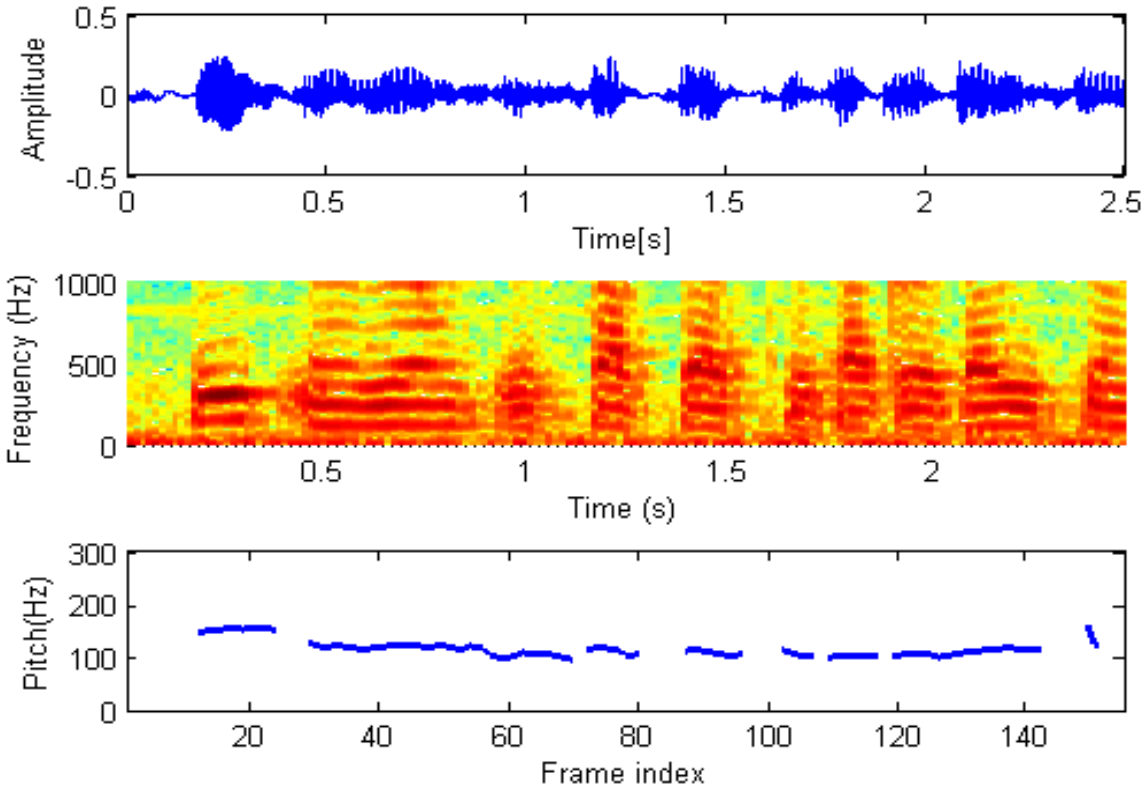


Figure 2.1: An example of a pitch tracking with a recording of a single speaker clean speech. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 2.5 seconds long. The plot in the middle corresponds to the periodogram of the original signal using a transform (Fast Fourier Transform). The x-axis in the middle plot represents the time in seconds, and the y-axis corresponds to the frequency. The plot at the bottom represents the pitch track of the audio signal. The pitch track of the speech signal is found using autocorrelation method.

There is significant information in the frequency domain that can be conveniently related to the fundamental frequency of a sound. Speech signals are composed of the fundamental frequency and its harmonics at integer multiples of the fundamental frequency. The frequency-domain methods use a transform (usually the Fast Fourier Transform, FFT) to break the signal down into its frequency components, yielding information about its amplitude vs. frequency. They then analyze the periodogram to

determine the fundamental frequency. In the frequency domain, the pitch can be found by searching for harmonic peaks in the power spectrum.

The cepstrum method (Rabiner and Schafer, 2010), (Noll, 1967) is another popular method for extracting the fundamental frequency. The cepstrum is found by computing the inverse Fourier transform of the log-magnitude Fourier spectrum, which captures the period in the speech harmonics, and thus shows a peak corresponding to the period in frequency. Another frequency-domain method is a filter-based method. The basic idea behind the filter-based methods is to try different bandpass filters with different center frequencies and comparing their output. When a spectral peak lines up with the passband of a filter, the result is a higher value in the output of the filter than when the passband does not line up (Lane, 1990), (Moorer, 1977).

However, in situations where speech is corrupted by noise, the performance of single-speaker pitch trackers degrades drastically, which makes the estimated pitch uninformative for speech applications. To overcome this situation, generally a statistical approach is used for pitch trackers. A statistical approach maintains multiple hypotheses with different probabilities to make the pitch trackers more robust to the acoustic noise (Liu and Wang, 2016), (Suk and Ellis, 2012). Although many recent studies tried to address the noise-robustness issue for single-speaker pitch tracking, it is still a challenging problem in the presence of background noise.

2.2 Multi-Speaker Pitch Tracking

When more than one speaker is involved, pitch tracking becomes a much more difficult problem (Radfar et al., 2011), (Vishnubhotla and Epsy-Wilson, 2008). In the literature, most of the pitch-tracking algorithms have been specifically designed for detecting a single pitch track with voiced/unvoiced decisions in noisy speech. However, due to the difficulty of working with noisy files and interference of the other speakers, designing a reliable multi-speaker pitch-tracking algorithm is very challenging. The problem of multi-speaker pitch tracking began to be addressed in the early 1980s, and a number of algorithms relying on various acoustic features have been used. As in single-speaker pitch tracking algorithms, multi-speaker pitch tracking algorithms can be classified into the statistical approach-based and feature-based algorithms. The feature-based approaches rely on certain properties of the signal in different domains to estimate the pitch values. Feature-based approaches include the autocorrelation, the average magnitude difference function and, the spectrum of the signal, etc. The earliest known 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 (Hess, 1983). In particular, since male and female speakers have different ranges of pitch, the optimal frequency resolution

required for accurate pitch estimation are significantly different. This makes it difficult to come up with a good set of parameters for the window length and shape that could yield robust estimates for both of speakers, especially when both genders occur simultaneously in the same speech mixture. The autocorrelation function is another measure used for pitch tracking algorithms. The autocorrelation function (ACF) of a signal compares a signal to a delayed version of itself, by multiplying the two versions together. This function of delay or lag shows a maximum value 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 harmonics. This property has been used to develop algorithms that calculate the autocorrelation of the speech signal and then assign the peak as the pitch of the input signal. Various improvements and variations, like the enhanced ACF (Ross, 1974) 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 (de Cheveigne, 1993). After the cancellation of the harmonics of the first speaker, the second pitch estimate is obtained using the ACF of the remaining signal. The issue with this approach is that harmonic cancellation does not always effectively cancel the effects of the dominant pitch. Further, the harmonic cancellation procedure is highly susceptible to the formant structure of both the speakers. In particular, in regions of speech where the first

formant, is close to the pitch of the dominant speaker, it is difficult to filter out the effects of the first formant, which often results in an error for second speaker pitch estimation.

The most successful approach towards multi-pitch detection is the so-called Spectro-temporal approach (Wu et al., 2003), 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. The issue of this Spectro-temporal approach to multiple pitch detection is that the second peak of the ACF 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 channels bands.

Another approach is machine learning, which is either to 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 first case, 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 (Weiss and Ellis, 2007). As one example of the second case, 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 (Kameoka et al., 2007). 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. Then, the pitch values are estimated from harmonic information. 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.

2.3 Chapter Summary

In this chapter, we explored the pitch-tracking problem. In Sec 2.1 we focused on the problem of single-speaker pitch tracking and we explained time-domain and frequency domain algorithms that are used in the literature for single-speaker pitch tracking problems. In Sec 2.2, we focused on the multi-speaker pitch tracking algorithms. In this section, we explained the various algorithms that address various aspects of the problem of multi-speaker pitch tracking problem.

Chapter 3: Multi-Speaker Pitch Tracking Challenges

In this thesis, we are proposing a new approach for estimating the pitch of two participating speakers speaking simultaneously in an unstructured audio environment. Our motivation comes from speech processing applications focused on the problem of analyzing the speech signal and enhancing the speech of a particular speaker. For example, in a recording of a meeting consisting of more than two speakers, it may be necessary to perform speech recognition, and also authenticate the specific speaker among other speakers. 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 use further processing to follow each speech signal. Furthermore, many real-world situations consist of speech signals, which are usually contaminated with various forms of noise, like noise from multiple sources as well as background speech from other speakers. For this reason, the task becomes more difficult, as there is also the need to enhance the quality of the speech by removing the background noise.

A lot of the techniques that fall under the feature-based category rely on the pitch of the individual source to perform speech separation and enhancement. Most of the multi-speaker pitch-tracking approaches have been formulated to perform in the non-noisy environment (i.e., assuming only two speakers are speaking simultaneously). The most important drawback of all these multi-speaker pitch-tracking approaches is that in practice it is unusual to be able to guarantee that there will be no more than two speakers.

However, information relevant to the number of speakers is rarely known beforehand and even if so, it becomes impractical to use one microphone for tracking the pitch of an individual speaker in the presence of more than one noise source. It becomes necessary to use at least two microphones to track the speaker's pitch in a noisy environment.

In this thesis, we address the task of multi-speaker pitch tracking along with speech enhancement, i.e., the removal of background noise in the speech signal. The problem description is as follows: There are two microphone signals available to the algorithm. There are two speakers whose pitch we want to track in a mixture of speech and background. We limit ourselves to tracking the pitch of individual speakers from a mixture containing a maximum of two simultaneous speakers and solve the case of two speakers speaking simultaneously in the presence of unstructured background noise.

We begin in this chapter by first exploring two multi-speaker pitch tracking algorithms 2-D AMDF and MP Tracker that are capable of identifying the presence of two simultaneous speakers and yield their pitch estimates (Sec 3.1). We then explain their implementations and present some experimental results of these algorithms tested with some mixture signals from our database. In Sec 3.2, we show their challenges under various noise conditions and highlight some of their limitations to motivate the need for the algorithm developed in this thesis. In Sec 3.3, we provide a detailed description of speech enhancement techniques that we took advantage of the in conjunction with *early abductive reasoning*. Also, in Sec 3.3, we explain their implementation and we present some relevant experimental results for these algorithms.

3.1 Multi- Speaker Pitch Tracking Techniques

3.1.1 2-D Average Magnitude Difference (2-D AMDF)

Various techniques can be used for multi-speaker pitch tracking, but most of them are only able to track pitches accurately in the voiced region. In reference (Vishnubhotla and Epsy-Wilson, 2008), a new multi-speaker pitch tracking method was proposed that estimates the pitch of each speaker and identifies the number of speakers in voiced/unvoiced regions. In this paper, 2-D Average Magnitude Difference Function (AMDF) is proposed for estimating the pitch as well as the number of speakers.

For multi-speaker pitch tracking, spectral methods are also used, but parameter selection is crucial (Schroeder, 1968). For instance, any change in window shape and size can cause inaccurate pitch estimation. Autocorrelation function (ACF) and 1-D AMDF are used for multi-speaker pitch tracking, but it has been proven that these methods suffer from inter-harmonic problem (Vishnubhotla and Epsy-Wilson, 2008). In reference (Vishnubhotla and Epsy-Wilson, 2008), Vishnubhotla and Epsy-Wilson claim that 2-D AMDF can cope with the inter-harmonic problem and estimate the pitches of two speakers. First, one of the input signals is fed into a 60 channels gamma-tone filter bank with center frequencies based on ERB scale (equivalent rectangular bandwidth), and we divide each channel signal into frames in time. Each frame is declared as either a silent frame or a non-silent frame based on its energy levels; silent frames are not to be used for the next stage of the method (Deshmukh et al., 2005). Then 2-D AMDF for each channel frame is calculated as:

$$AMDF[l_1, l_2] = \sum_{n}^{N} x[n] - x[n - l_1] - x[n - l_2] + x[n - l_1 - l_2] \qquad (29)$$

where l_1 and l_2 are lag parameters in a channel frame, N is the frame length. The AMDF value will fall to zero only when l_1 and l_2 are equal to the periods of the signals. Periods can be estimated by finding the point where the 2-dimensional AMDF is zero. This point is called dip point. In practice, potential dip points are found by searching for local minima of AMDF values. After local minima are found, four nearest local maxima around each local minimum are identified. The AMDF values at these four maxima are then used to interpolate the value of the AMDF at the location of the minimum. The difference between the interpolated value and the real value of the local minimum is called the strength of the dip point. The indices of the dip point with the highest strength value correspond to the periods of the speakers at that frame. It should be pointed out that for each channel frame; the AMDF is obtained by summing AMDFs of all the channels that correspond to frequencies below the current channel. After defining the dip point of each channel, the indices of all dip points are plotted on a histogram. Each value in the histogram represents the frequency-of-occurrence of a particular dip index. Then the values in the histogram are normalized such that the sum of the values is equal to one. By setting a threshold on the normalized frequency-of-occurrence, we can find the indices that exceed the threshold. If we obtain only one index above the threshold, we interpret this as representing the fact that there is only one speaker present and the corresponding pitch can be calculated from the index. Similarity, if there are two indices after thresholding, we say there are two speakers, and the indices can be used to calculate the

respective pitches. If there is no index with a value higher than the threshold, we say there is no speaker present in that frame. The authors empirically recommended a threshold of 0.2 and the same threshold value is used in our implementation. Vishnubhotla and Epsy-Wilson (Vishnubhotla and Epsy-Wilson, 2008) pointed that there are two situations that 2-D AMDF algorithm does not work properly. The first situation is that as the pitch difference gets less than 8Hz, the accuracy begins to degrade. The second situation is that as the relative loudness of one speaker begins to exceed the other speaker by 10dB, the accuracy also degrades.

3.1.1.1 Implementation and Experimental Results

We implemented and tested the AMDF algorithm with one microphone signal recording of two simultaneous speakers without background noise. We implemented AMDF technique as described above. We applied 32 ms Hanning window with an overlapping factor of 50% to divide the entire signal into frames. In the experiment stage, we observed that 2-D AMDF method fails in two situations. The first case is when the energy ratio of two speaker signals at a frame is more than 10dB; the algorithm can estimate only the pitch value of the speaker who has the higher energy. The second case is that when the pitch values of the speakers are less than 10 Hz apart. Besides these two situations, when in the absence of background noise, the 2-D AMDF method is able to find the number of speakers in voiced regions and estimate the pitches in these regions. Figure 3.1 shows an example of applying 2-D AMDF to a 1.5-second audio recording where two male speakers were talking simultaneously. The ratio of their energies is roughly 0 dB. The plot on the top represents the ground truth of the first speaker's pitch

while the plot in the middle corresponds to the ground truth of the second speaker's pitch. The plot at the bottom shows the pitches estimated by 2-D AMDF. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The first two plots correspond to reference pitches of the two speakers individually. The pitch of the first speaker is varying between 100Hz to 150Hz during the recording while the pitch of the second speaker is between 100Hz and 120Hz during the audio recording. The plot at the bottom shows the result of 2-D AMDF where both pitches are being tracked. As can be seen from the frames between 10 and 15, the pitches of the two speakers are very close. In this case, the 2-D AMDF mostly gives one pitch value since AMDF estimates that there is one speaker. However, in the frames between 50 and 70, the pitches are far away from each other, and 2-D AMDF gives accurate pitch estimates.

We also tested the efficacy of the 2-D AMDF algorithm in the presence of directional noise and diffuse noise environments. Figure 3.2 presents the performance of the 2-D AMDF in the presence of restaurant ambient noise synthesized with 3 second recorded speech signals where two male speakers were talking simultaneously. The energy ratio between the speech signals and directional noise is roughly 0dB. The plot on the top represents the ground truth of the first speaker's pitch while the plot in the middle corresponds to the ground truth of the second speaker's pitch. The plot at the bottom shows the pitches estimated by 2-D AMDF. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The first two plots correspond to reference pitches of the two speakers individually. As can be seen from Figure 3.2, 2-D AMDF could still estimate the pitch of the speakers accurately in the presence of diffuse noise.

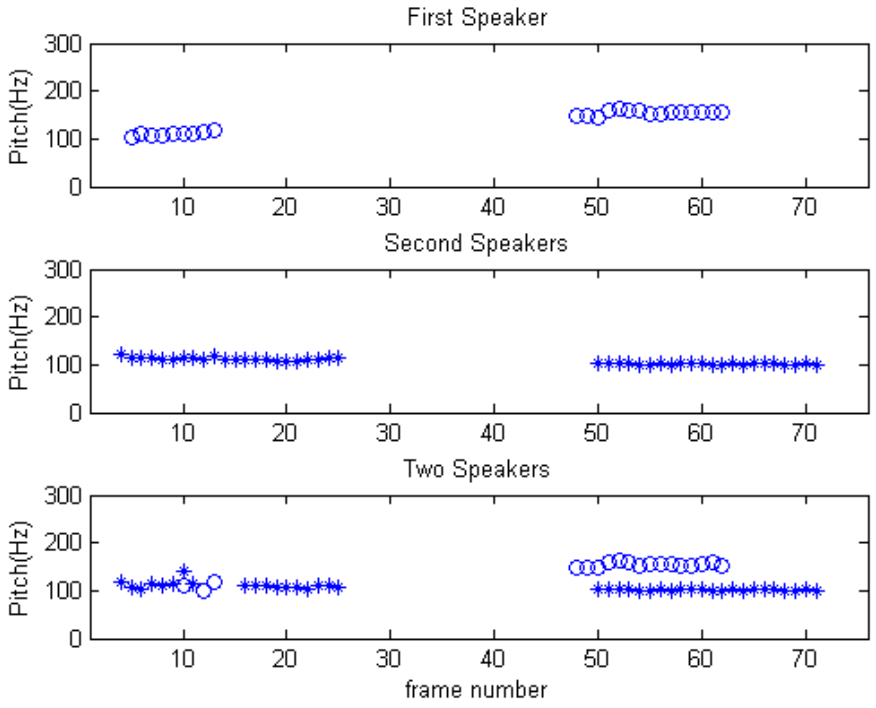


Figure 3.1: Multi-pitch determination performance of the 2-D AMDF algorithm. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz.

However, when the energy of diffuse noise signal is 10dB more than the total energy of speech signals, the performance of the 2-D AMDF degrades.

On the other hand, the pitch track estimation of two speakers in the presence of directional noise source, 2-D AMDF could not estimate the pitch of the speaker accurately. Figure 3.3 shows the performance of 2-D AMDF multi-speaker pitch tracking algorithm applied to the same speech signals as in Figure 3.2Figure 3.1. In this experiment, we used music as the directional noise source, and the energy of the music signal is within 3 dB of the total energy of the speech signals. Compare to Figure 3.2; we

can see that in most of the frames, the two pitch tracks are tracked inaccurately. However, in frames between, 100 and 140, 2-D AMDF could not track the pitch of the second

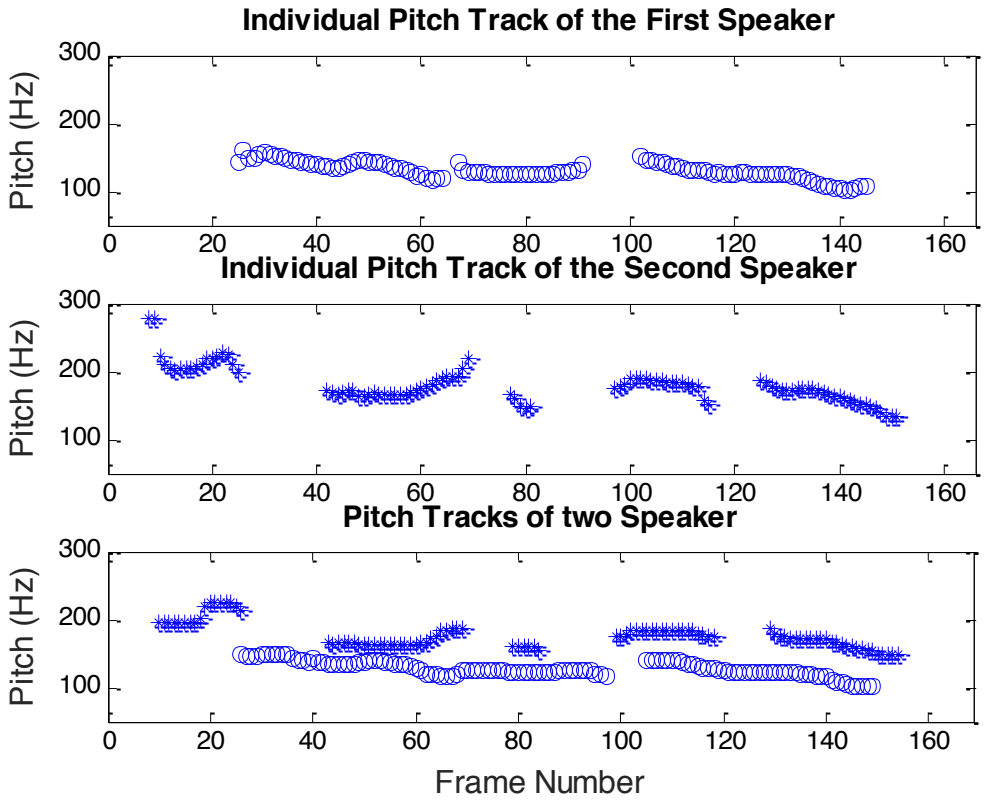


Figure 3.2: Multi-pitch determination performance of the 2-D AMDF algorithm in the presence of diffuse noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz.

speaker, because the second speaker voice is dominated during these frames by music and the algorithm started to track the pitch of the music. We could say from this experiment, 2-D AMDF algorithm is sensitive to the directional noise because of the quasi-periodic nature of the music signal.

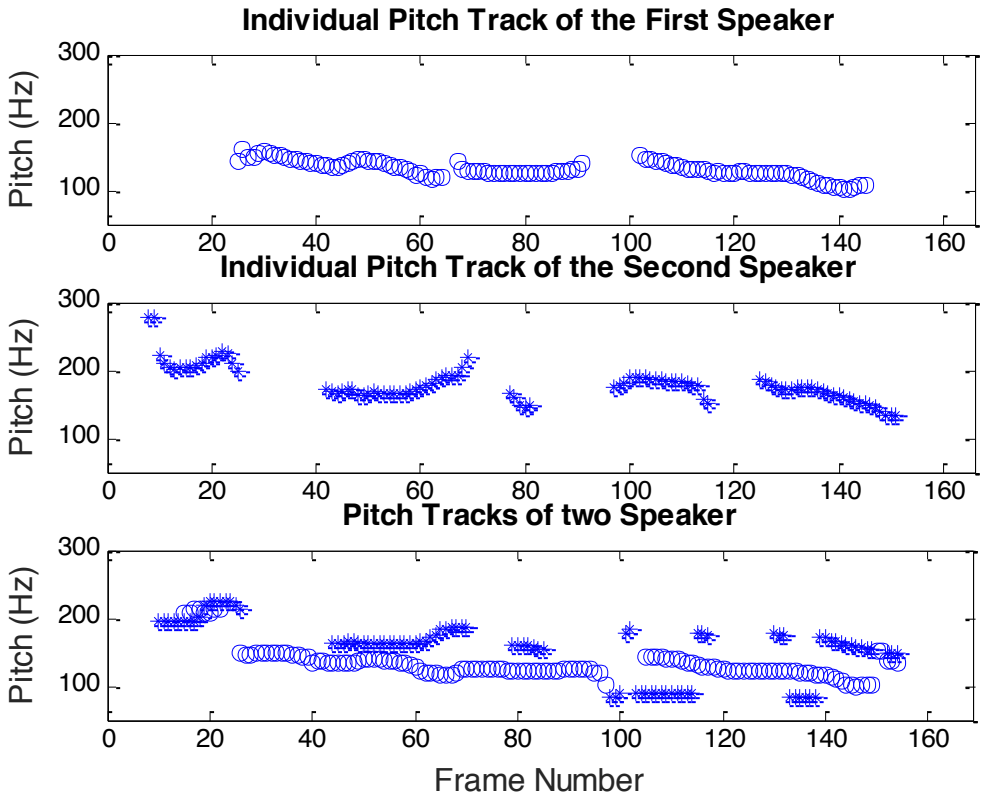


Figure 3.3: Multi-pitch determination performance of the 2-D AMDF algorithm in the presence of directional noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying 2-D AMDF results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz.

3.1.2 Multi-Pitch Tracker (MP TRACKER)

For this thesis, we also implemented and tested MP Tracker algorithm proposed by (Radfar et al., 2011) in order to compare the efficacy of our EAR- based approach for multi-speaker pitch tracking problem. The proposed algorithm aims to track and separate the pitch frequencies of two speakers from their mixture signals synthesized without any noise signal. The pitch frequencies are detected using spectral distortion optimization, which takes into account the sinusoidal modeling of the speech signal. MP Tracker

detects the pitch frequencies and assigns them to individual speakers. This algorithm consists of four stages: detection, grouping, and separation. In detection stage, MP tracker utilizes the log spectral distortion between the mixture spectrum and the parametric spectra of the underlying speech signals. Log-spectral distortion is calculated as:

$$d_{LS}(\theta_1, \theta_2) = \frac{1}{2\pi} \int_0^{2\pi} \left| \frac{P_{x1}(\omega; \theta_1) + P_{x2}(\omega; \theta_2) - P_y(\omega)}{P_y(\omega)} \right| d\omega. \;\; i \in \{1,2\} \qquad (30)$$

where $P_{xi}(\omega; \theta_i)$ and $P_y(\omega)$ represents the parametric power spectral densities of two speech signals and their original mixture, respectively, where ω and θ_i denote the angular frequency and the model parameters, respectively. Sinusoidal model is applied to the speech signal since the sine waves are well resolved for a speech signal and due to the fact that power spectral density is the square of the short time Fourier transform. Power spectral density can be written as:

$$P_{xi}(\omega; \theta_i) \approx \sum_{l=1}^{L} A_l^i W\left(\omega - \omega_l^i\right) \qquad (31)$$

where $\{\omega_l^i, A_l^i\}_{l=1}^L \in \theta_i$ represents the frequencies and squared amplitudes of the sinusoidal model given by $x_i(n) = \sum_{l=1}^{L} A_l^i \exp\left(j(n\omega_l^i + \varphi_l^i)\right)$. Also, $W(\omega)$ denotes the squared amplitude of the Fourier transform of a Hanning window. Since speech signal is quasi-periodic for voiced speech, spectral peaks occur at $\omega_l^i = l\omega_o^i + \alpha$ where ω_o^i denotes the fundamental frequency and α is a heuristic parameter, which determines the search interval for peak occurrence. The amplitude of the sinusoidal model is represented as:

$$A_l^i = \left(1 - \frac{1}{2} \exp\left(-\beta |\omega_l^1 - \omega_l^2|\right)\right) P_y(\omega_l^i)$$

where β is a scale factor. This formula becomes $A_l^i \approx P_y(\omega_l^i)$ when peaks are well separated. Having these setups, the spectral distortion (30) becomes a function of ω_o^1 and ω_o^2. Then the spectral distortion is minimized with respect to ω_o^1 and ω_o^2, i.e.

$$\tilde{\omega}_o^1, \tilde{\omega}_o^2 = arg \min_{\omega_o^1, \omega_o^2} d_{LS}(\omega_o^1, \omega_o^2) \qquad (32)$$

to estimate the pitch frequencies of underlying speech signals. After performing the pitch detection algorithm, two pitch candidates are obtained for each frame. The grouping stage is to group pitch candidates in order to compromise continuous, smooth tracks. These tracks are used in the next stage to separate the pitch contours of two speakers. In the grouping stage, let ω^t be the pitch candidate at frame t. Then, at frame t+1 a search is made over the interval $[\omega^t - \rho; \omega^t + \rho]$ for a pitch candidate. If a candidate, ω^{t+1}, exists then the value is appended to the track. This process is repeated until no candidate is found within the search interval. If two candidates ω_1^{t+1} and ω_1^{t+2} lie in the search interval at an identical distance with respect to ω^t, the pitch candidate with the smoother angle with respect to the previous candidates is chosen to prevent the abrupt changes. Performing the grouping stage, the set of tracks $G = g\{q\}_{q=1}^{Q}$ are obtained where $g\{q\}$ represents a group of (t, ω) which construct a track. In the separation stage, these tracks are assigned to corresponding speakers. Each track is considered as a cluster of data and the mean of each cluster as representative of that cluster. The q^{th} track is represented as $g\{q\} = \{\omega^t, \dots \omega^{t+1}, \dots, \omega^{t+P-1}\}$ with length P. Then the mean of the q^{th} cluster is defined as

$M^q = \frac{1}{P}\sum_{p=0}^{P-1}\omega^{t+p}$. The pitch tracks are classified into two groups (I and II), one belonging to each speaker. First, the longest track q^* is identified and its mean, M^{q^*}, is compared with the means of other tracks. The tracks whose means satisfy, $\left|M^q - M^{q^*}\right| > \gamma$ are classified into group II. The remaining tracks are classified into group I including the longest track. In case of two overlapping tracks are classified in the same group, re-allocation is done for these tracks. First overlapped tracks are determined in each group and then their means are compared with the mean of the longest track in each group. The overlapped track, which is closest to the mean of the longest tracks in the group, is kept and the other track is transferred to the other group. The obtained pitch contours having missing pitch frequencies are interpolated linearly to recover them. One disadvantage of this algorithm is that as the pitch difference gets less than 10 Hz or crossing pitch contours, the accuracy begins to degrade. Another disadvantage is that as the relative loudness of one speaker begins to exceed the other speaker by 18dB, the accuracy also degrades.

3.1.2.1 Implementation and Experimental Results

We tested the MP Tracker algorithm with one microphone signal recording of two simultaneous speakers without background noise. We implemented AMDF technique as described above. We applied 32 ms Hanning window with an overlapping factor of 50% to divide the entire signal into frames. In the experiment stage, we observed that MP Tracker fails in two situations. The first case is when the energy difference of two

speaker signals at a frame is more than 18dB so, the algorithm cannot estimate the pitch of the speaker correctly who has less energy. The second case is that when the pitch values of the speakers are too close, less than 10 Hz.

The efficacy of MP Tracker algorithm is also tested in the presence of directional noise and diffuse noise environment. Figure 3.4 presents the performance of the MP Tracker in the presence of restaurant ambient noise synthesized with 3-second recorded speech signals where two male speakers were talking simultaneously. The energy ratio between the speech signals and directional noise is roughly 0dB. The plot on the top represents the ground truth of the first speaker's pitch while the plot in the middle corresponds to the ground truth of the second speaker's pitch. The plot at the bottom shows the pitches estimated by MP Tracker. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The first two plots correspond to reference pitches of the two speakers individually. As it can be seen Figure 3.4, MP Tracker could still estimate the pitch of the speakers accurately in the presence of diffuse noise. MP Tracker algorithm. However, the ratio of the energy of diffuse noise signal to the total energy of speech signals is more than 10 dB, the performance of the MP Tracker decreases.

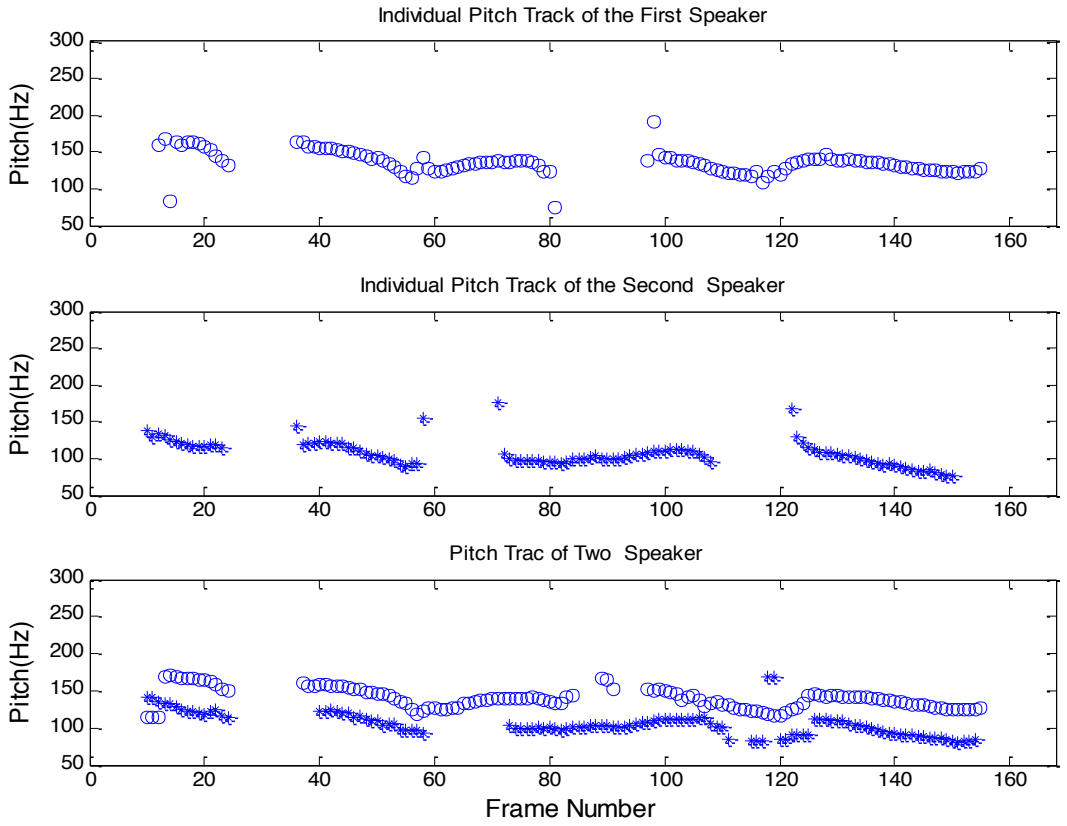


Figure 3.4: Multi-pitch determination performance of the MP Tracker algorithm in the presence of diffuse noise as background noise. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying MP Tracker results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz.

On the other hand, the pitch track estimation of two speakers in the presence of directional noise source, MP Tracker could not estimate the pitch of the speaker accurately. Figure 3.5 shows the performance of MP Tracker multi-speaker pitch tracking algorithm applied to the same speech signals in Figure 3.2. In this experiment, we used the music as the directional noise source, and the energy of the music signal is less than 3 dB from the total energy of the speech signals. Compare to the ground truths; we can see that in most of the frames, the two pitches are tracked inaccurately. However, in frames

between, 40 and 140, MP Tracker could not track the pitch of the speakers, because the speaker's voice is dominated during these frames by music and the algorithm started to track the pitch of the music. We could say from this experiment; MP Tracker algorithm is sensitive to the directional noise because of the quasi-periodic nature of the music signal.

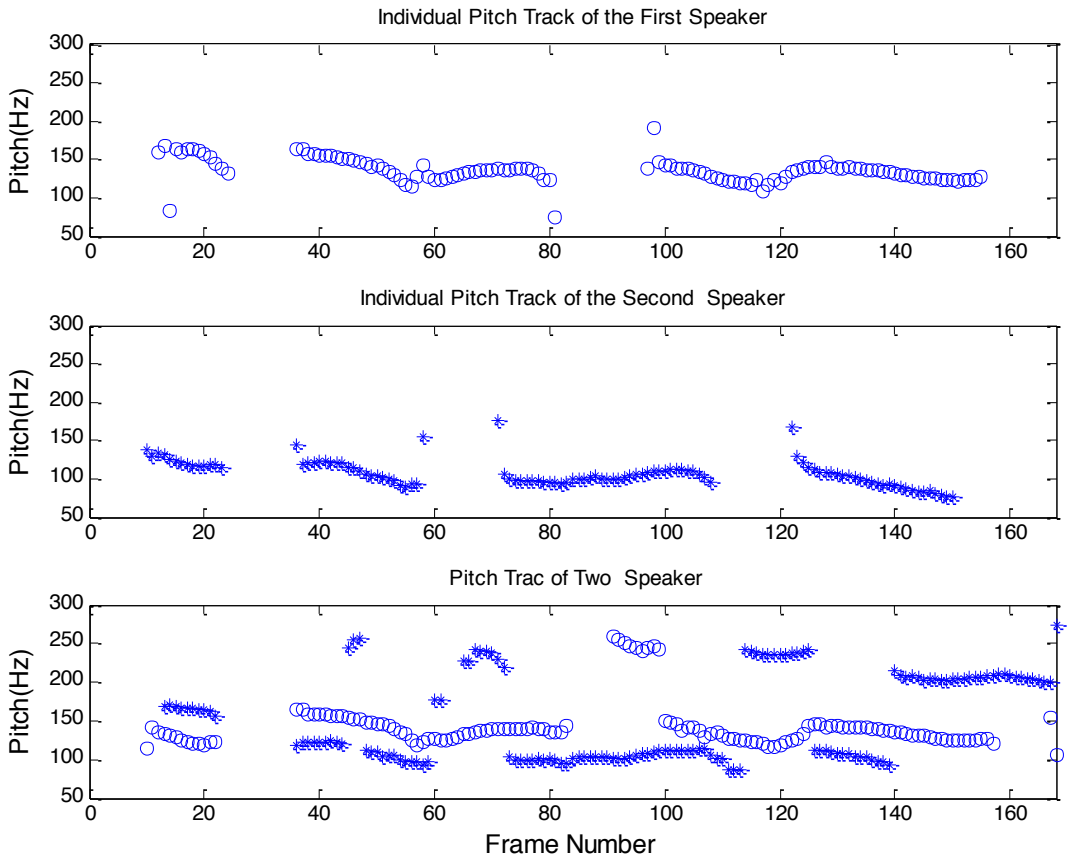


Figure 3.5: Multi-pitch determination performance of the MP Tracker algorithm. The plot on the top is the reference pitch of the first speaker. The plot in the middle corresponds to reference pitch of the second speaker while the figure in the bottom is the represents the result of applying MP Tracker results. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz.

3.2 State of the Art Multi-Speaker Pitch Tracking Methods and Their Limitations and Challenges

As has been discussed in Sec 3.1, one of the current multi-pitch tracking algorithms utilizes 2-D Average Magnitude Difference Function (AMDF), and on certain features extracted from the 2-D AMDF, it extracts the periodicity information of the two speakers. The other multi-pitch tracking algorithm we discussed is the frequency-domain approach called MP Tracker. The idea is to find the "dominant" fundamentals that have generated most of the peaks in the spectrum. The pitch is estimated from the harmonic information along the frequency axis for each source. The pitch values are then used for grouping and separating the harmonics of each source signal.

Before we investigate the challenges of these multi-pitch algorithms, we first describe the ideal performance desired from multi-pitch tracking. Figure 3.6 shows a waveform of a mixed speech signal containing speech from two male speakers (speaker A and speaker B), along with the periodogram and the individual pitch tracks as obtained by one of the single pitch tracking algorithms (Boersma & Weenink). The audio recording itself is 2.5 seconds long. In the first panel, the x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The middle panel shows the periodogram of the mixture signal using a transform (Fast Fourier Transform). The x-axis represents the time in seconds, and the y-axis corresponds to the frequency. The third panel represents the pitch track of the audio signal. The red lines represent the pitch track of speaker A while blue lines represent the pitch track of speaker

B. The individual pitch tracks of the speech signal are found using autocorrelation method (Boersma & Weenink) and overlaid for demonstration purpose.

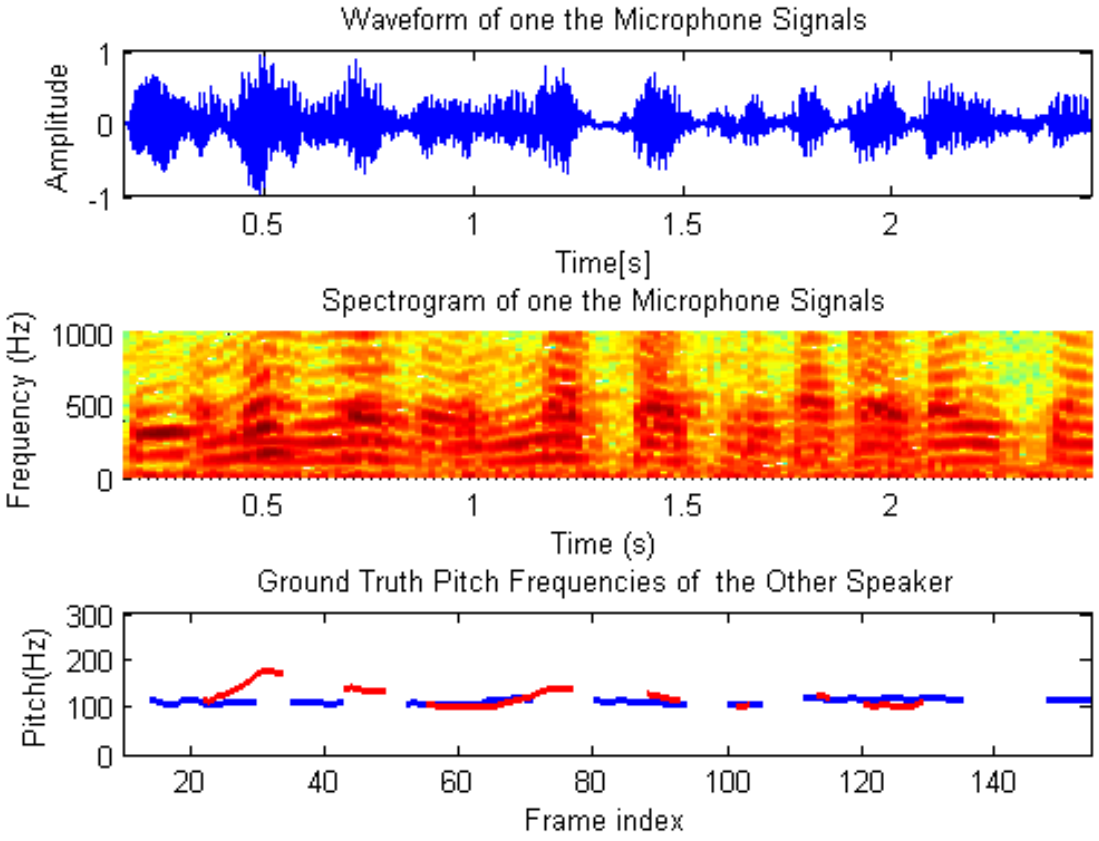


Figure 3.6: Explanation of the performance of an ideal multi-pitch tracker. First Panel: Waveform of a mixed speech signal containing speech from two male speakers (speaker A and speaker B). The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. Middle Panel: Periodogram of mixture signal containing speech from speaker's A and B where the x-axis represents the time in seconds, and the y-axis corresponds to the frequency. Third Panel: Pitch track of mixture signal found by (Boersma & Weenink), with red lines representing pitch track of speaker A and blue lines representing pitch track of speaker B.

As mentioned above, we tested the 2-D AMDF and MP Tracker algorithm with one microphone signal recording of two simultaneous speakers without background noise as well as with the different type of background noise. We implemented both techniques

as described in Sec 3.1. We applied 32 ms Hanning window with an overlapping factor of 50% to divide the entire mixture signal into frames.

In the experiment stage, we observed that both 2-D AMDF and MP Tracker methods are affected by two factors with a mixture signal containing two speaker's speech. The first factor is the energy difference of two speaker signals. The second factor is the difference between the pitch values of speakers. The first factor for 2-D AMDF is that as the pitch difference gets less than 10 Hz, the accuracy begins to degrade. The second factor is that as the relative loudness of one speaker begins to exceed the other speaker by 10 dB, the accuracy also degrades. On the other hand, the pitch difference factor for MP Tracker when the pitch difference gets less than 10 Hz and the relative loudness of one speaker begins to exceed the other speaker by 18 dB, the accuracy degrades.

Figure 3.7 shows an example of applying 2-D AMDF and MP Tracker to a 2.5-second audio recording where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The ratio of their total energies is roughly 0dB. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. The pitch of the speaker A is varying between 120Hz to 150Hz during the recording while the pitch

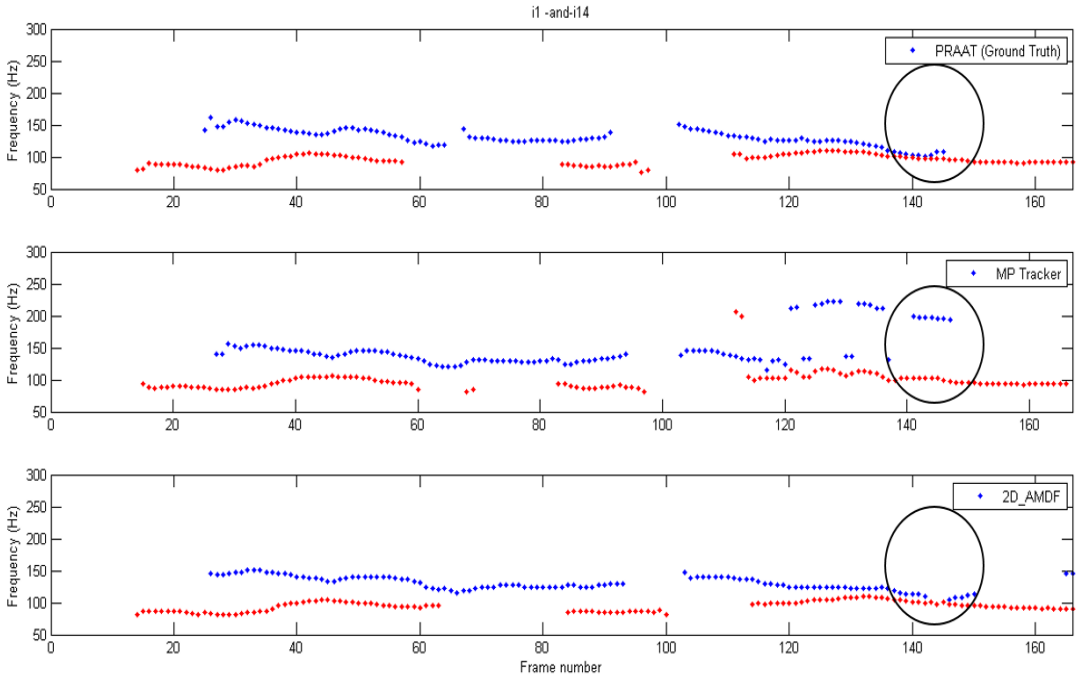


Figure 3.7: An example of applying 2-D AMDF and MP Tracker to a 2.5-second audio recording where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The ratio of their total energies is roughly 0dB. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted as-is. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B.

of the speaker B is between 80Hz and 120Hz during the audio recording. As can be seen from the frames between 135 and 145, the pitches of the two speakers are very close where the difference is less than 10Hz. In this case, the 2-D AMDF mostly gives one pitch value since AMDF estimates that there is one speaker while MP Tracker gives pitch values irrelevant with true pitch values for some reasons. These reasons will be explained in detail in Chapter 4. In the frames between 20 and 120, the pitches are far away from each other, and 2-D AMDF and MP Tracker give accurate pitch estimate in most frames.

Figure 3.8, we give an example of the performance multi-pitch tracking algorithms and examine how the performance of algorithms are influenced by the pitch difference factor. Figure 3.9 shows the performance of multi-pitch tracking algorithms in the frames where the energy difference between two speakers is more than 15dB. The audio recording is 3-second long where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. The pitch of the speaker A is varying between 80Hz to 135Hz during the recording while the pitch of the speaker, B is between 110Hz and 150Hz during the audio recording. The highlighted area between the frame number 80 and 110, the energy difference between the two speakers is more than 15dB. As can be seen from Figure 3.8 even if the pitches are far away from each other, the energy difference affects the algorithms working properly. In this case, the 2-D AMDF mostly gives one pitch value since AMDF estimates that there is one speaker due to the domination of speaker B in terms of energy. MP Tracker gives correct pitch estimate of the dominant speaker, but it assigns the pitch to the speakers incorrectly frames between 80 and 110. The two pitch values are estimated correctly by the algorithm in most of the frames out of this highlighted area.

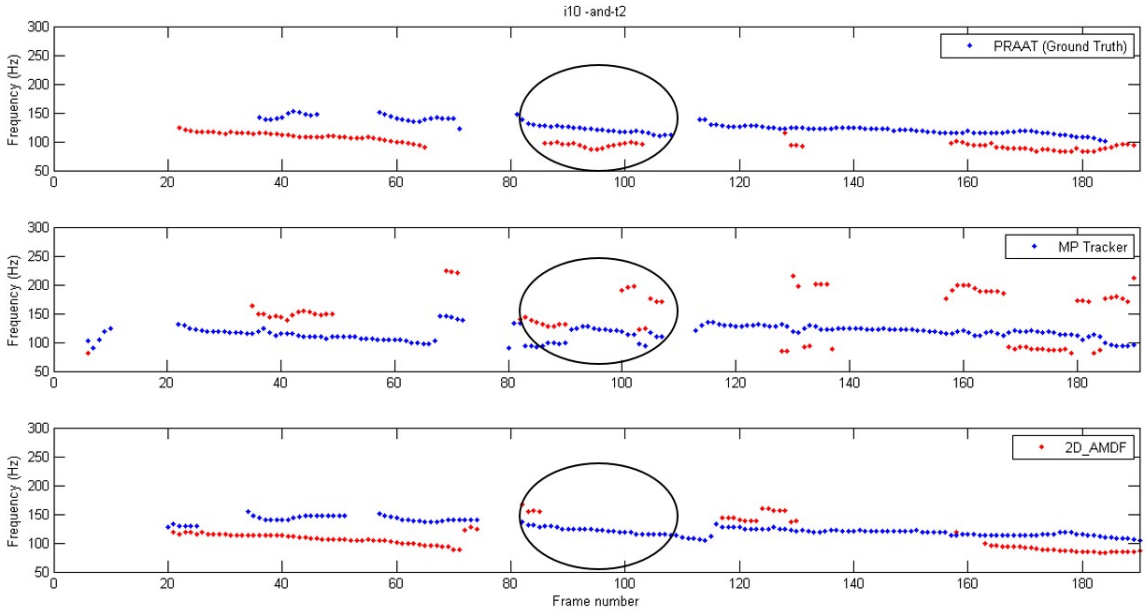


Figure 3.8: Performance of multi-pitch tracking algorithms when the energy difference between two speakers is more than 15dB. The audio recording is 3-second long where two male speakers (speaker A and speaker B) were talking simultaneously. Both the true (ground truth) pitch values, as well as the estimated pitch tracks obtained by these algorithms, are shown. The top plot shows the true pitch values estimated by (Boersma & Weenink) are plotted as-is. The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. The pitch of the speaker A is varying between 80Hz to 135Hz during the recording while the pitch of the speaker B is between 110Hz and 150Hz during the audio recording.

The efficacy of 2-D AMDF and MP Tracker algorithms is also tested in the presence of directional noise and diffuse noise environment. Figure 3.9 presents the performance of the algorithms in the presence of restaurant ambient noise synthesized with a 3-second recorded mixture speech signal where two male speakers were talking simultaneously. The difference of total energy between the mixture signal and diffuse noise is roughly 0dB. The plot on the top represents the ground truth of the speakers estimated by

(Boersma & Weenink). The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B. As it can be seen from Figure 3.9, 2-D AMDF could still estimate the pitch of the speakers accurately in the presence of diffuse noise while MP Tracker is more sensitive to the diffuse noise. The performance of MP Tracker decreases when the ratio of the energy of diffuse noise signal to the total energy of mixture signal is more than 5dB. However, the ratio of the energy of diffuse noise signal to the total energy of speech signals is more than 10 dB, the performance of the 2-D AMDF decreases. On the other hand, the pitch track estimation of two speakers in the presence of directional noise source is more difficult. Figure 3.10 shows the performance of multi-pitch tracking algorithms applied to the same speech signals in Figure 3.7 contaminated by directional noise (music). The energy of the music signal is less than 3 dB from the energy of mixture signal. Compare to the Figure 3.7; we can see that in most of the frames, the two pitches are tracked inaccurately by MP Tracker. The estimated pitch tracks of 2-D AMDF is more robust to the directional noise than MP Tracker. We can clearly see that in frames between, 140 and 160, both algorithms give pitch track of the music. 2-D AMDF, and MP Tracker algorithms could not estimate the pitch of the individual speaker accurately the speaker's speech are masked by a noise signal contains periodic signal. We could say from this experiment, 2-D AMDF and MP Tracker algorithms are sensitive to the noise which has a nature of the quasi-periodic signal.

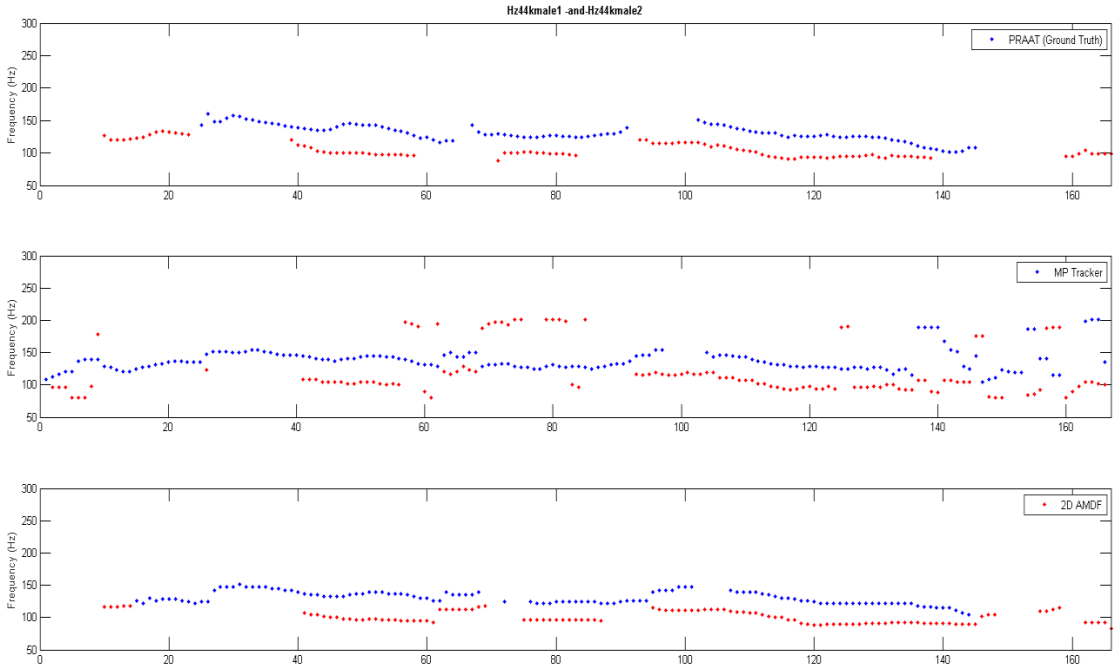


Figure 3.9: Performance of the algorithms in the presence of restaurant ambient noise synthesized with a 3-second recorded mixture speech signal where two male speakers were talking simultaneously. The difference in total energy between the mixture signal and diffuse noise is roughly 0dB. The plot on the top represents the ground truth of the speakers estimated by (Boersma & Weenink) . The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B.

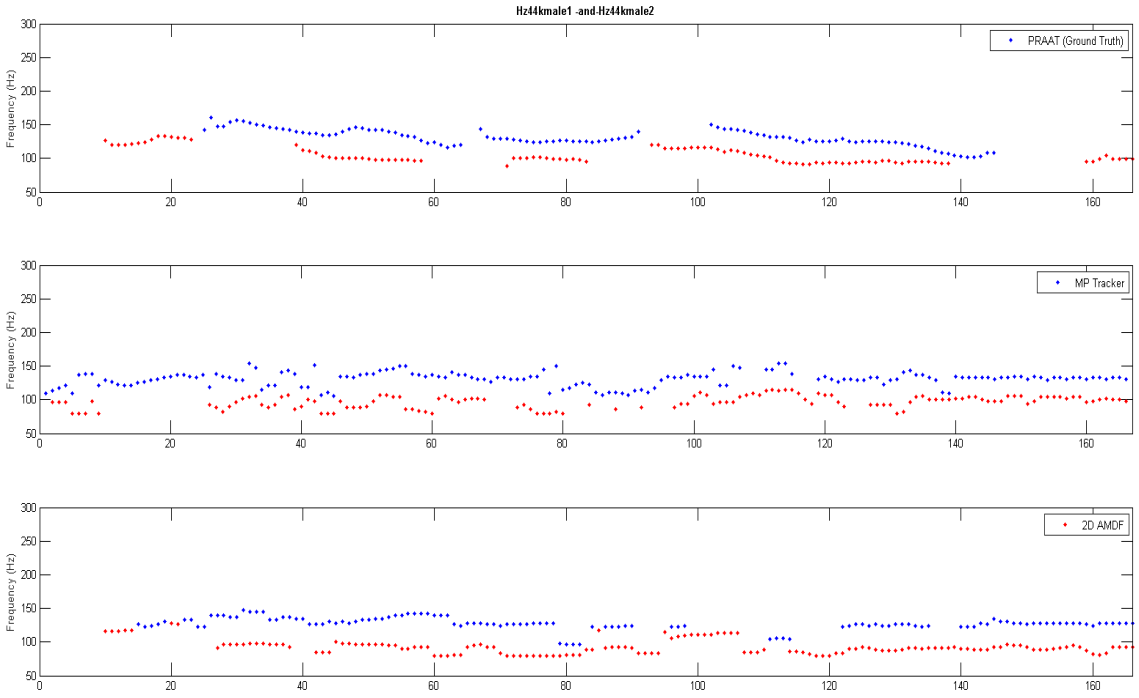


Figure 3.10: Performance of the algorithms in the presence of directional noise synthesized with a 3-second recorded mixture speech signal where two male speakers were talking simultaneously. The difference in total energy between the mixture signal and directional noise is roughly 0dB. The plot on the top represents the ground truth of the speakers estimated by (Boersma & Weenink). The second plot shows the pitch values estimated by MP Tracker while the plot at the bottom 2-D AMDF represents the estimated pitch values. The x-axis represents the frame number, and the y-axis corresponds to the pitch in Hz. The red dots represent the pitch track of speaker A, while blue dots represent the pitch track of speaker B.

In this thesis, we have developed an approach that uses a combination of existing speech enhancement, speech separation, and pitch detection techniques in conjunction with *early abductive reasoning* to obtain multiple pitch tracks in unstructured noisy environments. In the next section, we will discuss the speech enhancement and speech separation techniques that allow us to apply early abductive reasoning approach on the multi-speaker pitch tracking application.

3.3 Speech Enhancement Techniques

3.3.1 Coherence-Based Filtering

Coherence based filtering for speech enhancement was first proposed in reference (Allen et al., 1977) for reducing "diffuse" noise (noise that comes from many different directions rather than from a few discrete directions) in a noisy multi-microphone recording. The basic idea behind coherence-based filtering is to preserve the inter-microphone correlated signal (mostly speech) while removing the uncorrelated signals by a mask generated by coherence function values. If the magnitude of coherence function between two channels of the noisy input signal is one (or close to one), it is assumed that the dominant signal is speech and the noisy signal is passed without attenuation. If the magnitude is close to zero, the dominant signal is assumed to be noise, and the noisy input signal should be suppressed.

In reference (Abdipour et al., 2014) coherence-based filtering is used to attenuate the diffuse noise signal from the noisy input signal while preserving the speech intelligibility and quality. The performance of coherence-based filtering is useful in the case of the diffuse noise signal, but in "directional" noise the coherence-based method does not perform well since "directional" noise is also correlated at the two microphones.

Coherence Function

In noisy environments, the microphone system can be written as:

$$x_i(t) = s_i(t - \tau_i) + n_i(t) \qquad i = 1,2 \quad (1)$$

Here $x_i(t)$, $s_i(t)$ and $n_i(t)$ denote the noisy speech, speech, and the noise signals respectively captured by microphone i. Here, τ_i is the relative time delay between the two microphones and is formulated as $\tau_i = (\delta/c)\cos\Theta$ where δ is the distance between two microphones, c is the speed of sound in air, and Θ is the azimuth angle relative to the line that bisects the inter-microphone distance. We need to point out that the assumption on the signals is that noise signals and speech signal are uncorrelated.

After applying the short time Fourier Transform (STFT) on both sides of equation (1), the input signal is divided into time-frequency units. The power spectrum of the signal at microphone i can be written as:

$$X_i(\lambda, k) = S_i(\lambda, k) + N_i(\lambda, k) \qquad i = 1,2 \qquad (2)$$

λ and k represent the frame and frequency bin indices and $X_i(\lambda, k)$ are the spectra of microphone signals. Based on the spectra of the received signals coherence is computed as (Abdipour et al., 2014):

$$coh(\lambda, k) = \frac{|P_{X_1\ X_2}\ (\lambda,k)|}{\sqrt{|P_{X_1}\ (\lambda,k)|.|P_{X_2}\ (\lambda,k)|}} \qquad (3)$$

Here $P_{X_i}(\lambda, k)$ is the smoothed power spectrums of the signal x_i and is calculated recursively as in equation (4).

$$P_{X_i}(\lambda, k) = \alpha . P_{X_i}(\lambda - 1, k) + (1 - \alpha).|X_i(\lambda, k)|^2 \qquad (4)$$

$P_{X_1\ X_2}(\lambda, k)$ is the smoothed cross power spectral density (CPSD) of $X_1(\lambda, k)$ and $X_2(\lambda, k)$ and calculated as:

$$P_{X_1\ X_2}(\lambda, k) = \alpha P_{X_1\ X_2}(\lambda - 1, k) + (1 - \alpha).X_1(\lambda, k).X_2{}^*(\lambda, k) \quad (5)$$

Here, α is the smoothing factor and * denotes the conjugate transpose operations.

The coherence between two signals $x_1(t)$ and $x_2(t)$ at frequency bin, k shows the level of correlation between $X_1(\lambda, k)$ and $X_2(\lambda, k)$. When the coherence value is closer to one, this indicates high correlation between the Fourier components of the microphone signals. That means they are linearly dependent and only differ in terms of time of arrival and magnitude. A binary mask, $BM(\lambda, k)$, is then applied to the one of the microphones to get enhance signal spectrum:

$$BM(\lambda, k) = \begin{cases} 1, & coh(\lambda, k) \geq th \\ 0, & coh(\lambda, k) < th \end{cases} \qquad (6)$$

$$\widehat{S}(\lambda, k) = BM(\lambda, k).X_1(\lambda, k) \qquad (7)$$

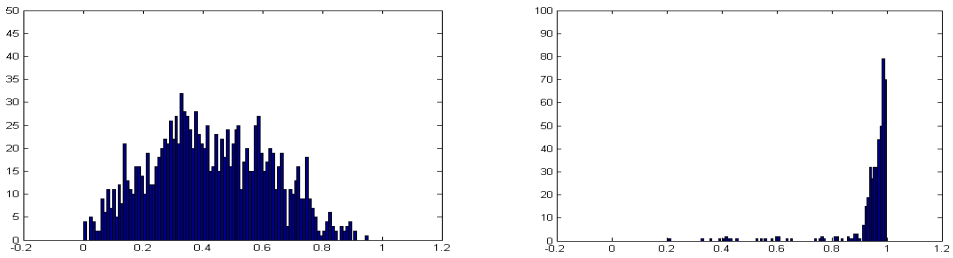
Here, th is the threshold for separating diffuse and directional dominated T-F units and $\widehat{S}(\lambda, k)$ is the enhanced spectrum. Finally, the inverse FFT may be taken in order to reconstruct enhanced signal $\hat{s}(n)$.

$$\hat{s}(n) = IFFT(\widehat{S}(\lambda, k)) \qquad (8)$$

3.3.1.1 Implementation and Experimental Results

In testing our implementation of coherence-based filtering, we used restaurant ambient noise for diffuse noise source. We first divide the signal into 32 ms frames with an overlapping factor 50%. Each frame is multiplied with a Hanning window and transformed into the frequency domain by applying short-time Fourier Transform (STFT) with an FFT length 1400. Then we find the coherence value of each Time-Frequency (T-F) unit based on the formula in (3). Part (a) and (b) in Figure 3.11 show the histogram of coherence values for diffuse noise and target speech dominated T-F units centered at 2

kHz respectively. As can be seen from Figure 3.11, in the case of the directional source signal, the coherence values are concentrated around 1. On the other hand, in the case of the diffuse source signal, the coherence values of T-F units are scattered between 0 and 1.

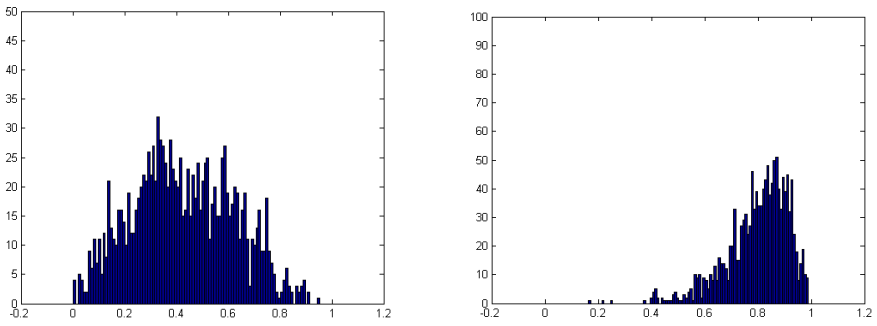


(a) *(b)*

Figure 3.11: (a) Histogram of coherence of diffuse dominated T-F units centered at 2 kHz. (b) Histogram of coherence of directional dominated T-F units centered at 2 kHz. The axes are the coherence values and its probability.

As seen from Figure 3.11, a threshold, for example, 0.9, can separate speech dominated T-F units from diffuse source dominated T-F units. By designing a filter based on the coherence values, which attenuates the T-F units having coherence value less than a threshold, a significant amount of noise can be suppressed. It should be pointed that in low frequencies for the diffuse source dominated T-F units are becoming coherent because low frequencies have large wavelengths and the phase differences become negligible between closely spaced microphones. Figure 3.12 (a) and (b) shows the coherent values of T-F units of a diffuse noise signal centered at 2 kHz and 200 Hz respectively. The advantage of this method is that coherence based filtering does not require noise statistics estimation, but parameter selection (threshold) and gain of the mask have significant effects on the performance of coherence based filtering. A small threshold will cause residual noise in the output, and a higher threshold will cause

distortion on the speech. Beside this, a binary mask (0 and 1) will also produce musical noise in the output because of the transition from coherent to non-coherent T-F units. In order to overcome this problem, we may apply a continuous mask using Gaussian distribution on coherence values.



(a) *(b)*

Figure 3.12: (a) Histogram of coherence of diffuse dominated T-F units centered at 2 kHz. (b) Histogram of coherence of diffuse dominated T-F units centered at 200 Hz. The axes are the coherence values and its probability.

We applied two different Gaussian distributions for the low and high-frequency T-F units. Since the T-F units in the low frequencies are prone to be coherent, we apply a narrow Gaussian distribution for the low frequencies in order to suppress diffuse noise dominant T-F units in the low frequencies. In order to preserve the speech quality, we apply a wider Gaussian distribution. Coherence based filtering is easy to implement, computationally inexpensive and is able to suppress diffuse source signal sufficiently.

3.3.2 Phase Error-Based Filtering

The phase error is also sometimes used as a feature in the literature for dual-microphone speech enhancement system (Rahmani et al., 2009) (Arabi and Guannji, 2004). The main

goal is to enhance the desired speech from a directional noise source by using the direction of the noise source relative to the desired source. If our signal model is defined in (1) and $X_1(\lambda, k)$ and $X_2(\lambda, k)$ are the two spectra of each microphone, then phase error may be calculated as:

$$PE(\lambda, k) = \Delta\varphi(\lambda, k) - 2\pi k.ITD \quad (9)$$

Here, $\Delta\varphi(\lambda, k) = \angle X_1(\lambda, k) - \angle X_2(\lambda, k)$ and ITD is the time delay of the desired speech source which is given. Phase error (PE) is clustered around zero when T-F unit is speech dominated, otherwise PE will be far from zero, around -π and π because PE is wrapped between -π and π. In (Abdipour et al., 2014), phase error is used to mask T-F units in order to suppress directional noise using parameterized scaling strategy. A binary mask, $BM(\lambda, k)$, is then applied to one of the microphones to get an enhanced signal spectrum:

$$BM(\lambda, k) = \begin{cases} 1, & |PE(\lambda, k)| < th \\ 0, & |PE(\lambda, k)| \geq th \end{cases} \quad (10)$$

$$\widehat{S}(\lambda, k) = BM(\lambda, k).X_1(\lambda, k) \quad (11)$$

Here, th is the threshold for separating speech and directional noise source dominated T-F units and $\widehat{S}(\lambda, k)$ is the enhanced spectrum. Finally, inverse FFT may be taken in order to reconstruct enhanced signal $\hat{s}(n)$.

$$\hat{s}(n) = IFFT(\widehat{S}(\lambda, k)) \quad (12)$$

3.3.2.1 Implementation and Experimental Results

In our implementation, we used music signals as a noise sources and speech signals for directional source signal. The same parameters are used for taking STFT of the

microphone signals. The music noise and the speech noise are placed at $\pm$90° where 0° is perpendicular to the line, which combines the two microphones.

Figure 3.13 shows the histogram of phase error values of T-F units of a directional noise signal centered at 2 kHz where music source and speech source are placed at -90° and +90° respectively. Phase error values are centered on zero for speech dominated TF units, and phase error values are far from zero (around $\pm\pi$) for noise dominated TF units. By a threshold on the phase error values, we could separate the speech signal from the noise signal. We can then apply a continuous mask using Gaussian distribution instead of binary mask on phase error value in order to prevent musical noise in the output.

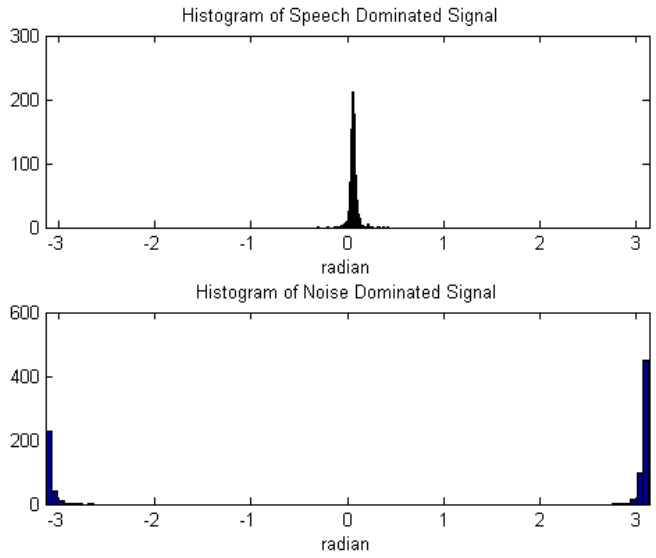


Figure 3.13: Comparison of speech dominated and directional noise dominated TF units, and noise source and speech source are placed -90° and 90° respectively. Phase error values are centered around zero for speech dominated TF units, and its value is far from zero (around $\pm\boldsymbol{\pi}$) for noise dominated TF units.

3.3.3 Minimum Variance Distortionless Response (MVDR)

Minimum variance distortionless response is an advanced beamforming technique that performs data-dependent spatial filtering on microphone arrays in order to enhance the desired speech signal. The basic idea is to place a null in the direction of the noise source while keeping unity gain in the direction of the desired source. Beamforming algorithms may be divided into two categories: static beamformers and adaptive beamformers. Static beamformers have fixed filter coefficients. The Delay-and-Sum beamformer, originally developed for sonar and radar systems (Pan et al., 2014), is the simplest form of the static beamformer. First, microphone signals are shifted by a proper amount of time to synchronize the desired signal component across all sensors. Then these delayed signals are weighted and summed together. Because its components add up constructively, the desired signal is amplified while the sources from other directions are suppressed due to destructive interference. It can be shown that this technique is most effective when the signals from the different directions are narrowband signals of the *same* temporal frequency. When the signals are broadband (e.g., speech), the fixed coefficients of the delay-and-sum beamformer give it different spatial responses at a different frequency, thus causing distortion of the desired signal and artifacts in the residual noise (Ward et al., 1998). Decomposing the microphone signals into sub-bands and applying beamforming to each sub-signal solves this problem (Frost, 1972). This method is called filter-and-sum beamforming, and it is more effective in suppressing noise than delay-and-sum beamforming. However, it is still limited in the sense that the beam pattern (and thus the nulls) of the microphone array is fixed at each temporal frequency.

In real audio environments, noise sources may come from different directions at different times. Adaptive beamformers can cope with this situation by adapting the filter-and-sum beamforming coefficients according to the evolving statistics of the noise field. They can thus adaptively place nulls in the directions of noise sources and thereby more significantly suppress the noise. The MVDR technique for adaptive beamforming was originally developed by Capon (Capon, 1969) and attracted a great deal of attention in the field of acoustic signal processing. In reference (Pan et al., 2014), the practical implementation of MVDR is explained in detail for different noise scenarios (white noise, diffuse noise, diffuse plus white noise and directional noise plus white noise) in order to achieve the best performance of MVDR regarding signal enhancement and noise suppression.

MVDR is a spatial filtering process that is applied to the two microphone signals in order to enhance the desired signal. If our signal model is defined as in (1), the frequency domain representation can be written as:

$$X_i(\omega) = S_i(\omega) + N_i(\omega) \quad (13)$$

$$= e^{-j\omega\tau_i} S(\omega) + N_i(\omega) \quad (14)$$

$$= e^{-j(i-1)\omega\tau_1 cos\theta} S(\omega) + N_i(\omega) \quad (15)$$

Here, $X_i(\omega)$, $S_i(\omega)$, $N_i(\omega)$, and $S(\omega)$ are the Fourier Transform of $x_i(t)$, $s_i(t)$, $n_i(t)$, and $s(t)$ respectively and where $s(t)$ is the reference signal, j is the imaginary unit, $\omega = 2\pi f$ is the angular frequency and f is the frequency. We can rewrite eqn (15) in the following vector form:

$$\boldsymbol{x}(\omega) = \boldsymbol{d}_\theta(\omega)S(\omega) + \boldsymbol{n}(\omega) \quad (16)$$

$$\boldsymbol{d}_\theta(\omega) \triangleq [1 \quad e^{-j\omega\tau_1 cos\theta} \ldots \quad e^{-j(i-1)\omega\tau_1 cos\theta}]^T \text{ (17)}$$

Here, $\boldsymbol{d}_\theta(\omega)$ is the steering vector of the desired speech signal.

The aim of the beamforming is to recover the speech signal $S(\omega)$. This can be achieved by applying a complex weight to microphone signals $X_i\,(\omega)$, and then summing up all the weighted signals together.

$$\hat{S}(\omega) = \sum_{i=1}^{I} H_i{}^*\,(\omega) X_i\,(\omega) \text{ (18)}$$

$$= \boldsymbol{h}^H(\omega)\boldsymbol{x}(\omega) \quad \text{(19)}$$

$$= \boldsymbol{h}^H(\omega)\boldsymbol{d}_\theta(\omega)X(\omega) + \boldsymbol{h}^H(\omega)N(\omega) \text{ (20)}$$

Here, the superscript, *, is the complex conjugate operator and $\hat{S}(\omega)$ is an estimate of $S(\omega)$. The beamformer filter is defined as:

$$\boldsymbol{h}(\omega) = [H_1(\omega) \;\; H_2(\omega) \ldots \ldots \; H_I(\omega)]^{\,T} \text{ (21)}$$

The MVDR beamformer is formulated by minimizing the variance of the residual noise at the output of the beamformer with the constraint that the speech signal is passed through without any attenuation. Mathematically it can be written as:

$$\boldsymbol{h_S}(\omega) = \underset{\boldsymbol{h}(\omega)}{\operatorname{argmin}}\, \phi_{Nrd}(\omega) \text{ (22)}$$

$$subject\ to \;\; \boldsymbol{h}^H(\omega)\boldsymbol{d}_\theta(\omega) = 1 \text{ (23)}$$

Here $\phi_{Nrd}(\omega) \triangleq E\,[\,|\,\boldsymbol{h}^H(\omega)\boldsymbol{n}(\omega)|^2]$ which is the variance of the residual noise at the output. The constrained optimization problem can be solved by the method of Lagrange multipliers and $\boldsymbol{h}(\omega)$ can be formulated as:

$$\boldsymbol{h_S}(\omega) = \frac{\boldsymbol{R}^{-1}(\omega)\boldsymbol{d}_\theta(\omega)}{\boldsymbol{d}_\theta(\omega)^H\boldsymbol{R}^{-1}(\omega)\boldsymbol{d}_\theta(\omega)} \text{ (24)}$$

Here $\boldsymbol{R}^{-1}(\omega)$ is the covariance matrix of the noise. As seen from (24), MVDR beamformer is a function of the steering vector corresponding to the desired signal and the covariance matrix of the noise.

3.3.3.1 Implementation and Experimental Results

We tested our MVDR beamformer implementation with three audio recordings that involve two speakers who were talking simultaneously. In this experiment, the aim was to suppress the voice of one speaker while preserving the other speaker's voice. We first divide the signal into 32 ms frames with an overlapping factor 50%. Each frame was multiplied with a Hann window and transformed into the frequency domain by applying short-time Fourier Transform (STFT) with an FFT length 1400. We directly computed the noise covariance matrix of the noise from a noise only region by averaging 10 frames. Then the MVDR was applied to the noisy signals. To study the effect of the angle between the sources on the performance of the MVDR beamformer, we placed the speakers at $\pm 90°$, $\pm 45°$, and $\pm 15°$, with respect to the microphones.

The three plots in Figure 3.14 represent an example of applying MVDR beamformer to a recording that involves two speakers. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 25 seconds long, and it could be divided into three regions. The first region, starting from 0.5 seconds to 8 seconds, is where the first speaker is speaking alone. The second region, 9 seconds to 15 seconds, is where the second speaker is

speaking by himself. The third region, 15 to 25 second, is where the two speakers talking simultaneously and speakers were placed at -90° and +90°. The plot in the middle corresponds to the spectrogram of the original signal while the plot at the bottom represents the spectrogram of the result of applying the MVDR beamformer to suppress the second speaker. As can be seen from the two spectrograms, in the first region there is no significant change in the harmonic pattern before and after MVDR.

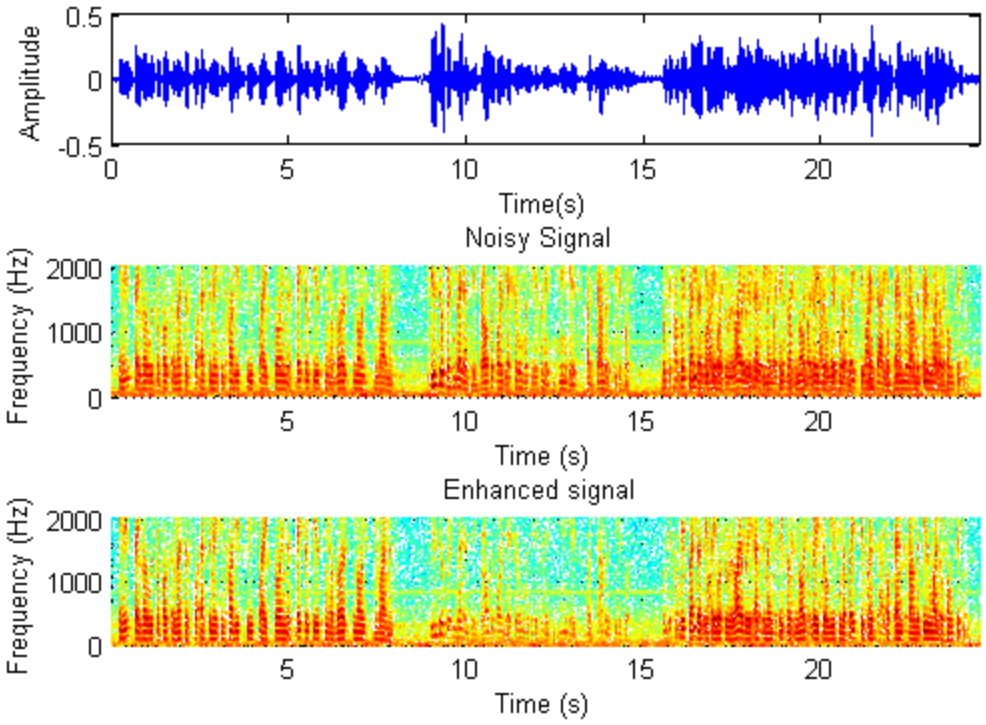


Figure 3.14: An example of applying MVDR beamformer to a recording that involves two speakers. The plot on the top is the waveform of the original (unmodified) audio signal. The x-axis represents the time in seconds, and the y-axis corresponds to the amplitude of the audio signal. The audio recording itself is 25 seconds long, and it could be divided into three regions. The first region, starting from 0.5 seconds to 8 seconds, is where the first speaker is speaking alone. The second region, 9 seconds to 15 seconds, is where the second speaker is speaking by himself. The third region, 16 to 25 second, is where the two speakers talking simultaneously. The plot in the middle corresponds to the periodogram of the original signal while the plot at the bottom represents the periodogram of the result of applying the MVDR beamformer to suppress the second speaker.

In contrast, in the second region where the MVDR is adjusted to suppress the second speaker, the harmonic pattern is weak and hardly visible. When we check the third region, after MVDR, the harmonics of the second speaker can be more easily identified compared to the spectrum of the original signal. Figure 3.15 shows the region between 15 seconds to 25 seconds. Black circles show some examples of the TF units of the second speaker that are suppressed by MVDR.

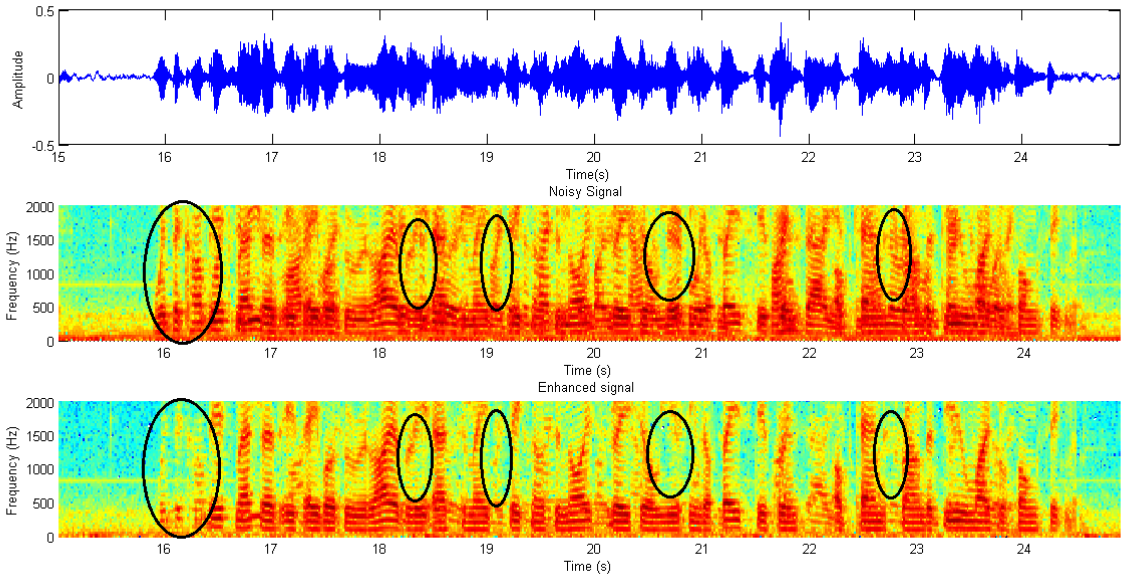


Figure 3.15: The third region between 15second to 25 seconds of the same file where two speakers are talking simultaneously. Black circles show some examples of the TF units of the second speaker that are suppressed by MVDR.

By listening to the three output signals, we observed that the performance of MVDR depends on the angle between the sources. When the angle decreases, the suppression of the unwanted speech is not sufficient to enhance the target speech. If the speakers are in the opposite direction, MVDR gives the maximum performance.

3.3.4 Cross-Correlation Subtraction

The cross-correlation subtraction method is based on aligning two microphone signals according to the noise components then perform subtraction between the microphone signals in order to remove the noise components. In details, the time delay of the noise source is estimated by utilizing its cross-correlation function. This information is used to align the noise components of the microphone signals across all sensors (Ziolko et al., 2009). Then the signals are subtracted from each other after normalization according to an energy ratio. The time delay between two microphone signals is found by finding the maximum value of the cross-correlation. As given before in equation (1), we redefine our system according to a directional noise source. It can be rewritten as:

$$x_i(t) = s_i(t - \tau_s) + n_i(t - \tau_n) \qquad i = 1,2 \quad (25)$$

Here, τ_s is the time difference of the speech and τ_n is the time difference of noise signal. The value τ_n is computed from band-passed microphone signals by utilizing the cross-correlation function where the index of the maximum value of the cross correlation provides the time difference, τ_n, of the noise components of microphone signals. Band-pass filtered signal is used because we try to find frequency bands where only the noise source components exist.

$$r_{X_1\ X_2}(\tau_n) = \underset{\tau}{\operatorname{argmax}} \sum_{n} x_{b1}(n)\, x_{b2}(\mathrm{n} - \tau) \qquad (26)$$

Here, $r_{X_1\ X_2}$ is the cross-correlation values of two microphone signals and x_{b1} and x_{b2} are the band-pass filtered version of the microphone signals. Then speech signal can be found as

$$s(t) = x_1\,(n - \tau_n) - \; kx_2(\mathrm{n}) \qquad (27)$$

Here k is the energy ratio and found as

$$k = \frac{\sqrt{\sum_n (x_{b1}\,(n))^2}}{\sqrt{\sum_n (x_{b2}\,(n))^2}} \qquad (28)$$

3.3.4.1 Implementation and Experimental Results

In our implementation, we used music as directional noise source and speech signal as directional source signal. The directional noise and the speech are placed at $-30°$ and $+30°$ respectively where 0° is perpendicular to the straight line joining the two microphones. We observe that the performance of the cross-correlation subtraction technique depends on the distance between the microphones. When the distance between the two microphones increases, the waveforms of the source signals at the microphones become more different beyond simple delays. For this reason, the background noise cannot be suppressed sufficiently for larger inter-microphone distances. The cross-correlation subtraction technique works well if the noise signal has components in the higher frequencies where the speech components are not present. Otherwise, τ_n may not be correctly calculated and the noise signal components may not be perfectly aligned. That may lead to either distortion on the speech signal or excessive amount of residual noise in the output.

3.3.5 Harmonic Product Spectrum (HPS)

Periodic signals have repetitive peaks in the frequency domain, and these peaks can become the basis for pitch tracking. These peaks correspond to the fundamental frequency and integer multiples of the fundamental frequency (also called harmonic components). Harmonic Power Spectrum (HPS) analysis involves compressing of the spectrums a number of times (decimation) and multiplying the compressed spectra together (Ding et al., 2006). For instance, the first peak in the original spectrum will line up with the second peak of the compressed by a factor of 2 spectra and so on. For this reason, the result of multiplication of such compressed spectra creates a clear peak at the location of the fundamental frequency (Figure 3.16). HPS analysis is a process that is performed in frequency domain. First the input signal, $x\ (t)$, is divided into frames by applying a window and then the Short-Time Fourier Transform of each frame is calculated. For each spectral frame $X(\omega)$, multiplication of the compressed spectrum is calculated as:

$$Y(\omega) = \prod_{r=1}^{R} |X(\omega r)|^2 \qquad (30)$$

$$\hat{Y} = \max_{\omega_i} \{Y(\omega_i)\} \qquad (31)$$

Here R is the compression factor, and ω is the frequency. The result of the multiplication, (ω), is used to search for a maximum value, $\hat{Y}$, as is shown in equation (31). The frequency location of $\hat{Y}$ is the possible fundamental frequency for that frame. This HPS analysis is applied to each frame separately (Figure 3.16).

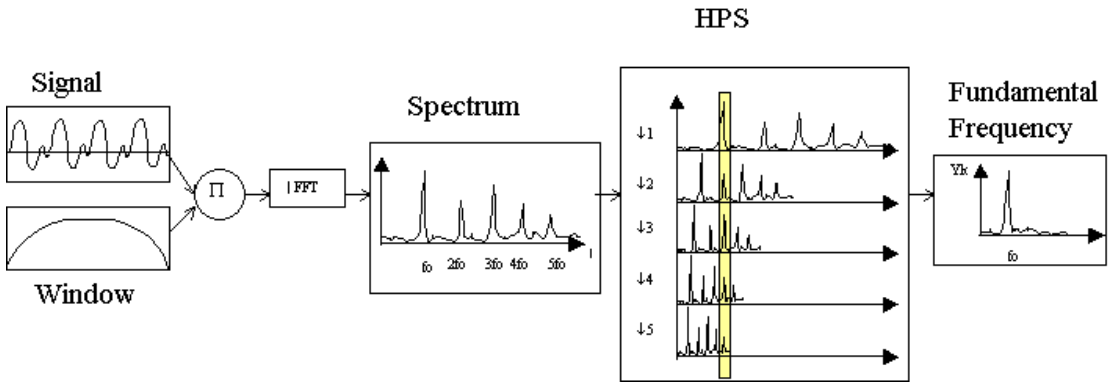


Figure 3.16: Overview of HPS algorithm taken from www. ccrma.stanford.edu

3.3.5.1 Implementation and Experimental Results

In our implementation, we have tested the performance of HPS analysis on a 5-second speech signal in the presence of restaurant ambient noise and music. For each noisy file, speech and noise signals have equal energy. We first divide the signal into 32 ms frames with an overlapping factor 50%. Each frame was multiplied with a Hanning window and transformed into the frequency domain by applying short-time Fourier Transform (STFT) with an FFT length 1400.

Figure 3.17 shows the performance HPS analysis on the clean speech signal. The figure on the top represents the spectrogram of a female speech signal while the figure on the bottom shows the estimated pitch by HPS analysis. From the spectrogram plot in Figure 3.17, we could say that the fundamental frequency of the female speaker varies between 200 Hz and 300Hz.

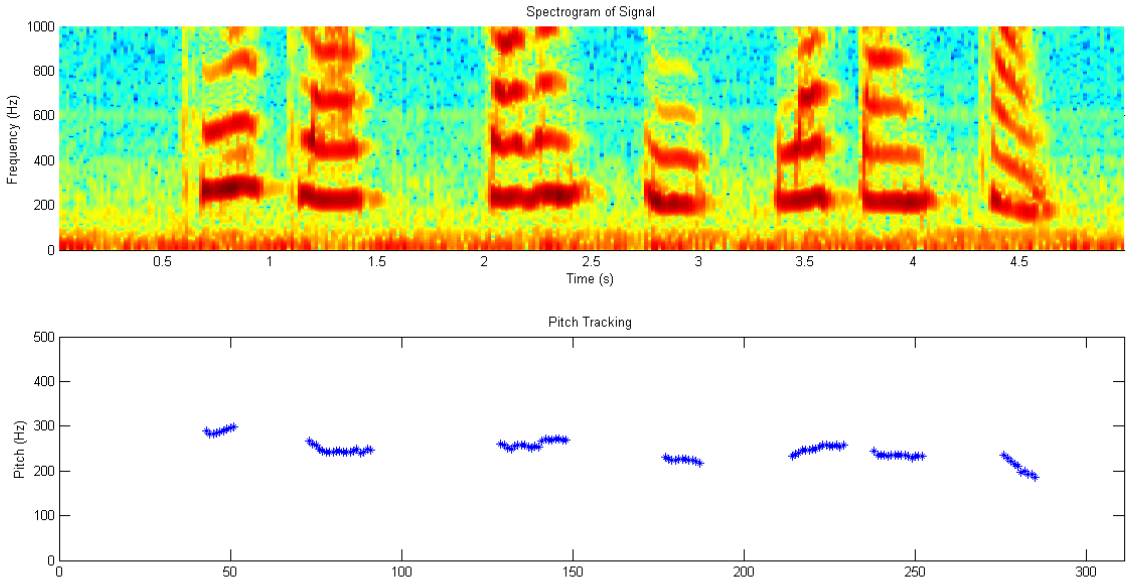


Figure 3.17: Spectrogram and the result of HPS analysis for clean speech.

If we see the pitch-tracking plot in the same figure, HPS analysis can track the pitch of the speaker correctly. The performance of HPS analysis in the presence of the diffuse noise is depicted in Figure 3.18. A recorded restaurant noise is used for diffuse noise signal. Before synthesizing the speech signal and diffuse noise signal, each pair of the signal was relatively normalized so that the ratio of their energies is 0 dB. The pitch of the speaker can be correctly estimated by HPS analysis when the energy ratio is 0 dB. The loudness of the diffuse noise begins to exceed the speaker by 10 dB, the efficiency degrades.

Figure 3.19 Performance of the HPS analysis in the presence of directional noise. A music signal is used for directional noise. The energy ratio between the speech signal and directional noise is 0 dB. Comparing to the estimated pitch in Figure 3.17, HPS analysis could still estimate the pitch of the speaker accurately in the presence of directional noise.

However, the energy ratio between the speech signal and directional noise signal decrease less than -10 dB, the performance of HPS starts to degrade. We could say from this experiment, HPS analysis is less sensitive to the additive noise.

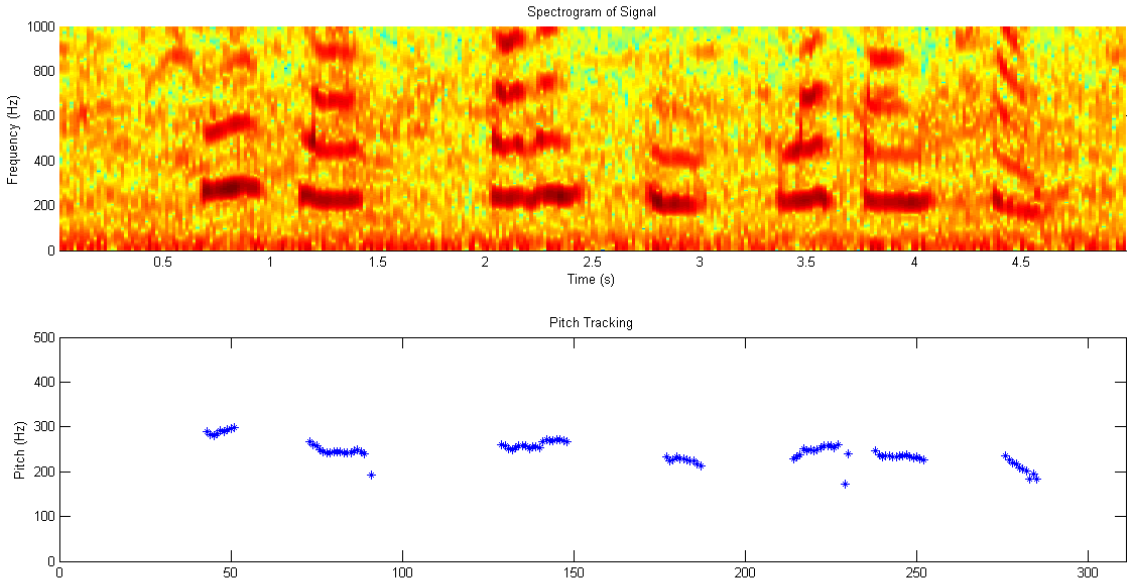


Figure 3.18: Spectrogram and the result of HPS analysis for speech contaminated with diffuse noise.

HPS analysis does not work well when there are two speakers' speech in audio file due to the fact that we cannot obtain good time resolution and frequency resolution at the same time. There is a tradeoff between time resolution and frequency resolution. In order to capture the variation in the pitch, we need to use relatively short window such as 20ms. However, a short window does not give enough frequency resolution to separate the fundamental frequencies or the harmonics of the fundamental frequencies, which HPS depends upon.

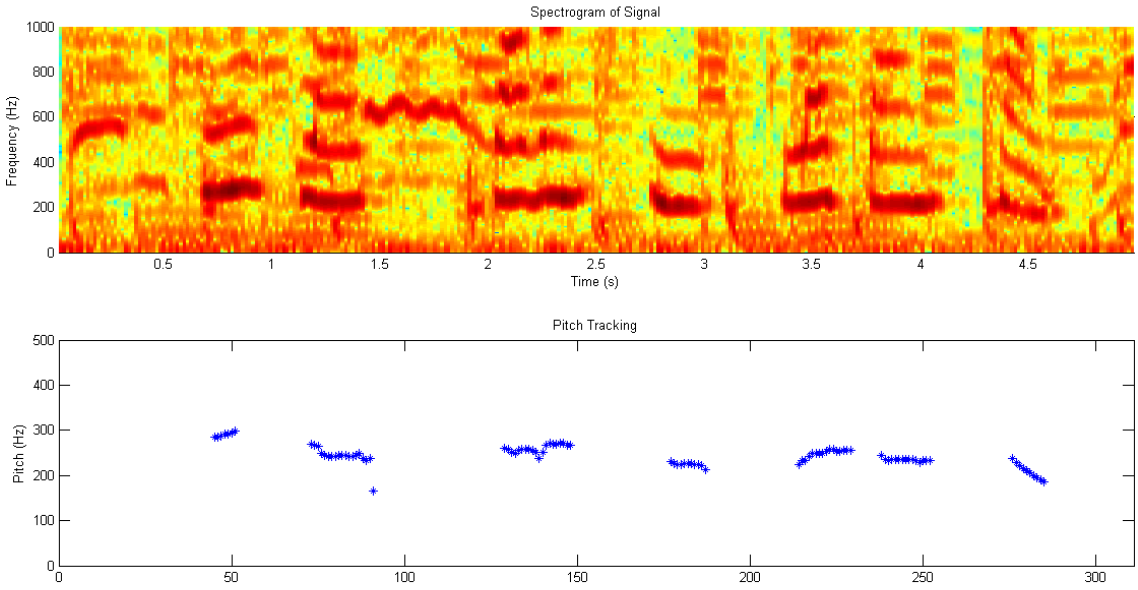


Figure 3.19: Spectrogram and the result of HPS analysis for speech contaminated with directional noise.

3.4 Chapter Summary

In this chapter, we described the limitations and challenges of two state-of-the-art multi-pitch tracking algorithms that are capable of identifying the presence of two simultaneous voiced speakers and yield their pitch estimates. These algorithms can also give a single pitch estimate in case there is only voiced source in the input signal. The algorithm depends on a feature called the 2-D AMDF, which is an extension of the traditional AMDF used in pitch detection algorithms and MP tracker, which utilized the sinusoidal modeling for pitch tracking. We have tested these algorithms with diffuse and directional noise environment to see the behavior of these algorithms in the real-world environment. We conclude that 2-D AMDF and MP Tracker algorithms are susceptible to the effects of background noise Also algorithms are prone to track the pitch of the simultaneous

speakers in the violation of the limitations. Besides than the limitations and the presence of background noise source, these algorithms fail in case of different situations such as inappropriate parameter selection, the speed of speaker ‘s talking, and different gender speech signals.

In this thesis, we established an approach that uses a combination of existing speech enhancement, speech separation, and pitch detection techniques in conjunction with the *early abductive reasoning* to obtain multiple pitch tracks in unstructured noisy environments. We designed an algorithm to take advantage of the combination of various techniques, and thus make it reliable for real-world conditions. Our early abductive reasoning approach is described in detail in the next chapter.

Chapter 4: Early Abductive Reasoning Approach

In this chapter, we provide a detailed explanation of our approach to early abductive reasoning for the multi-pitch tracking problem. In chapter 3, we have discussed the constraints and challenges of 2-D AMDF and MP Tracker algorithms. In this chapter, we discuss how these constraints and challenges on multi-pitch tracking algorithms can be utilized to search for valid pitch representations on the underlying signal efficiently. Furthermore, we discuss the appropriateness of a Blackboard (BB) framework (Nawab and Lesser, 1992) from Artificial Intelligence as the system architecture for this type of search process. Finally, we provide details on how we have incorporated early abductive reasoning approach into a Blackboard framework.

The use of constraints and challenges of 2-D AMDF and MP Tracker algorithms for multi-pitch tracking was elaborated upon in Chapter 3. In this chapter, we describe the details of how early abductive reasoning can use these constraints and challenging conditions to guide the detection and to reprocess incorrect regions in the initial pitch tracks of the simultaneous speakers. The BB framework as the system architecture for our approach and its appropriateness are discussed in Sec 4.1. We examine the advantages of modularity and incremental development that the BB framework offers. Furthermore, we also argue how the early abductive reasoning approach can be implemented through the specialized mechanisms (IPUS) (Lesser et al., 1995) on BB system. This BB mechanism fulfills the need for interaction between constraint matching and signal reprocessing in our approach. Specific details on how the BB framework is used to support our approach

are given in Sec 4.2. We elaborate on the various data representations that we have utilized within early abductive reasoning process to facilitate constraint matching and signal reprocessing. Furthermore, we provide a detailed description of the algorithms that we have developed for abductive reasoning such as discrepancy detection and reprocessing planning.

4.1 Appropriateness of Blackboard Architecture

Abductive reasoning is a process in which incomplete or partial evidence is used to conjecture explanations for what gave rise to that evidence. In the signal separation problem, abductive reasoning is done in order to conjecture about input signal data from the behavior of signal processing algorithms, which can be changed by using different algorithms and various parameter settings. In our application domain that involves multi-speaker pitch tracking in unstructured audio environments, abductive reasoning is used to predict pitch track of each speaker despite the unpredictable dynamics of the environment. These unpredictable dynamics arise because the time-frequency components of speakers' speech may be interacting with each other and with the time-frequency components of other sounds that may be in the background. In our application, multiple signal processing algorithms with different parameter settings are required to apply to data through abductive reasoning in order to get the conjection for the input signals. In other words, multi-speaker pitch tracking in an unstructured environment requires a process in which multiple signal processing algorithms with different parameters are applied to data through abductive reasoning in order to get the best

explanation. In the context of our research, we refer to this type of process as early abductive reasoning because of the fact that the abductive reasoning process is utilized before the signal processing transformation is completed.

In our research, we developed and implemented a computational approach that uses a combination of existing speech enhancement, speech separation, and pitch detection techniques in conjunction with early abductive reasoning to obtain multiple pitch tracks in unstructured noisy environments. Since early abductive reasoning may be viewed as a data-adaptive process for selecting the most appropriate signal processing for blind signal separation, we decided to utilize well-known Artificial Intelligence (Lesser et al., 1995), (Mani, 1998) techniques for implementing early abductive reasoning. Specifically, we decided to utilize the so-called Blackboard architecture (Nawab and Lesser, 1992), (Lesser et al., 1995) from Artificial Intelligence (AI) to implement the required abductive reasoning processes. Blackboard systems are appropriate for problems where (1) data is represented at different levels of abstraction (such as waveform, spectrum…) and (2) the solution is constructed based on the selection of independent signal processing algorithms. Blackboard architecture helps to coordinate the activities of many different signal processing algorithms on data that is represented at different levels of abstraction. We selected to use Blackboard framework because our application domain has the following features:

- Multiple Signal Processing Algorithms (STFT, MVDR, AMDF…)
- Intermediate data that the system produces is at a different level of abstraction level.

The multi-speaker pitch-tracking problem requires a data-adaptive process, which has the ability to select the most appropriate signal processing. The BB systems provide a signal understanding at any given time that may need adaptive strategies according to available incomplete or partial evidence. BB architecture is suitable for our early abductive reasoning approach because it provides an iterative search process for evidence at each level in order to give the best explanation at the source level. For these reasons, we decided to utilize BB architecture to implement the abductive reasoning process.

4.2 Abductive Reasoning Process

Early Abductive reasoning is a process in which *BB system* interacts with *abductive reasoning loop* as shown in Figure 4.1. The abductive reasoning loop is iterative because the elimination of discrepancy requires a search over various plausible explanations. During each iteration, processing is carried out on signal representations that are stored at various levels of abstraction in a BB database. Algorithms for constraint matching and signal reprocessing, which are called modules, operate on these data abstractions. The modules utilized in the abductive reasoning loop are of four types: *discrepancy detection, discrepancy diagnosis, reprocessing planning and reprocessing*. In Figure 4.2, we show how these modules are invoked during the abductive reasoning loop. The *discrepancy detection* module identifies discrepancies in the pitch tracking from the results of initial multi-pitch tracking algorithms. Discrepancies identified during discrepancy detection are analyzed by a *discrepancy diagnosis* module. This module generates plausible explanations for the cause of each discrepancy. The generation of these explanations may

be viewed as the process of using one or more diagnosis to find the evidence for the explanation for this discrepancy.

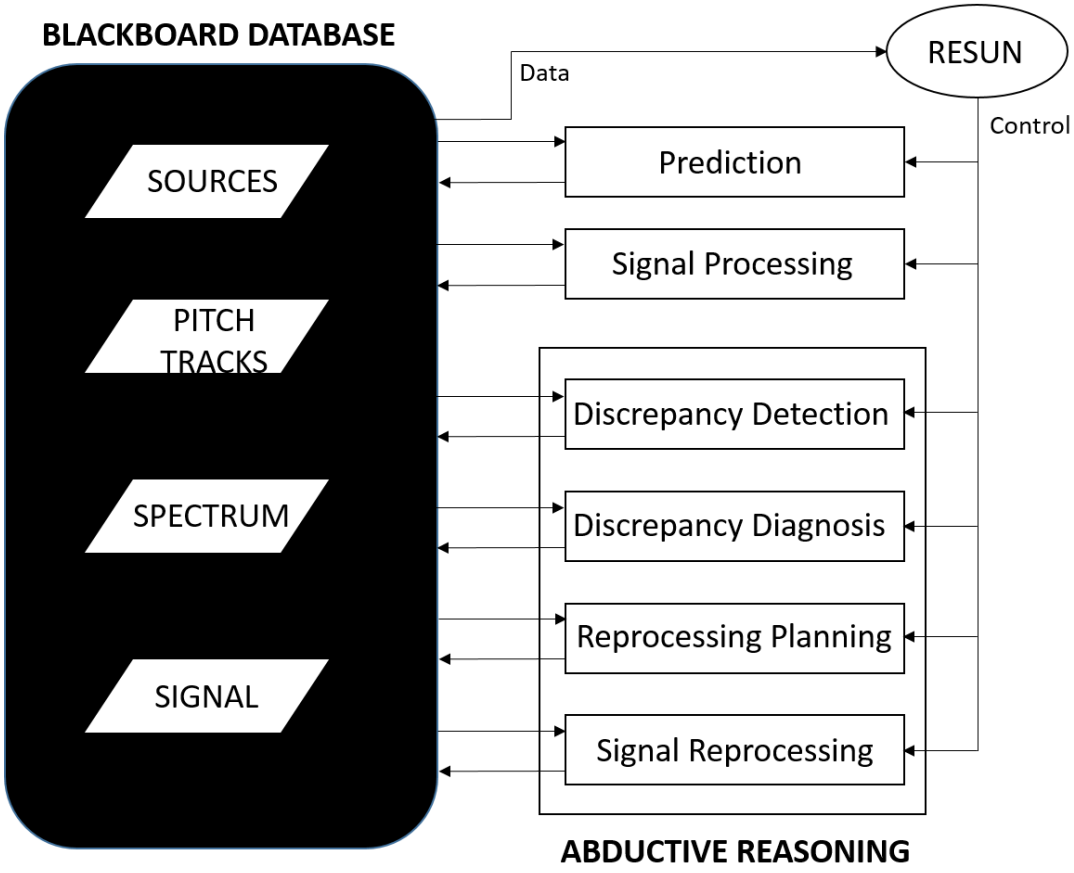


Figure 4.1: Generic Organizational Framework of EAR approach.
IPC: IPUS C++ PLATFORM
(Diagram taken from [Mani, 1999]

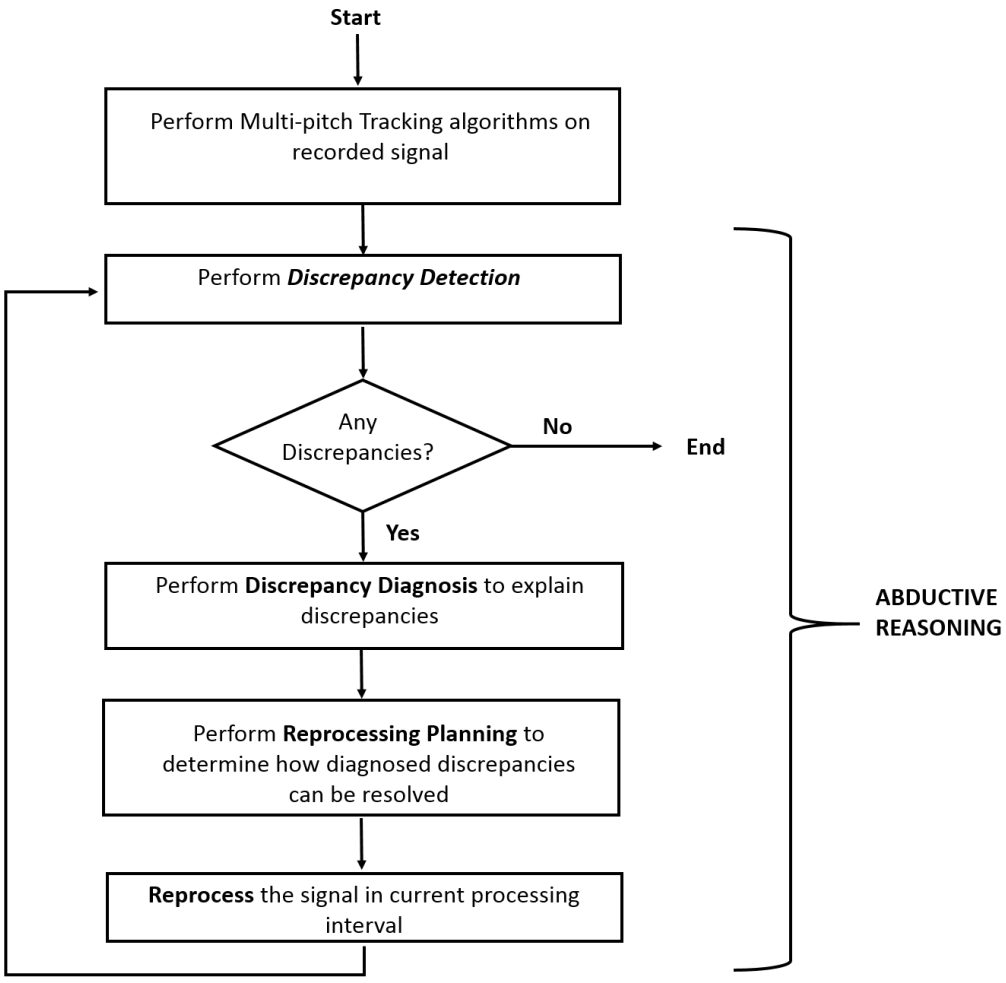


Figure 4.2: Abductive reasoning process

We illustrate an example for discrepancy diagnosis. Consider one pitch value detected from the multi-pitch tracking algorithm outputs at a time instant, while two pitch values detected around current time instant. Two pitch values indicate the presence of two speakers, while one pitch value shows the presence of a single speaker. This discrepancy can be explained by the lack of frequency resolution. That means the pitch values of two speakers are close to each other and the frequency resolution of the initial multi-speaker pitch-tracking algorithm is not enough to declare these two pitch values at

this time instant. One diagnosis for this example would be the use of higher frequency resolution. It should be noted that there may not be a unique diagnosis to explain a discrepancy and the diagnosis module is not guaranteed to produce the "correct" explanation in the first attempt. Multiple iterations of the abductive reasoning loop need to be performed to sift through the explanations and chose the correct one. For this reason, the discrepancy diagnosis module has knowledge of constraints as well as the properties of signal processing algorithms used in the application. Following diagnosis, a *reprocessing planning* module utilizes the diagnosis to hypothesize remedial reprocessing module that could be applied to the data at the different level of abstraction. This requires that the reprocessing planning module has an accurate knowledge of the behaviors of various signal-processing algorithms. We illustrate reprocessing planning using the example considered in the description of discrepancy diagnosis. In this example, a high-frequency resolution processing was identified to explain the discrepancy of two pitch values into a single pitch value. When the reprocessing planning module finds such a signal processing method, it hypothesizes that the signal should be reprocessed with a window having longer window length. In certain situations, reprocessing planning cannot arrive at the best reprocessing module in the first iteration of the abductive reasoning loop. Multiple iterations of the abductive reasoning are again necessary to ensure that the correct strategy is identified. *Reprocessing* module is finally applied to the signal data to try and to remove the discrepancies. Once re-processing has been performed, the abductive reasoning loop is repeated to identify and resolve discrepancies that may persist. Persisting discrepancies result from incorrect diagnosis or the reprocessing

planning failing to specify proper signal processing. Repetitions of the loop are carried out until all diagnosis is performed to remove the discrepancy we called early abductive reasoning process. If no such explanation is found, initial multi-speaker pitch tracking result is accepted. We called early abductive reasoning process.

4.3 Early Abductive Reasoning Process

We now demonstrate a system realization of our early abductive reasoning approach. In Figure 4.3, we depict a flow diagram of our early abductive reasoning process with abductive reasoning modules of prediction, discrepancy detection, discrepancy diagnosis, reprocessing planning, and reprocessing. We begin by using the prediction and discrepancy detection mechanisms to obtain an initial pitch track of the signal and to identify sub-regions of this representation that need to be reprocessed.

The prediction mechanism is utilized to extract the pitch trajectories of speakers in the signal intervals using 2-D AMDF algorithm. There are two main reasons for choosing 2-D AMDF in the prediction mechanism. One of the reasons is that 2-D AMDF works explicitly in the Time-Frequency domain; because other algorithms, such as machine learning based algorithms do not lend themselves to abductive reasoning (Gerlach et al., 2014), (Wu et al, 2003) (Lin et al., 2014), (Jin and Wang, 2010). Another reason is that its performance on clean speech is comparable that of other existing algorithms (Abhijith et al., 2014), (Wohlmayr et al., 2011), (Gerlach et al., 2014), (Wu et al., 2003), (Jin and Wang, 2010), (Lin et al., 2014) and is better than the other Time-Frequency domain algorithm, MP Tracker (Radfar et al., 2011) . The pitch trajectories

obtained from the resulting 2-D AMDF are compared against the predictions by using the discrepancy detection mechanism. Such a comparison helps identify sub-regions where the initial pitch representation needs to be reprocessed.

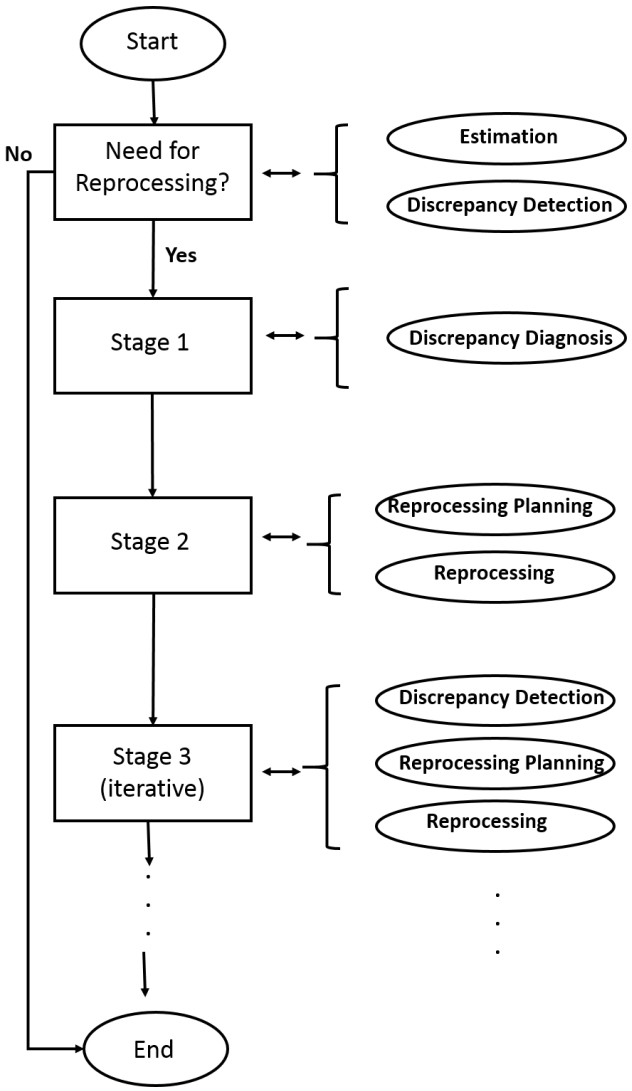


Figure 4.3: Depiction of the flowchart of Early Abductive Reasoning approach.

The discrepancy diagnosis mechanism is utilized to facilitate the processing of discrepant regions using the first stage of our early abductive reasoning approach. In

particular, this mechanism helps to identify the possible pitch values in the current processing frame. Reprocessing planning and reprocessing mechanisms are utilized to carry out the second stage of our early abductive reasoning approach on discrepant regions. The reprocessing planning mechanism helps in designing appropriate signal processing methods and appropriate parameter based on the diagnosis. Reprocessing is then carried out using one or more than signal processing methods to reveal the explanation for the source signal. Finally, a combination of discrepancy detection, reprocessing planning and reprocessing are used to carry out the third stage of processing in our approach. The discrepancy detection mechanism checks pitch values obtained during reprocessing in the second stage. New discrepant regions are then targeted for reprocessing using adjusted signal processing methods. The reprocessing planning mechanism helps in adjusting the parameters and the appropriate signal processing methods for this discrepancy. Reprocessing is carried out using the signal processing methods, and parameters are decided on the discrepancy properties. The pitch values of reprocessed regions are again checked by the discrepancy detection mechanism to determine if pitch values are sufficiently extracted from these discrepant regions. The combination of discrepancy detection, reprocessing planning and reprocessing is iteratively applied until no discrepancies are found by discrepancy detection mechanism. This completes the early abductive reasoning analysis of the signal interval.

4.4 Abductive Reasoning (EAR) Modules

In this section, we describe the specific details on how the BB framework was used to support our approach. It should be noted that the processing in our EAR-based system is carried out on overlapping signal blocks. Each block is subjected to the three-stage T-F analysis approach described in the previous section. We explained how pitch values are represented in each block using multi-pitch algorithms in Sec 3.1. In Secs 4.4.1 through Sect. 4.4.3, we describe discrepancy detection, discrepancy diagnosis, reprocessing planning, and reprocessing modules that we have applied.

4.4.1 Discrepancy Detection Module

Discrepancy detection is the process of identifying the mismatches between the tracks formed from the result of initial estimation and the system expectations or system constraints. The discrepancy detection module first detects the discrepancies and then identifies them. During this process, the module compares the value of pitch tracks at a frame to the values of pitch tracks within its time vicinity. The discrepancies are defined by a number of conditions determined as a result of this comparison. To simplify the process of discrepancy detection, we perform "clustering" on the pitch tracks, which are in close temporal proximity in the current mismatched interval. The clustering simplifies the process of discrepancy detection; it also ensures that discrepancies in close temporal proximity are kept together. This greatly helps the diagnosis or explanation of these discrepancies because discrepancies, that are close to each other in the time-frequency domain often have a common cause. At first, clustering is performed to the data predicted

by 2-D AMDF at the initial multi-speaker pitch tracking where the pitch tracks are represented in the time-frequency domain. The general idea of clustering is to compare the pitch values of the current frame to the pitch values of the previous frame and try to detect and identify the discrepancy. The procedure that we adopt for discrepancy detection module is outlined in Figure 4.4.

Discrepancy detection module starts from the first frame and compares each frame to the previous frames in order to detect the discrepant frame. This module nominates a frame as a candidate discrepancy if a sudden stop, a sudden start, or a rapid change are seen in consecutive pitch values in the pitch trajectories. After the discrepant frame is detected, the module clusters four frames around the discrepant frame. Discrepancy detection module identifies the frames of each discrepant frame and declares the corresponding 5 frames as a discrepancy. Discrepant frames are detected in the case of sudden stop, sudden start, rapid change, and continuity of the same value on the data as in shown in Figure 4.4.

After each start and end frames are found independently, if any consecutive discrepancy durations are overlapped, the discrepancy detection module enlarges the discrepancy region and considers this region as a new discrepancy. The module continues to "enlarge" the cluster in this manner until it is unable to combine any new discrepancy.

Having finished clustering discrepant frames and detecting discrepancies, the discrepancy detection module identifies the discrepancy based on the distortion operations found in discrepancy regions. Distortion operators are found by comparing pitch values in each track at frames in the discrepancy regions. A summary of the

distortion operators utilized in our EAR-based solution is provided in Table 4.1. Discrepancy operator may be detected for two pitch tracks as well as one pitch track in discrepancy regions.

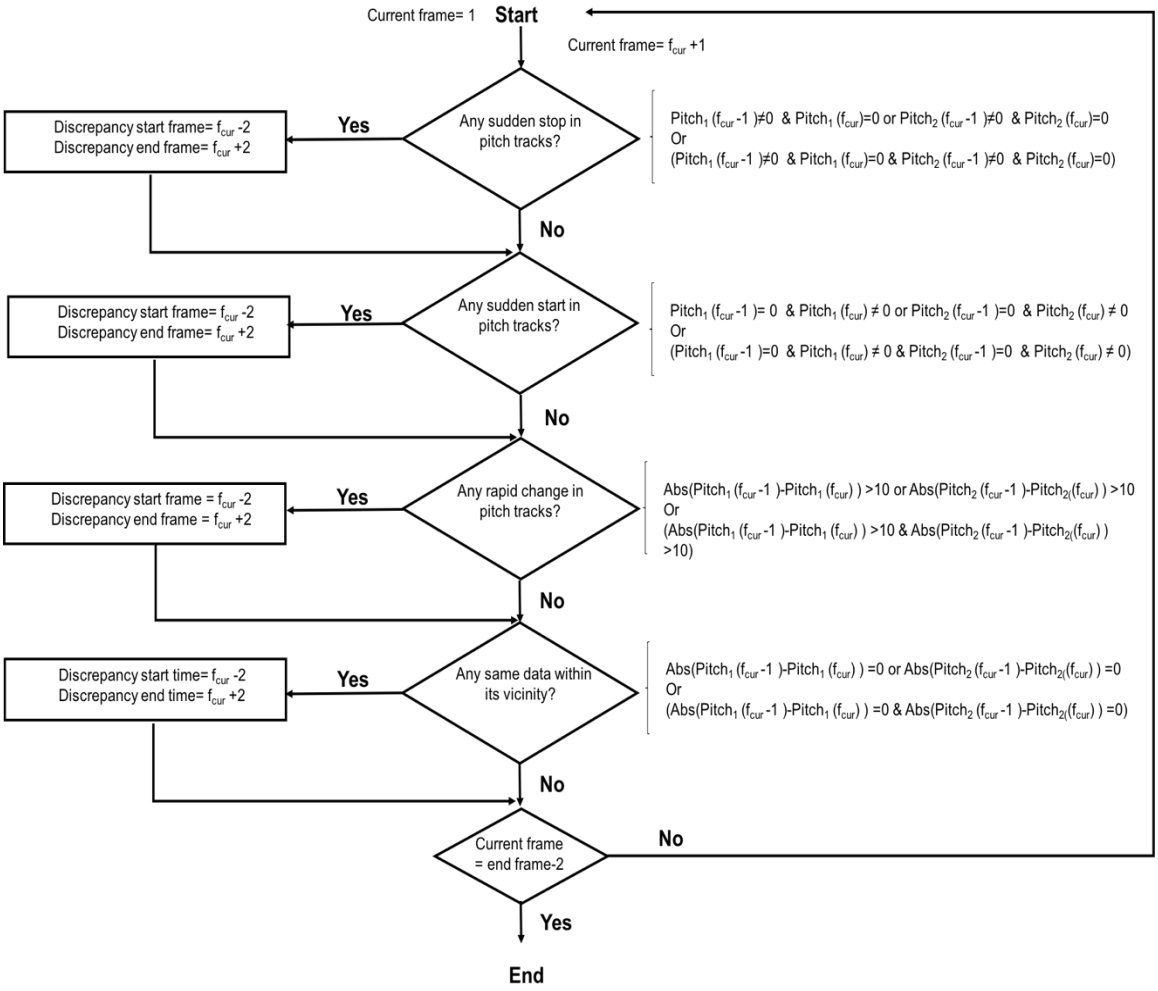


Figure 4.4: Procedure for discrepancy detection

Figure 4.5 shows some examples of discrepancies involving two speaker pitch tracks represented with blue and red lines. In part (a), the discrepancy detection module

detects that these two tracks are converging in the discrepancy regions and then the blue track stops suddenly. The highlighted area represents the discrepancy regions. CPT and EPT operators are captured respectively in the discrepancy region. Another discrepancy example is shown in Figure 4.5 part (b).

Distortion Indicators	**Description**
Start of a Pitch Track (SPT)	Indicates start of a new pitch
End of a Pitch Track (EPT)	Indicates termination of a pitch
Converging Pitch Tracks (CPT)	Indicates pitch values of individuals get closer at next frames
Diverging Pitch Track (DPT)	Indicates pitch values of individuals get further away at next frames
Rapid Increase in Pitch Tracks (RIPT)	Low Time Resolution (LTR) Indicates the window length used for processing is too long to accurately resolve the pitch values at a frame
Rapid Decrease in Pitch Tracks (RDPT)	Low Time Resolution (LTR) Indicates the window length used for processing is too long to accurately resolve the pitch values at a frame
Fluctuations in Pitch Track (FPT)	Indicates that speaker's speech is dominated by a diffuse noise
Incorrect Speaker Estimation (ISE)	Indicates that pitch values were picked with too small a threshold leading to incorrect speaker tracks
Constant Pitch Values in two pitch tracks (CPVT)	Indicates inadequate frequency resolution provided by filters or speaker's speech is dominated by a directional noise.

Table 4.1. Distortion Indicators and Descriptions

In part (b), the discrepancy detection module captures CPT, EPT, SPT, and DPT operators, respectively. But in this example, initially, two discrepant frames are detected. By clustering, two different discrepancy regions are identified separately in which CPT, EPT operators are detected in one region and SPT, DPT is detected in the second region.

Because of the overlapping of these regions, the module combines the discrepancy regions and declares a new discrepancy in new clustered region. Discrepancy detection module may capture one or more than one distortion operations in different order. The distortion operators and their description are listed in Table 4.1. For example, if there is rapid increase or decrease in the two consecutive pitch values of the same speaker, the discrepancy is captured by the operator (RIDPT) shown in Figure 4.5 part (c).

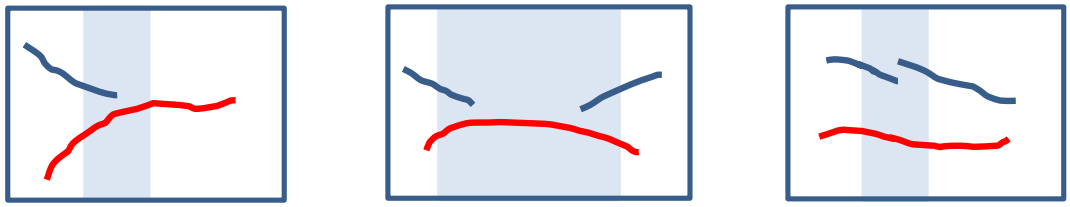


Figure 4.5: Examples of discrepancies

Based on the distortion captured in each discrepancy regions, we now name the discrepancies that will help discrepancy diagnosis module in next section. This is performed by using a lookup table that is indexed by one or a combination of distortion operators detected in the discrepancy regions. We show a lookup table in Table 4.2. Abbreviations listed in Table 4.2 are used to indicate the distortion operators. Subscriptions in the distortion operation indexes represent the pitch tracks number.

Distortion Indicators Found	*Discrepancy Name*
CPT_1, EPT_1	Missing prediction in an edge
CPT_1, EPT_1, SPT_1, and DPT_1	Missing prediction with converging one pitch track
CPT_1, EPT_1, SPT_1, and DPT_1 CPT_2, EPT_2, SPT_2, and DPT_2	Missing prediction with converging two pitch tracks
$RIPT_1$, $RDPT_1$	Fluctuations in a pitch track
$RIPT_1$, $RDPT_1$ $RIPT_2$, $RDPT_2$	Fluctuations in two pitch tracks
ISE	Incorrect speaker assignment
$RIPT_1$	Missing consistency in a pitch track
$RIPT_1$ $RIPT_2$	Missing consistency in two pitch tracks
SPT_1, DPT_1, CPT_1, and EPT_1	Missing prediction at two edges
EPT_1, SPT_1 EPT_2, SPT_2	Missing prediction
FPT_1	Contamination with diffuse noise source
CVPT	Contamination with directional noise source.

Table 4.2. Lookup table for identifying the discrepancies

Having finished identifying discrepancies, twelve different types of discrepancies detected by nine distortion operators are shown in Figure 4.6:

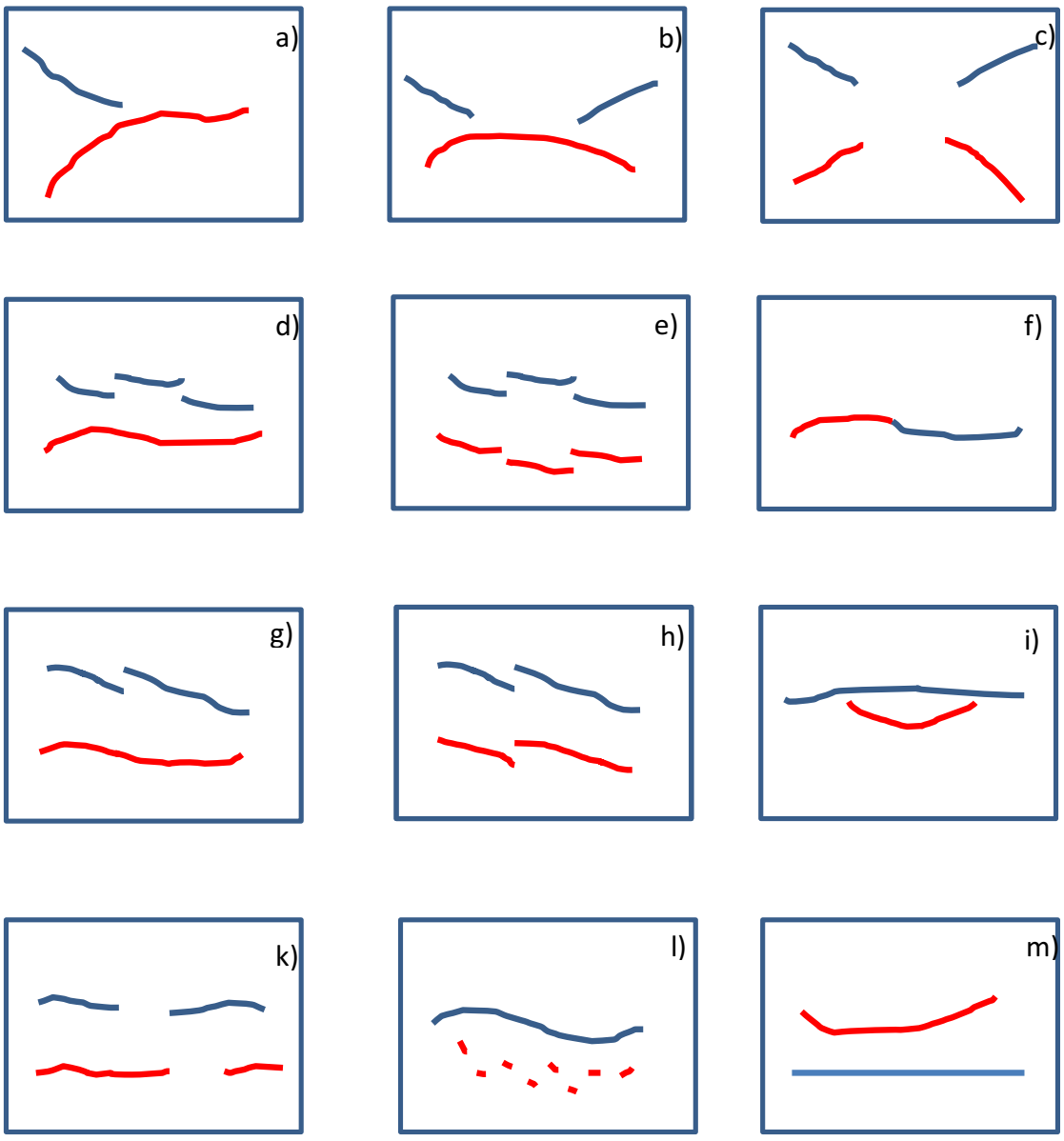


Figure 4.6: Types of discrepancies. (a) Missing prediction in an edge (b) Missing prediction with converging one pitch track (c) Missing prediction with converging two pitch tracks (d) Fluctuations in a pitch track (e) Fluctuations in two pitch tracks (f) Incorrect speaker assignment (g) Missing consistency in a pitch track (h) Missing consistency in two pitch tracks (j) Missing prediction at two edges (k) Missing prediction (l) Contamination with diffuse noise source (m) Contamination with directional noise source.

4.4.2 Discrepancy Diagnosis Module

The process of discrepancy diagnosis attributes probable causes to the discrepancy identified in the current processing interval. After discrepancies are detected, each discrepancy is diagnosed independently. At this point, this should be noted that the discrepancy diagnosis is performed after all discrepancies are detected. For each discrepancy, the diagnostic process involves the utilization of a set of "discrepancy operators" to provide a mapping between an "initial state" consisting of the pitch values within the discrepancy and a "goal state" representing actual pitch values in the discrepancy region. Discrepancy operators serve the two purposes of (a) identifying probable causes for each discrepancy, (b) hypothesize different possible situations based on the possible causes for each discrepancy. A summary of the distortion operators utilized in our EAR-based solution is provided in Table 4.3. These distortion operators also help to reprocess planning and reprocessing module in order to identify explanations for the discrepancies. This should be noted that there might be more than one distortion operators that cause one discrepancy. In the other words, one discrepancy can be associated with different distortion operators. The key idea used in diagnosis involves hypothesizing different possible explanation that may be actually happened in this discrepancy region. This requires knowledge about all possible discrepancy scenarios that can arise in the context of speech signals with two speakers. In the EAR-based algorithm, all possible scenarios are predefined according to the discrepancy type.

Distortion Operator	Description
Low-Frequency Resolution (LFR)	Indicates frequency separation between pitch values was smaller than frequency resolution
Low Time Resolution (LTR)	Indicates the window length used for processing is not enough to accurately estimate the pitch values
High Energy Difference (HED)	Indicates the energy of one speaker is not enough to estimate the pitch values belonging that speaker.
Fluctuations in One Pitch Track (FOPT)	Indicates that one speaker's speech is dominated by a directional noise
High Energy of Harmonics (HEH)	Indicates that the energy of the harmonics is higher than the energy of actual pitch.
Constant Pitch Values in One Pitch Track (CPVOPT)	Indicates that one speakers' speech is dominated by a strong directional noise source.

Table 4.3. Distortion Operators

We now perform an analysis that identifies all such scenarios for each discrepancy. Figure 4.7 depicts all possible scenarios involving two speaker pitch tracks but we may conclude with one or two pitch tracks in discrepancy regions. The individual scenarios are obtained according to twelve types of discrepancies. In each scenario shown in Figure 4.7, the lines indicate the pitch tracks that correspond to the speakers. The blue line corresponds to one of the speakers while the red one belongs to another speaker. The job of four-stage abductive reasoning process is to prune the list of 38 scenarios down to a few specific scenarios as the likely candidates for explaining the discrepancies. In the first step, which is shown in Table 4.4, the type of discrepancies is used to perform an initial pruning of the possible scenarios. It should be noted that the numbers shown in

Table 4.4 at the ends of the branches correspond to the scenario numbers used in Figure 4.7. In the next step, the distortion operators in diagnosis module are utilized for carrying out further narrowing down of the possible scenarios. For instance, if one side missing prediction is detected in a discrepancy region, then the scenarios are bounded to be 1, 2, 3, and 4. (See Figure 4.7) Randomly, one of the candidate diagnosis from the list of pruned scenarios is chosen in order to diagnose the discrepancy. For example, if LFR distortion operator is chosen within the discrepancy region, the possible scenarios are bounded to be 1, 2, and 3. Then, reprocessing planning module is designed accordingly to that operator. After reprocessing is completed, the algorithm utilized discrepancy detection module in order to seek for evidence to support that diagnosis. If negative evidence is found, the algorithm chooses a different diagnosis to explain the discrepancy. During subsequent diagnoses, different candidate scenarios may be chosen if discrepancies still remain.

Based on the chosen scenario, abductive reasoning process hypothesizes a set of possible distortion operators that could potentially cause the discrepancies found in discrepancy region. This is performed by using a lookup table (Table 4.4) that is indexed by the types of discrepancies. Abbreviations listed in Table 4.4 are used to indicate the distortion operators.

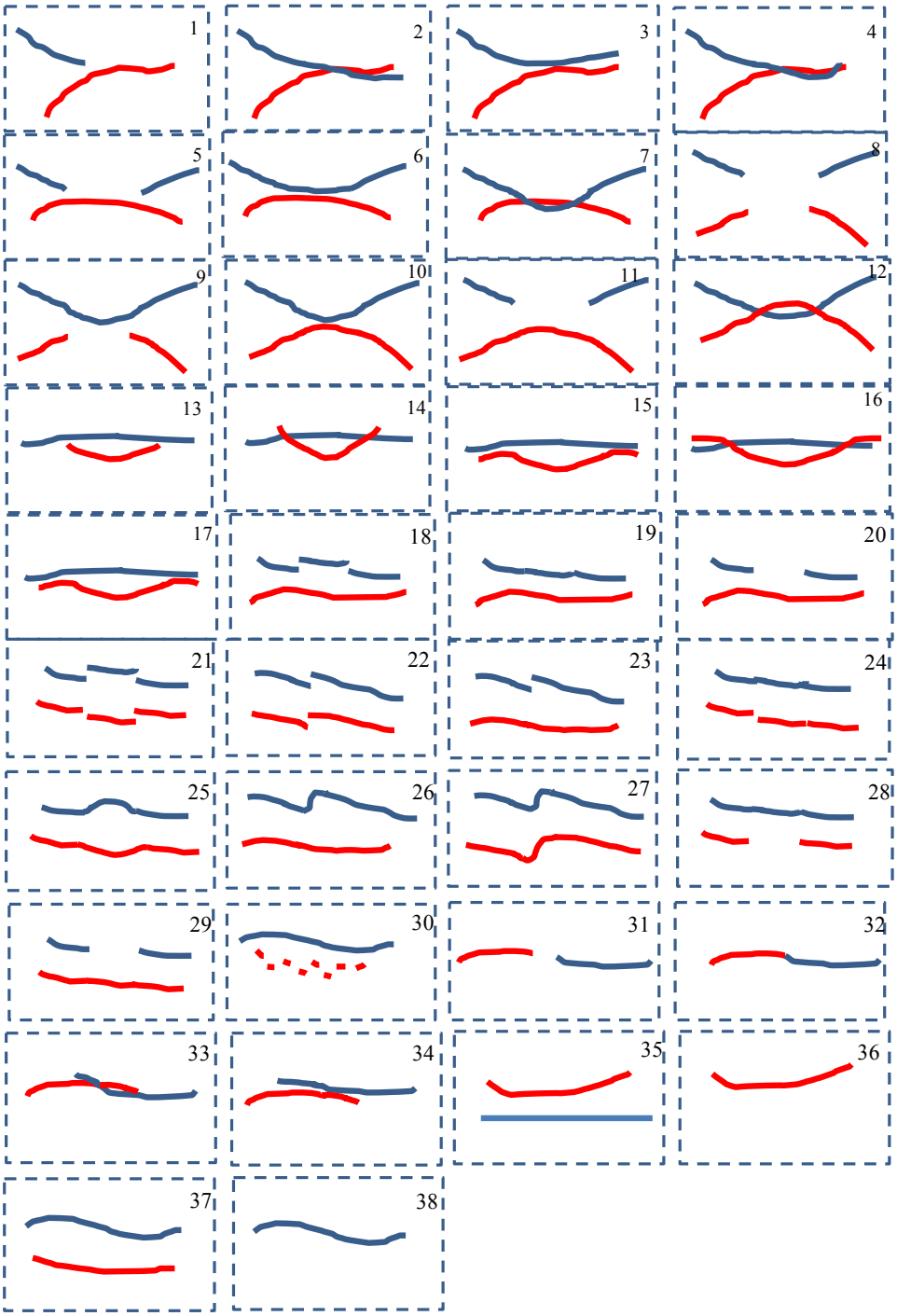


Figure 4.7: Possible scenarios

Based on Discrepancy Types		***Based on Distortion Operators***	
One side missing prediction	1-4	LFR	1, 2, 3
		HED	1
Missing prediction with converging one pitch track	5-7	LFR	6, 7
		HED	5
Missing prediction with converging two pitch tracks	8-12	LFR	8, 9, 10, 11,12
		HED	8, 9, 11
Fluctuations in a pitch track	18-20	LFR	18, 19
		HED	20
		HEH	18, 20
Fluctuations in two pitch tracks	18-21, 24-25, 28	LTR	18, 19, 21, 24, 25
		HED	20, 28
		HEH	18, 20, 21, 24, 28
Incorrect speaker assignment	31-34	LTR	33, 34
		HEH	31
		ISE	31, 32
Missing consistency in a pitch track	23-24, 26	LTR	23, 24, 26
		HEH	23, 24
Missing consistency in two pitch tracks	22-24, 26-27	LTR	22, 23, 24, 26, 27
		HEH	22, 23, 24
Missing prediction at two edges	13-17	LFR	13, 14, 15, 16, 17
		HED	13, 14
Missing prediction	28-29	HED	28, 29
		HEH	28,
Contamination with diffuse noise source	30, 37-38	HED	37, 38
		FOTP	30
Contamination with directional noise source.	35-37	HED	36, 37
		CPVOPT	35

Table 4.4. Abductive reasoning Progress

4.4.3 Reprocessing Planning and Reprocessing Module

The sequence of diagnostic operators associated with a discrepancy region is used to choose a suitable reprocessing plan from a library of plans. These plans result in a modification of the predictions within the region (if necessary) and a reprocessing of the data (if necessary) through a set of signal processing algorithms, and finally the tracking algorithm. The library of plans consists of the following three categories:

• *Plans that result in speech enhancement techniques:* These plans are executed when high energy difference, fluctuations in one pitch track, high energy difference, fluctuations in one pitch track, high energy of harmonics, and constant pitch values in one pitch track distortion operators are found in the pitch trajectories. The execution of these plans results in the design of speech enhancement techniques such as MVDR, coherence filtering techniques followed by the application of these techniques to the signal.

• *Plans that result in higher frequency resolution:* These plans are executed when low-frequency resolution distortion operator is found in the pitch trajectories. The execution of these plans results in suitable parameters of pitch tracking algorithms. The multi-pitch tracking algorithm tracker is then applied to predict the pitch values of speakers.

• *Plans that result only in high time resolution:* These plans are executed when low-time resolution distortion operator is detected in both of the pitch trajectories of two speakers. The execution of these plans results in suitable parameters of pitch tracking algorithms.

Once re-processing has been completed, the processes of discrepancy detection, diagnosis, and reprocessing are again carried out on the discrepancy region. This is

repeated until the discrepancy is devoid of diagnosis.

We perform an analysis, which explains the EAR-based process step by step and resolve a discrepancy. For example, Figure 4.8 depicted an example of discrepancy involving two speaker's pitch tracks. The type of discrepancy is identified in discrepancy detection module and declared as a missing prediction with converging two pitch tracks. By this type of discrepancy, diagnosis module concludes possible hypotheses that involved in this situation. As shown in Figure 4.8, these hypotheses are shown in dashed boxes and lead to categories of scenarios involving two speakers. The individual scenarios in this discrepancy are obtained by noting that there are two pitch tracks and a termination and a commencement of a speaker's speech within a pitch track. Different scenarios shown in Figure 4.8, the solid lines indicate predictions within the discrepancy region; the dotted lines indicate pitch tracks that correspond to the pitch track scenarios. It is now the job of the iterative diagnostic process to prune these possible scenarios down to specific scenarios as the likely candidates for expressing the true pitch track within the discrepancy interval. In the first step of this diagnostic process, one of the scenarios is chosen and then reprocessing planning, and reprocessing are performed to carry out a further narrowing down of the possible scenarios. In this example, the algorithm found some negative evidence and leaves the hypothesis. The negative evidence does not only help to leave the hypothesis also helps to narrow down the number of hypothesis for further processing. In this example, the algorithm estimated some pitch values that belong to speakers, then the number of hypotheses is bound to 2 (see Figure 4.8). Further processing is done based on the discrepancy and operator types.

In this example, a higher frequency resolution signal processing is performed by reprocessing algorithm for further processing. The evidence for the chosen diagnosis is found then the algorithm explains the discrepancy. Otherwise, another diagnosis is chosen to provide the explanation for the discrepancy. That should be noted that if the algorithm cannot find any evidence that will support any diagnosis, the algorithm will accept the initial multi-speaker pitch tracking result.

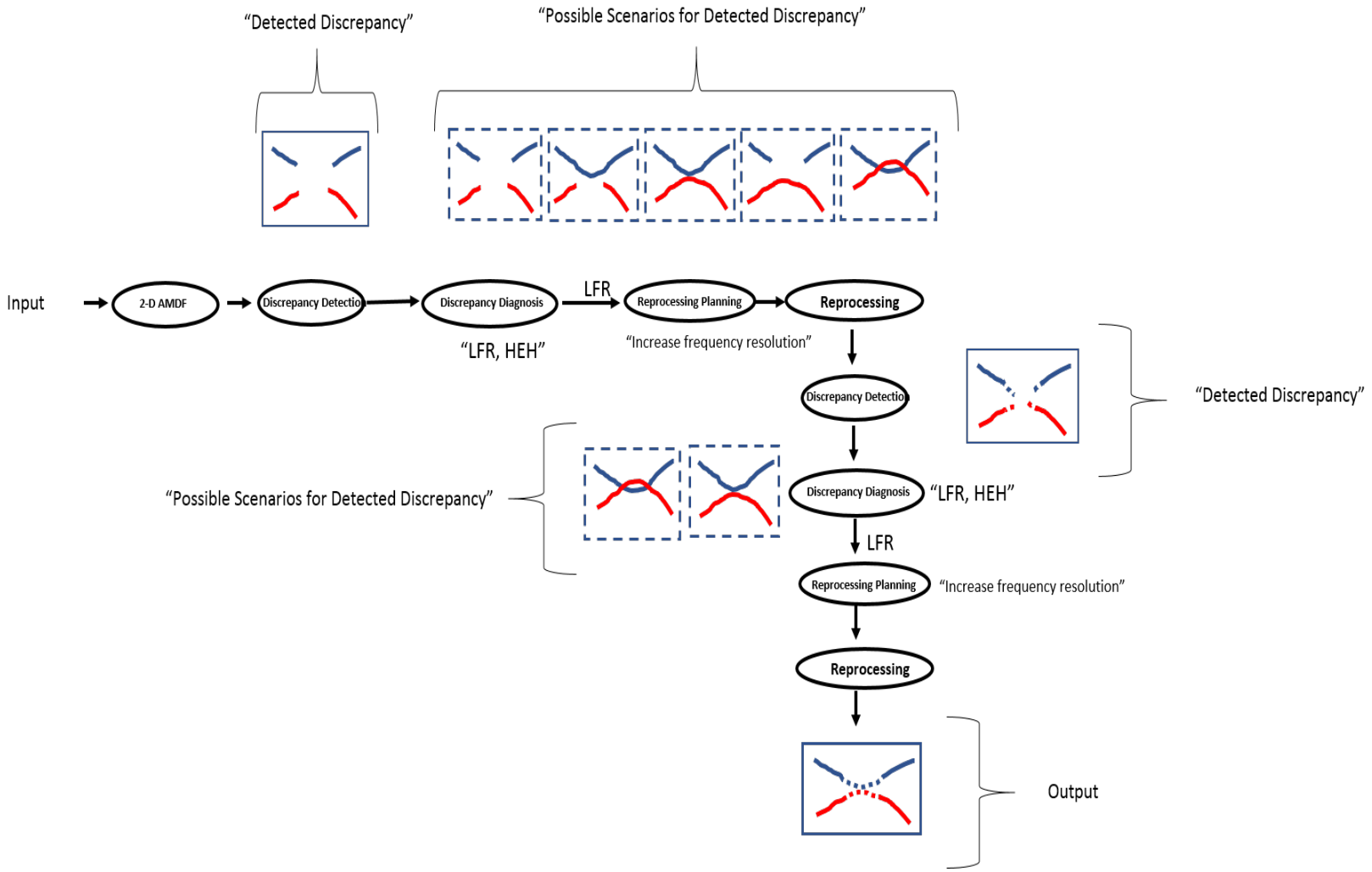


Figure 4.8: An example of EAR-based System

4.5 Chapter Summary

In this chapter, we described the details of how early abductive reasoning uses the BB framework as the system architecture and we explained the appropriateness of BB framework are discussed in Sec 4.1. We described the specific details of the EAR-based approach and how the BB framework was used to support our approach in Sec 4.2. We elaborated on the various data representations that we have utilized within the BB framework to facilitate constraint matching and signal reprocessing. We provided a detailed description of the algorithms that we have developed for BB mechanisms of discrepancy detection, discrepancy diagnosis, reprocessing planning and reprocessing in Sec 4.3 and Sec 4.4.

Chapter 5: Evaluation of EAR- Based Algorithm on Speech Mixtures

5.1 Introduction

We carried out some experimental validation to demonstrate the efficacy of EAR-based multi-pitch tracking. In the literature, there are some sets of standard criteria used to compare the performance of multi-pitch tracking algorithms quantitatively such as gross error rate (GER), separation error (SE). In this thesis, we used a second methodology which is the method of evaluating speech quality is through subjective listening tests. In this chapter, we report the performance of the algorithm on both the *Enhancement-based evaluation* and *Error-based evaluation*, using subjective and objective measures. We also compare the proposed algorithm to other algorithms in the literature, namely, 2-D AMDF and MP Tracker. Furthermore, we report the performance of the algorithms in the case of background noise and different energy levels.

In next section, we describe the databases used in the experiments.

5.1.1 Database

The database consists of 570 mixture signals created from 20 clean speech files (10 male and 10 female) from TIMIT database (Garofolo J. , et al., 1990). The database includes a mixture of speech signals and noisy mixture speech signals. Four different types of noises were used in the creation of this database: (1) restaurant noise, (2) street noise, (3) trumpet music, (4) pop music. We recorded stereo mixture speech signals by placing two speech sources in the direction of $\pm 45°$, and $\pm 90°$ where 0° is perpendicular to the line which combines the two microphones. For each direction, the two signals received at the

microphones placed 3.3cm apart, saved as the corpus of two-speaker files. Figure 5.2 depicts the microphones set up for recording scenarios. Three classes of mixture speech signals were created: different gender (FM), same gender (male, MM) and same gender (female, FF). For each class, 45 pairs of sentences with lengths closest to each other were identified, and 90 two-speaker signals are recorded on a pair of omnidirectional microphones (MM Series BSM-5-Micro-Binarual Stereo Microphone). A photograph of the dual microphones we used in this research is provided in Figure 5.1 These microphones were placed 3.3 cm apart from each other with an inter-microphone distance and we captured two- channel signals in the presence of sound activities in the audio environment. Signals from the microphones were collected by us with a Handy Recorder HN4 with a sampling rate 44.1kHz and ultimately transferred over to a computer for further processing.

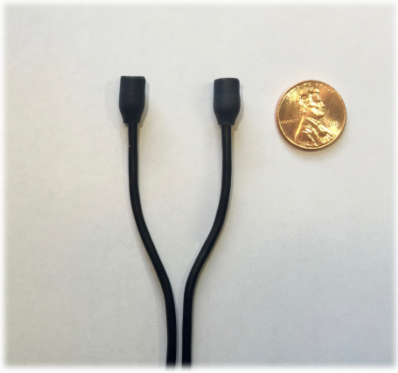


Figure 5.1: Omni-directional dual microphones used for recording our data.

Care was taken during this process that no speaker was the same on any pair and two source mixture signals were not placed in the same direction. Before recording each speech, signals were normalized to one regarding amplitude (max amplitude value is ± 1) and played simultaneously with the same type of speakers so that the ratio of their

energies was 0 dB, i.e., all signals were equally strong. These were then recorded in the different target to masker ratios (TMRs), ranging from 0 dB to 20 dB in steps of 5dB. This procedure gave a total of 270 mixture signals (90 for each class) for each of the 5 TMRs.

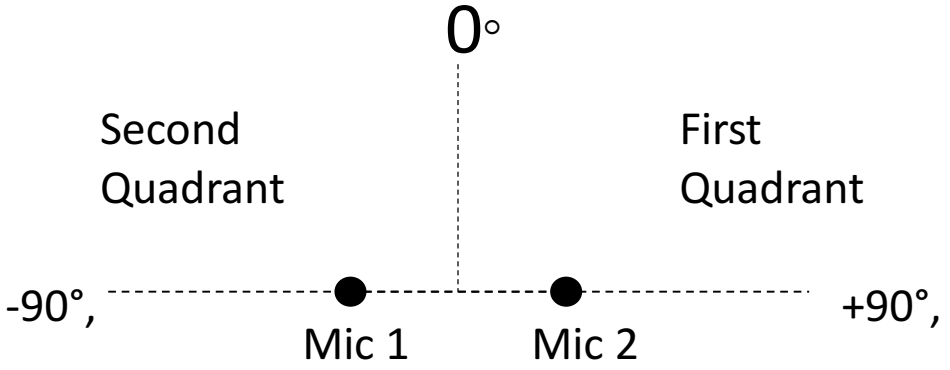


Figure 5.2: Placement of the two omnidirectional microphones and sound sources.

In a similar procedure, as above for the mixture speech signal database, each of the recorded noise files was added to each of the speech files in different signal to noise ratios (SNRs), ranging from -10 dB to 10dB in steps of 5dB. As a total 570 speech files were used for evaluating the objective and subjective scores of the signals processed by the proposed EAR-based multi-pitch algorithm and we compared the performance of our approach to 2-D AMDF and MP Tracker algorithms.

5.1.2 Error-Based Evaluation Methodology

Since many pitch tracking algorithms in the literature report their evaluations on the percentage of pitch values correctly estimated, we will use that criterion as a benchmark for comparison. The criteria used are namely gross error rate (GER) and separation error (SE). GER is defined in (Rabiner et al., 1976) and is the percentage of estimated pitch values varies by more than 10 Hz from true pitch values. Separation error (SE) is the false detection regarding the total percentage of the pitch frequencies of one speaker allocated mistakenly to another speaker (Frost, 1972). To compare the performance of three algorithms we calculate GER and SE, defined as follows:

$$GER = \frac{\#\ pitch\ values\ varies\ by\ more\ than\ 10\ Hz\ from\ true\ pitch}{\#\ pitch\ values} * 100 \tag{5.1}$$

$$SE = \frac{\#\ pitch\ values\ are\ mistakenly\ declared\ for\ speakers}{\#\ pitch\ values} * 100 \tag{5.2}$$

The ground truth pitch tracks of a speech mixture were obtained by computing pitch trajectories of the individual speech signals using Praat (Boersma & Weenink). We compared our system to the two recent models: (1) 2D-AMDF technique proposed by Vishnubhotla and Epsy-Wilson (Vishnubhotla and Epsy-Wilson, 2008) and (2) the one proposed by Radfar et al. (Radfar et al, 2011) called MP Tracker. We evaluated these three methods regarding GER and SE to compare the efficacy of the EAR-based algorithm and other two methods.

Table 5.1 demonstrates the performance of state-of-the-art multi-pitch algorithms and EAR-based approach multi-pitch tracking algorithm using these proposed measures GER and SE. The multi-pitch algorithms are tested over an entire database with 0dB TMR.

	MP Tracker		2-D AMDF		EAR-based	
	GER	***SE***	***GER***	***SE***	***GER***	***SE***
Two Speaker	25%	24%	15%	11%	5%	<1%
Two Speaker w/ Street Noise	27%	15%	18%	15%	6%	<1%
Two Speaker w/ Restaurant Noise	29%	19%	21%	16%	7%	<2%
Two Speaker w/ Trumpet Music	48%	27%	29%	18%	8%	<3%
Two Speaker w/ Pop Music	59%	31%	34%	19%	11%	<3%

Table 5.1. Comparison of the performance of the proposed algorithm with that of 2-D AMDF and MP Tracker regarding the criteria GER and SE when TMR is 0dB.

It can be seen that the values of GER and SE are in general lower for the proposed algorithm than for comparable multi-pitch algorithms. The performance of multi-pitch algorithms is lower regarding GER and SE in the non-noisy environment. In the presence of directional noise, which is pop music or trumpet music, the performance of the 2-D AMDF and MP Tracker algorithms decreased dramatically regarding GER and SE. In addition to that, the MP tracker algorithm has higher GER, and SE rates compare to 2-D AMDF.

The algorithm is also evaluated on the database consisting of recorded mixture of speech signals where the mixtures are combined in different TMRs. Experimental results regarding versus TMRs obtained from our approach, the 2-D AMDF algorithm, and the MP Tracker algorithm is reported speech signals of male-male, female-female, and male-female mixtures in Figure 5.3 to Figure 5.5.

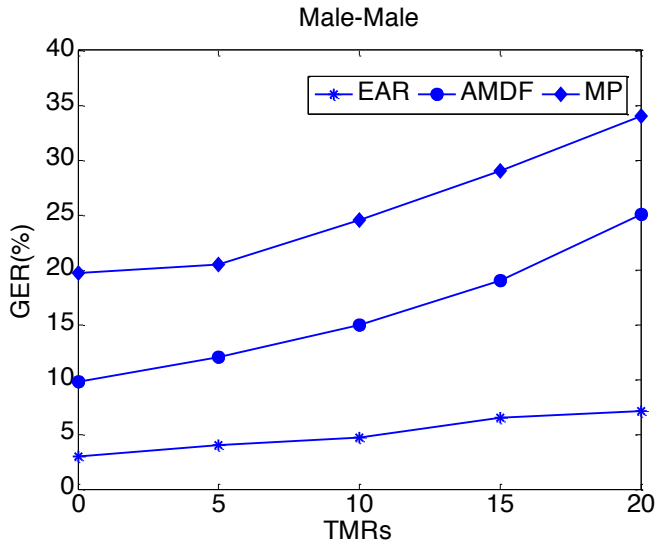


Figure 5.3: GER versus TMR for male-male mixture speech signals. The results are averaged over 90 mixture speech signals.

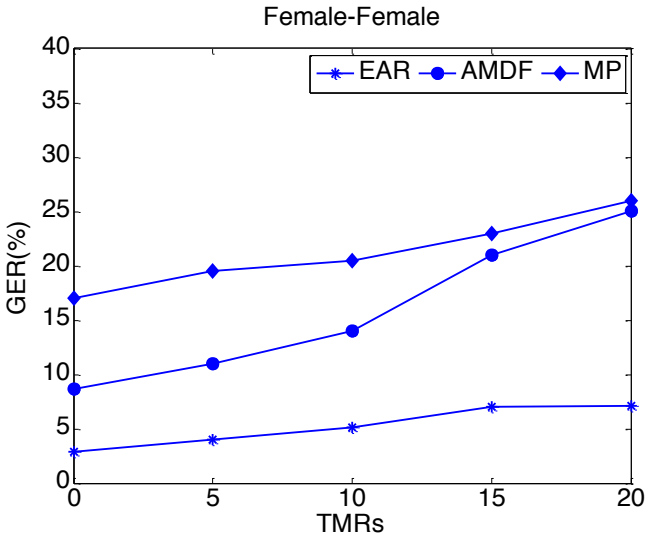


Figure 5.4: GER versus TMR for female-female mixture speech signals. The results are averaged over 90 mixture speech signals.

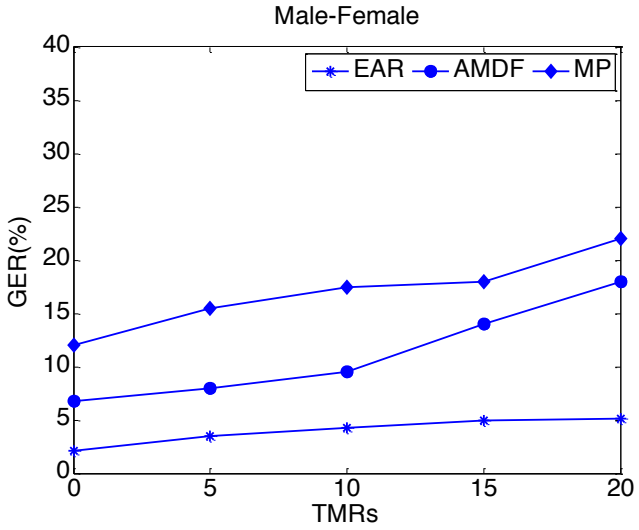


Figure 5.5: GER versus TMR for male-female mixture speech signals. The results are averaged over 90 mixture speech signals.

The results for each case are averaged over 90, 90, and 90 for male-male, female-female, male-female mixture signals, respectively. From

Figure 5.3, Figure 5.4 and Figure 5.5, we observed that the GER results obtained from the proposed approach are, on average, less than those of methods 2-D AMDF and MP Tracker, respectively for all types of mixtures. In addition, algorithm 2-D AMDF outperforms MP Tracker. The results show the EAR-based approach outperforms 2D-AMDF, on average, 11.1%, 10.72%, 7.2% for male-male, female-female, and male-female mixtures respectively. Also, EAR-based approach outperforms MP Tracker, on average, 20.4%, 15.98%, and 13% for male-male, female-female, and male-female mixtures respectively.

The separation error is shown in Figure 5.6 where we observed that our approach separates the pitch contours better than 2-D AMDF and MP Tracker. The averaged SE for our method, 2-D AMDF, and MP Tracker is 1.56%, 10.2%, and 22.6%, respectively.

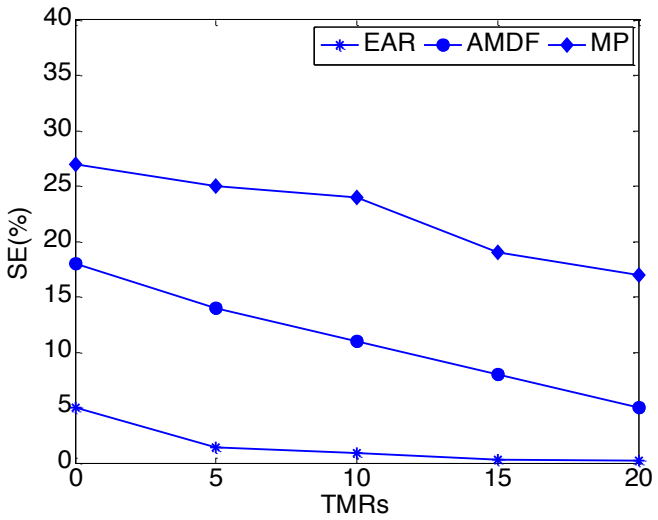


Figure 5.6: SE versus TMR for averaged over 270 mixture speech signals.

In order to evaluate the accuracy of three algorithms in a noisy environment, we calculated the GER and SE rates at various SNRs, defined as the ratio between the energy of the mixture speech signal to the energy of noise signal:

$$SNR = \frac{\sum_n speech\ signal[n]^2}{\sum_n noise\ signal[n]^2} \tag{5.3}$$

The SNR is calculated for the entire signal. Figure 5.7 shows the performance of three algorithms in the presence of directional noise in terms of GER and SE rates. It can be seen from the figures from both figures that even at very low SNRs, the EAR-based approach outperforms 2-D AMDF and MP Tracker. The EAR-based approach was able to estimate the pitch tracks with very low GER and SE rates. The GER and SE rates, on average, are shown in Figure 5.7 (a) and (b) are 6.78% and 1.62% respectively for the EAR-based approach. Similarly, the GER and SE rates for 2-D AMDF are 24.8% and 12% while the GER and SE rates are 29.2% and 19.3% for MP Tracker in the presence of directional noise.

Figure 5.8 shows the performance of three algorithms in the presence of diffuse noise. It can be seen from the Figure 5.8 the EAR-based approach outperforms 2-D AMDF and MP Tracker algorithms in terms of GER and SE. The GER and SE, on average, are shown in Figure 5.8 (a) and (b) are 5.3% and 1.2 % respectively for the EAR-based approach. Similarly, the GER and SE rates for 2-D AMDF are 20.4% and 11% while the GER and SE rates are 25.2% and 20% for MP Tracker in the presence of diffuse noise.

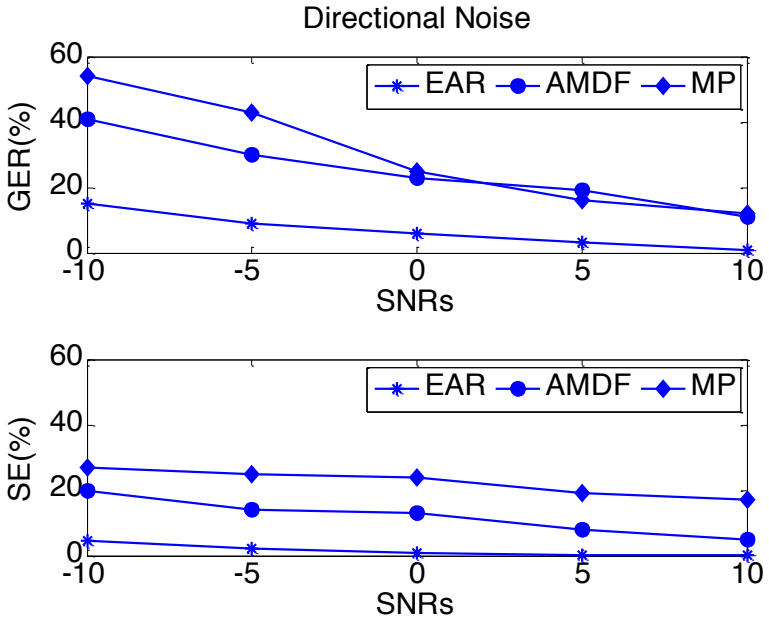


Figure 5.7: GER and SE versus SNR for averaged over 270 mixture speech signals in a directional noise environment.

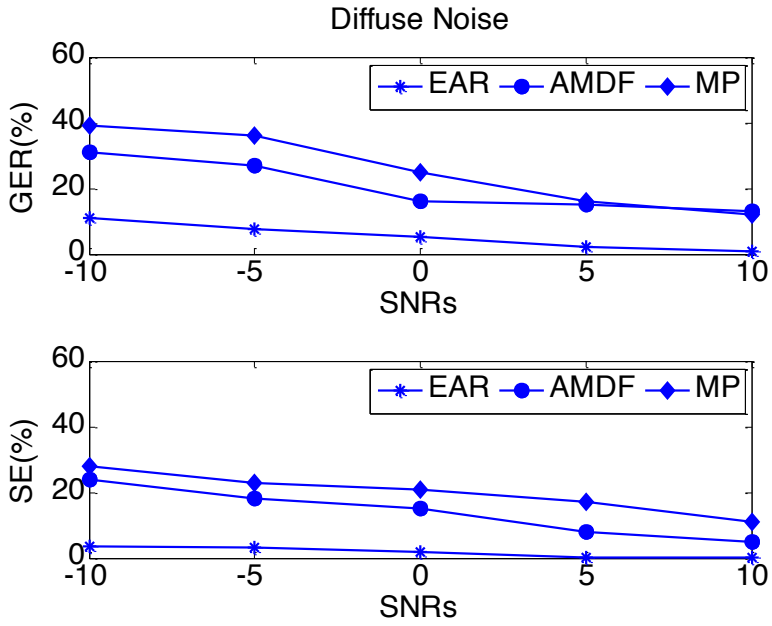


Figure 5.8: GER and SE versus SNR for averaged over 270 mixture speech signals in a diffuse noise environment.

In addition, we observed that all methods have less GER and SE rates in the presence of diffuse noise compare to the directional noise environment. GER rates in the diffuse noise environment are lower than the GER and SE rates in the presence of directional noise environment.

5.1.3 Enhancement-Based Evaluation Methodology

In this method, we used each extracted pitch track to enhance the speech of the corresponding speaker. The enhanced speech was then evaluated through listening tests to compare the quality of the enhanced speech to the speech in the original recordings. The enhancement was done by using harmonic filtering technique. The basic idea is to suppress the frequency components of the noise signal, which belong to the interference speech while preserving the fundamental frequency and its harmonics of the target speech. Enhanced speech was then evaluated through a subjective test to compare the quality of the enhanced speech to the speech in the original recordings. Subjective tests were designed according to ITU-T Recommendation P.835 methodology intended to evaluate the speech quality along three components: signal distortion, noise suppression, and overall quality. This method instructs the listener to attend to successively and rates the enhanced speech signal on (P.835, 2003):

1) The speech signal alone using a five-point scale of signal distortion (SIG)

5 - Very natural, no degradation
4- Fairly natural, little degradation
3-Somewhat natural, somewhat degraded
2-Fairly unnatural, fairly degraded
1- Very unnatural, very degraded

Table 5.2. Scale of signal distortion (SIG)

2) The background noise suppression alone using a five-point scale of background intrusiveness (BAK)

5 - Not noticeable
4- Somewhat noticeable
3- Noticeable but not intrusive
2- Fairly conspicuous, somewhat intrusive
1- Very conspicuous, very intrusive

Table 5.3. Scale of background intrusiveness (BAK)

(3) The overall effect using the scale of the Mean Opinion Score (OVRL) – [1 = bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent].

We presented the mean scores for the SIG, BAK, and OVRL scales for speech enhanced by three different multi-pitch algorithms evaluated in various types of background. The average scores for the noisy (unprocessed) speech files are also shown

for reference.

To reduce the length of the subjective evaluations, only a subset of the database was used for subjective evaluation. A total of 10 speech signals corrupted in two background noise environments (restaurant and music) at two levels of SNR (0dB and 5dB) were used. Four mixture signals were used from male-male, female-female speakers at 0dB TMR level. A total of 20 signals were evaluated by listeners and rated in terms of different aspect.

The process of rating the mixture signal and noisy mixture signal was designed to lead the listener to integrate the effects of both the signal and the background in making their ratings of overall quality. Each trial in a P.835 test involved a triad of files, where each sample consisted of two speaker signals or mixture speech signal synthesized with background noise signal. For each sample within the triad, listeners successively used one of the three five-point rating scales (SIG, BAK, and OVRL) to register their judgments of the quality of the test condition. In addition to the experimental conditions, each experiment included a number of reference conditions designed to independently vary the listener's SIG, BAK, and OVRL ratings over the entire five-point range of the rating scales. The P.835 standard permits the use of triads made up of either three different samples, or the same sample repeated three times. For this experiment, the same sample was used three times in each triad. A total of 20 listeners were recruited for the listening tests. Listeners were between the ages of 20 and 35 years of age. No listener participated in more than one experiment. The processed speech material was presented to listeners seated at separate. Speech materials were presented via Samsung headphones.

Figure 5.9 shows the mean scores for the SIG, BAK, and OVRL scales for enhanced individual speaker speech by harmonic filtering using EAR-based approach multi-pitch tracking at 0dB TMR level. Figure 5.10 – Figure 5.13 shows the mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with restaurant and music noise for SNR level of 0dB and 5dB.

In terms of signal distortion (SIG), the EAR-based multi-pitch algorithm performed equally well for 0dB TMR and for most SNR conditions and two types of noise. As seen from Figure 5.10 through Figure 5.13, mean scores for SIG have higher scores compared to mean scores for BAK and OVRL because of the perception of the noise signal in the mixture signals. Listener state that the enhanced signals with background noise such as music and restaurant noise have more degradation on the speaker's speech compare to the enhanced signal without background noise.

Higher noise distortion (i.e., higher BAK scores) was observed with our proposed approach to the condition of music noise. One explanation for that is the music is quasi-periodic signal and has close pitch value with speaker's pitch value. In this case, the harmonic filtering is not able to remove the fundamental frequency and harmonics of music noise.

In terms of overall quality, there was a significant difference in overall quality when the signals are contaminated with music noise for SNR 0dB and 5dB. The maximum OVRL mean score was observed for the mixture of two speakers signal.

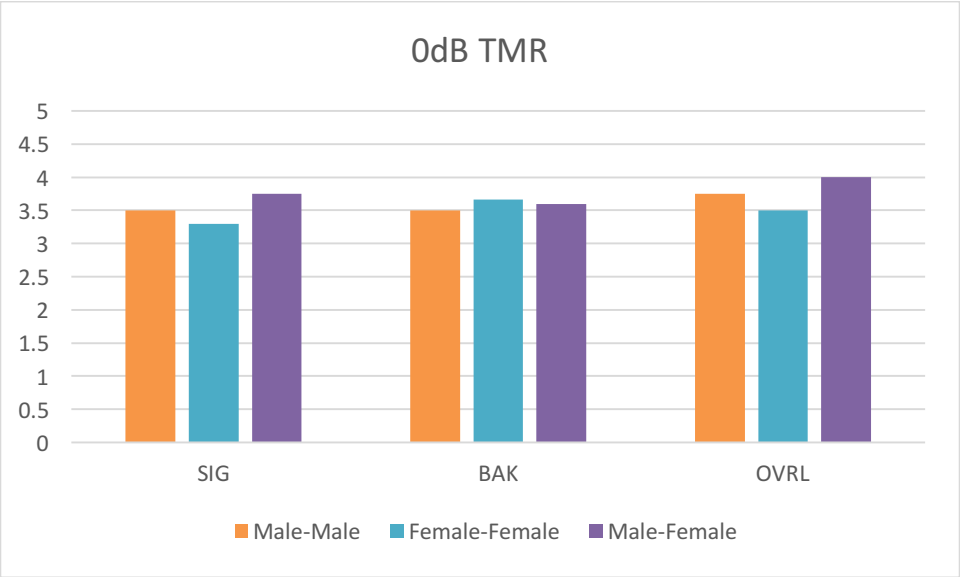


Figure 5.9: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals for TMR level of 0dB.

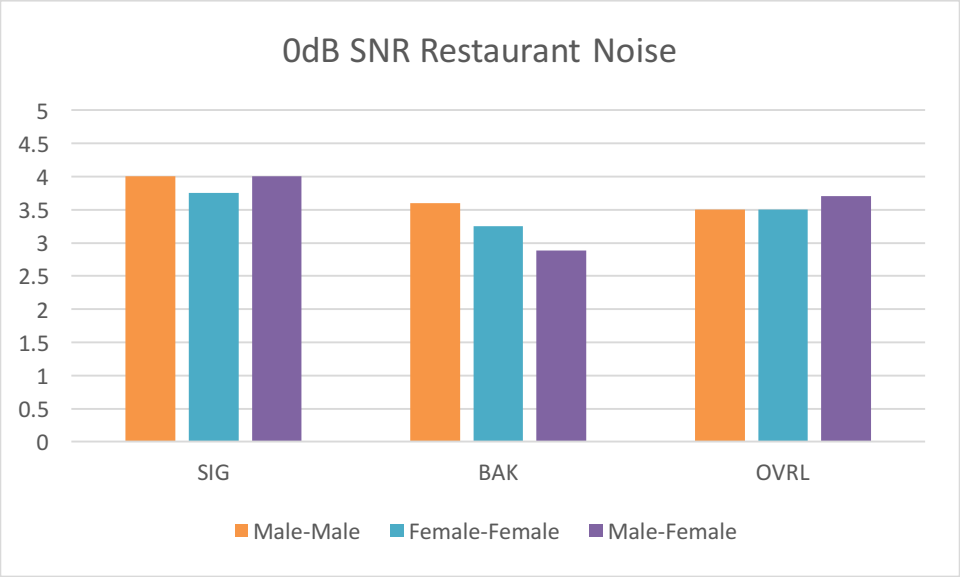


Figure 5.10: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with restaurant noise for SNR level of 0dB.

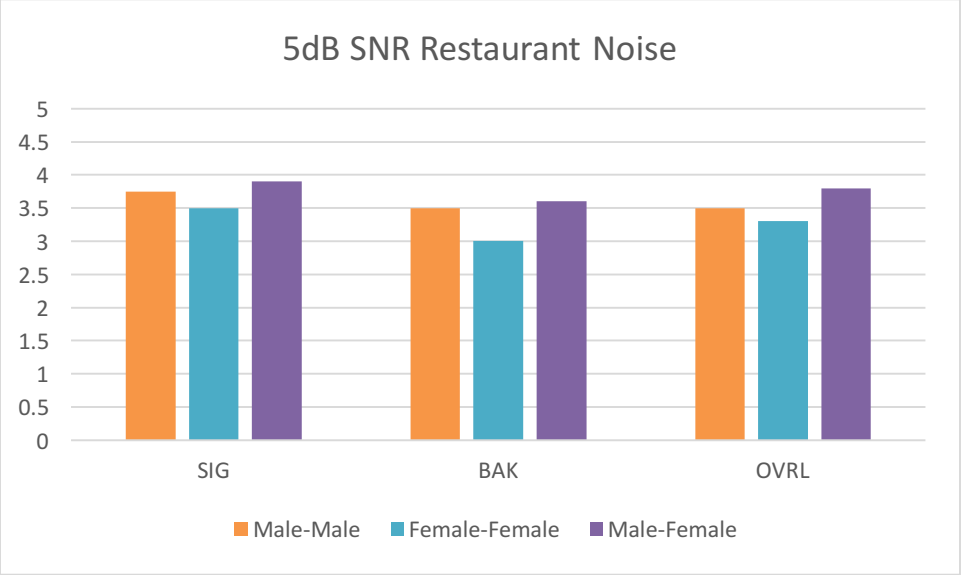


Figure 5.11: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with restaurant noise for SNR level of 5dB.

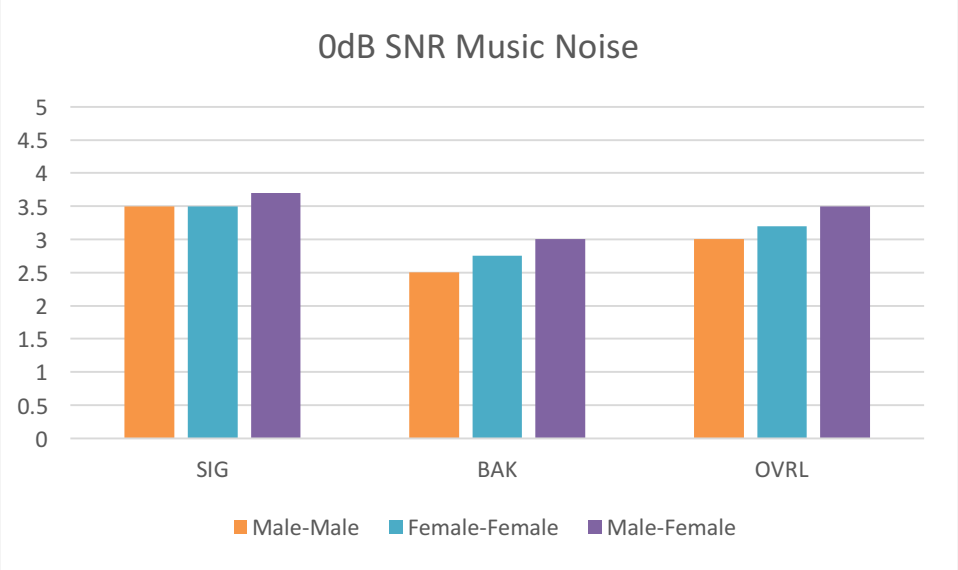


Figure 5.12: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with music noise for SNR level of 0dB.

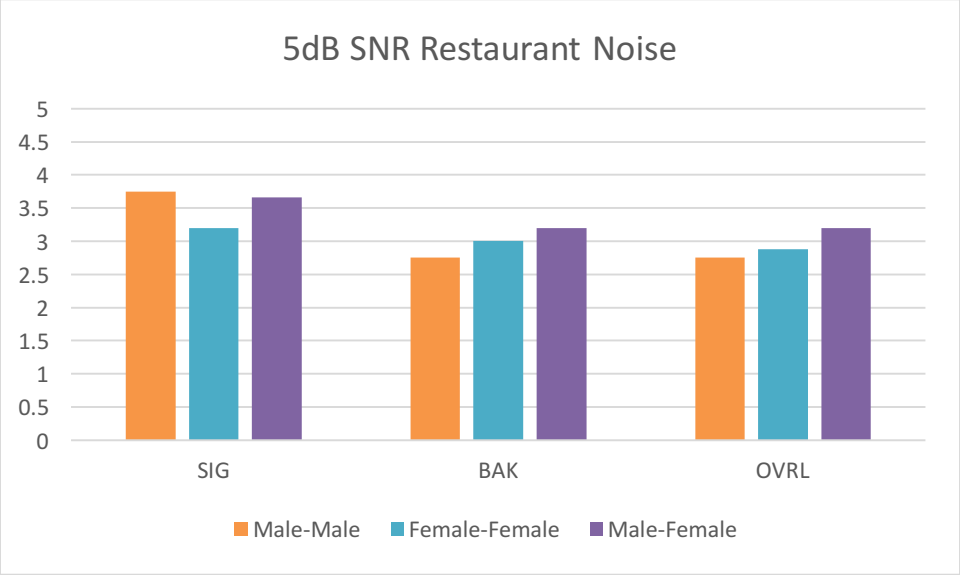


Figure 5.13: The mean scores for SIG, BAK, and OVRL scales for the EAR-based approach for multi-pitch tracking evaluated for two-speaker mixture signals contaminated with music noise for SNR level of 5dB

5.2 Chapter Summary

In this chapter, we have described the performance of the multi-pitch tracking using various objective error-based criteria and a subjective criterion. The proposed EAR-based approach for the multi-pitch algorithm is capable of estimating the pitch of overlapping speech in the unstructured audio environment. The utility of the estimated pitch resulting from the algorithm has been demonstrated on various performance evaluation criteria, showing improved performance compared to state of the art. We will next discuss some directions for future research.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In this thesis, we have introduced the Early Abductive Reasoning approach to a real-world problem involving highly unconstrained signals recorded with two omnidirectional microphones. Specifically, we developed and described an early abductive reasoning approach that adapts conventional signal processing methods to address multi-pitch tracking in unstructured multi-speech environments. This EAR-based multi-pitch tracking algorithm is meant to recover pitch contours from either noisy speech or speech mixtures. The algorithm has been developed with the aim of estimating pitch contours of individuals in the presence of any interference. Currently, the algorithm has been designed to estimate the pitch tracks of a maximum of two speakers speaking simultaneously.

In this thesis, we introduced the concept of "Early Abductive Reasoning," in which the abductive reasoning takes place before the signal processing is completed. We utilized abductive reasoning because working with signals recorded in an unstructured environment means that more than one signal processing transformation is needed in order to overcome dynamic changes in the background. Essentially, the abductive reasoning process provides the capability to drive a data-adaptive process for selecting the most appropriate signal processing transformation at any given time. By incorporating abductive reasoning earlier in the signal separation process, we made the signal processing more efficient and more effective.

The results of the research presented in this thesis demonstrate the power of

combining early abductive reasoning and the conventional signal processing methods to address a multi-speaker, pitch-tracking problem. We have successfully integrated early abductive reasoning signal processing into multi-speaker pitch tracking algorithms; this integration allowed us to overcome the complexities of unstructured noisy recordings. One of the main contributions of this thesis is the establishment of early abductive reasoning approach for tracking the pitch of two individuals that are speaking simultaneously in unstructured audio environments. Previous multi-pitch tracking algorithms are not robust to noisy environments.

Another contribution of the thesis is to establish the EAR-based algorithm as one that does not need to make any assumptions about the audio environment. Previously, the blackboard (BB) framework has been used to decompose of EMG signals and recognize movement disorders that involve real-world signal environments. The earliest applications of BB framework to the analysis of sound signals (Lesser et al., 1995) and music signals (Mani and Nawab, 1999) were limited to synthetic signals because at that time BB technology was not mature enough to deal with the complexities of real-world conditions. Beyond the contributions of this thesis mentioned above, this is the first use of the BB-based system tested on the real-world audio signal.

We have implemented multi-pitch tracking algorithms (Vishnubhotla and Epsy-Wilson, 2008) (Radfar et al., 2011) to compare the quantitative accuracy of the EAR-based algorithm. Experiments showed that the proposed approach outperforms state-of-the-art algorithms.

In Chapter 2, we formulated the multi-pitch tracking problem and investigated of the difficulties that must be overcome to create a robust solution to this problem. This is followed in Chapter3 by the signal processing methods utilized in developing our multi-pitch tracking solutions are discussed in Chapter 3. Next, in Chapter 4 we presented our multi-pitch tracking algorithm using the early abductive reasoning approach. The performance of two multi-pitch algorithms (2D-AMDF and MP Tracker) was compared in Chapter 5 to that of a solution based on the early abductive reasoning, in which different signal processing techniques were applied to build a system that can work in an unstructured audio environment.

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 the robustness of pitch 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 (< 5dB). 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 handling such situations include combining the proposed algorithm with pre-processing noise-suppression algorithms to improve the SNR for pitch detection and to obtain better pitch estimation.

• ***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 proposed in this thesis). As such, there is a need to better recover the unvoiced regions. In this region, pitch 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 multi-pitch tracking methods, possibly relying on models, which characterize unvoiced speech.

• ***Integrating Machine Learning to the EAR-based Algorithm:*** Future work on the system could be made by utilizing machine learning algorithms into the EAR-based approach. Currently, the EAR-based approach relies on the explicit design of discrepancy detection and diagnosis. In the case of more complex environments, the detection of discrepancies and the creation of diagnosis for discrepancies will necessarily become increasingly time-consuming and difficult. An implicit learning of discrepancies and diagnosis can be done through the use of machine learning such as neural networks

(NNs), support vector machines (SVMs), etc. The development of advanced signal understanding rules for discrepancy detection and discrepancy diagnosis will alleviate these difficulties arising from more complex environments.

• ***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 3-dimensional multi-pitch tracking algorithms to estimate three simultaneous pitch periods, and our preliminary studies have already shown promising results. The three-speaker case would also bring in additional problems regarding extra processing time, additional discrepancy types, as well as their solutions. Finally, the multi-pitch tracking problem, in that case, would become even 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.

The incorporation of these additional directions, combined with the intellectual contributions of this thesis, should result in both EAR-based solutions to complex pitch tracking problems as well as the development of a reliable pitch tracking for speech enhancement, speaker recognition and speaker identification in noisy environments.

References

Abdipour, R., Akbari, A., & Rahmani, M. (Oct. 2014). Two-Microphone Binary Mask Speech Enhancement: Application to Diffuse and Directional Noise Fields. *Electronics and Telecommunications Research Institute Journal, 36*(5), 772–782.

Abhijith, M. N., Ghosh, P. K., & Rajgopal, K. (2014). Multi-pitch tracking using Gaussian mixture model with time varying parameters and Grating Compression Transform. *Acoustics, Speech and Signal Processing (ICASSP).* Florence, Italy.

Allen, J. B., Berkley, D., & Blauert, J. (Oct. 1977). Multi- microphone signal processing technique to remove room reverberation from speech signals. *Journal of the Acoustical Society of America, 62*(4), 912–915.

Arabi, P., & Guannji, S. (2004). Phased-based dual microphone robust speech enhancement. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34*(4), 1763–1773.

Ba, H., Yang, N., & Ilker, D. (2012). BaNa: A Hybrid Approach for Noise Resilient Pitch. *In 2012 IEEE Statistical Signal Processing Workshop (SSP).*

Baer, T. (1979). Articulatory modeling and phonetics. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 27*(3), 299–300.

Boersma, P., & Weenink, D. (2016). *Praat:doing phonetics by computer*. Retrieved from http://www.fon.hum.uva.nl/praat/

Bokhoven, Y. H., & Van, W. M. (1991). "Co-channel speech separation using frequency bin nonlinear adaptive filter. *Acoustics, Speech and Signal Processing (ICASSP)*, 949–952.

Capon, J. (Aug.1969). High Resolution Frequency-Wavenumber spectrum Analysis. *Proceedings of the IEEE, 57*(8), 1408–1418.

Chakroun, R., Zouari, L. B., Frikha, M., & Hamida, A. (2015). A novel approach based on Support Vector Machines for automatic speaker identification. *IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. Marrakech, Morocco.

Chazan, D., Stettiner, Y., & Malah, D. (April 1993). Optimal multi-pitch estimation using the EM algorithm for co-channel speech separation. *Acoustics, Speech and Signal Processing (ICASSP)*, 728–731.

Cole, B. T. (2011). *Integrated machine learning and signal understanding for movement disorder recognition.* Doctoral dissertation – Boston University.

Cole, B. T., Roy, S. H., & Nawab, S. H. (2011). Detecting freezing-of-gate during unscripted and unconstrained activity. *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Boston, Massachusetts.

Cole, B. T., Roy, S. H., De Luca, C. J., & Nawab, S. H. (2010). Dynamic neural network detection of tremor and dyskinesia from wearable sensor data. *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (pp. 6062–6065). Buenos Aires, Argentina.

de Cheveigné, A. (June 1993). Separation of Concurrent Harmonic Sounds: Fundamental Frequency Estimation and a Timedomain Cancellation Model of Auditory Processing. *Journal of the Acoustical Society of America, 93*, 3271–3290.

de Cheveigné, A., & Kawahara, H. (2002). YIN, a Fundamental Frequency Estimator forSpeech and Music. *Journal of the Acoustical Society of America, 111*, 1917–1930.

De Luca, C. J., Adam, A., Wotiz, R. G., & Nawab, S. H. (2006). Decomposition of surface EMG signals. *Journal of Neurophysiology, 3*(96), 1646–1657.

Deshmukh, O., Epsy-Wilson, C., Salomon, A., & Singh, J. (Sept. 2005). Use of Temporal Information: Detection of the Periodicity and Aperiodicty Profile of Speech. *IEEE Transactions on Speech and Audio Processing, 13*(5), 776–786.

Ding, H., Qian, B., & Tang, Z. (Dec. 2006). A Method Combining LPC-Based Cepstrum and Harmonic Product Spectrum for Pitch Detection. *IEEE 2006 International Conference on Intelligent Information Hiding and Multimedia*, 537 – 540.

Dudgeon, D. (1977). Fundamentals of Digital Array Processing. *Proceedings of the IEEE , 65*(6), 898–904.

Even, J., Saruwatari, H., & Shikano, K. (2008). Frequency domain semi-blind signal separation: application to the rejection of internal noises. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference.* Las Vegas, NV, USA.

Frost, O. (Aug. 1972). An Algorithm for Linearly Constrained Adaptive Array Processing. *Proceedings of the IEEE, 60*, 926–935.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallet, D. S., Dahlgren, N., & Zue, V. (1993). Timit acoustic-phonetic continuous speech corpus. *Linguistic data consortium*. Available from https://catalog.ldc.upenn.edu/ldc93s1

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., & Dahlgren, N. (1990). TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. National Institute of Standards and Technology. National Institute of Standards and Technology .

Gerlach, S., Bitzer, J., Goetze, S., & Doclo, S. (2014). Joint estimation of pitch and direction of arrival: improving robustness and accuracy for multi-speaker scenarios. *EURASIP Journal on Audio, Speech, and Music Processing, 2014*(31).

Hess, W. (1983). *Pitch Determination of Speech Signals.* Berlin: Springer Verlag.

Hu, Y. C., & Loizou, P. (2008). Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(1), 229–238.

Ingale, P. P., & Nalbalwar, S. L. (2016). Cochannel Speech Segregation with Sparse Coding. *Electrical, Electronics, and Optimization Techniques (ICEEOT).* Chennai, India.

Jackson, P., & Shadle, C. (2000). Performance of the pitch-scaled harmonic filter and applications in speech analysis. *Proceedings. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00.* Istanbul, Turkey.

Jin, W., Liu, X., Scordilis, M. S., & Han, L. (2009). Speech Enhancement Using Harmonic Emphasis and Adaptive Comb Filtering. *IEEE Transactions on Audio, Speech, and Language Processing, 18*(2), 356 – 368.

Jin, Z., & Wang, D. (2010). HMM-Based Multipitch Tracking for Noisy and Reverberant Speech. *IEEE Transactions on Audio, Speech, and Language Processing, 19*(5), 1091–1102.

Jokic, I. D., Jokic, S. D., Delic, V. D., & Perie, Z. H. (2015). Mel-frequency cepstral coefficients as features for automatic speaker recognition. *Telecommunications Forum Telfor (TELFOR), 23rd.* Belgrade, Serbia.

Josephson, J. R., & Josephson, S. G. (1996). Abductive Inference. Cambridge University Press.

Kameoka, H., Nishimoto, T., & Sagayama, S. (Mar,2007). A multipitch analyzer based on harmonic temporal structured clustering. *IEEE Transactions on Audio, Speech, and Language Processing, 15*(3), 982–994.

Klassner, F., Lesser, L., & Nawab, S. H. (n.d.). *The IPUS Blackboard Architecture as a Framework for Computational Auditory Scene Analysis.*

Lane, J. E. (1990). Pitch detection using a tunable IIR filter. *Computer Music Journal, 14*(3), 46–57.

Lee, S. W., Soong, F. K., Ching, P. C., & Lee, T. (2008). *Chinese Spoken Language Processing, 2008. ISCSLP '08. 6th International Symposium.* Kunming, China,.

Lesser, V. R., Nawab, S. H., & Klasser, F. I. (1995). IPUS: An Architecture for the Integrated Processing and Understanding of Signals. *Artificial Intelligence, 77*, 129–171.

Lin, J., Zhang, G., Fu, B., & Hao, Y. (2014). Multipitch tracking with continuous correlation feature and hybrid DBNS/HMM model. *Wavelet Active Media Technology and Information Processing (ICCWAMTIP).* Chengdu, China.

Liu, Y., & Wang, D. (2016). Robust pitch tracking in noisy speech using speaker-dependent deep neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Shanghai, China.

Loizou, P. (2007). *Speech Enhancement: Theory and Practice.* CRC Press.

Mani, R. (1998). *Time-Frequency Signal Representation for Polyphonic Music.* Doctoral dissertation – Boston University.

Mani, R., & Nawab, S. H. (1999). Knowledge-based processing of multicomponent signals in a musical application. *Signal Processing, 74*, 47–69.

Maurya, A., & Aggarwal, R. K. (2016). Speaker recognition for noisy speech in telephonic channel. *In Proceedings of 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT).* Bangalore, India, India.

McLeod, P., & Wyvill, G. (2005). A smarter way to find pitch. *International Computer Music Conference (ICMC'05).*

Ming, J., Hazen, T. J., & Glass, J. R. (January 2010.). Combining missing-feature theory, speech enhancement, and speaker-dependent/-independent modeling for speech separation. *Computer Speech and Language, 24*, 67–76.

Moorer, J. A. (November 1977). On the transcription of musical sound by computer. *Computer Music Journal*, *1*(4), 32–38.

Mukai, R., Sawada, H., Araki, S., & Makino, S. (2006). Blind Source Separation of Many Signals in the Frequency Domain. *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.* Toulouse, France.

Nawab, S. H., & Cole, B. T. (2011). What is IPUS and how does it help resolve biosignal complexity? *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Boston, Mass.

Nawab, S. H., & Lesser, V. (1992). *Symbolic and Knowledge-Based Signal Processing.* Upper Saddle River, NJ: Prentice Hall.

Nawab, S. H., & Quatieri, T. F. (1988). Short-time Fourier transform. In *Advanced Topics in Signal Processing.* Englewood Cliffs, N.J.: Prentice Hall.

Nawab, S. H., Chang, S. S., & De Luca , C. J. (2010). High-yield decomposition of surface EMG signals. *Clinical Neurophysiology, 121*(10), 1602–1615.

Noll, A. (1967). Cepstrum pitch determination. *Journal of the Acoustical Society of America, 41*, 293–309.

Oppenheim, A. V., & Nawab, S. H. (1997). *Signals and Systems.* Upper Saddle River, NJ: Prentice Hall.

O'Shaughnessy, D. (2000). Coding of Speech Signals. In *Speech Communications: Human and Machine* (pp. 229–322). Wiley-IEEE Press eBook Chapters.

P.835, I.-T. (2003). *Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm.*

Pan, C., Chen, J., & Benesty, J. (Jan 2014). Performance Study of the MVDR Beamformer as a Function of the Source Incidence Angle. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*(1), 67–79.

Rabiner, L. R., & Schafer, R. W. (2010). *Theory and Application of Digital Speech Processing.* Upper Saddle River, NJ, USA: Prentice-Hall.

Rabiner, L., Cheng, M., Rosenberg, A., & McGonegal, C. (Oct 1976). A comparative performance study of several pitch detection algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 24*(5), 399–418.

Radfar, M. H., Dansereau, R. M., Chan, W. Y., & Wong, W. (May 2011). MPTRACKER: A New Multi-Pitch Detection and Separation Algorithm for Mixed Speech Signals. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4468 – 4471.

Rahmani, M., Akbari, A., Ayad, B., & Lithgow, B. (2009). Noise cross PSD estimation using phase information in diffuse noise field. *Signal Processing, 89*(5), 703–709.

Rao, V., & Rao, P. (September 1–4, 2008). Vocal Melody Detection In The Presence Of Pitched Accompaniment Using Harmonic Matching Methods. *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*. Espoo, Finland.

Ross, M. J., Shaffer, L. A., Cohen, A., & Freudberg, R. M. (Oct 1974). Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustics, Speech and Signal Processing, 22*(5), 353–362.

Schroeder, M. (1968). Period Histogram and Product Spectrum: New Method for Fundamental Frequency Measurement. *Journal of the Acoustical Society of America, 43*(4), 829–834.

Schwarz, A., Reindl, K., & Kellermann, W. (2012). A two-channel reverberation suppression scheme based on blind signal separation and wiener filtering. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference.* Kyoto, Japan.

Suk Lee, B., & Ellis, D. P. (2012). Noise robust pitch tracking by subband autocorrelation classification. Available from: https://www.ee.columbia.edu/~dpwe/pubs/LeeEllis12-SAcC.pdf

Tazi, E. B. (2016). A robust Speaker Identification System based on the combination of GFCC and MFCC methods. *5th International Conference on Multimedia Computing and Systems (ICMCS).* Marrakech, Morocco.

Terhardt, E. (1979). Calculating Virtual Pitch. *Hearing Research*, *1*(2), 155–182.

Vishnubhotla, S., & Epsy-Wilson, C. (2008). An Algorithm for MULTI-PItch Tracking in Co-Channel Speech. *9th Annual Conference of the International Speech Communication Association 2008: (INTERSPEECH 2008).* Brisbane, Australia.

Ward, D. B., Williamson, R. C., & Kennedy, A. (April1998). Broadband Microphone Arrays for Speech Acquistion. *Acoustics Australia, 26*, 17–20.

Weiss, R. J., & Ellis, D. P. (2007). Monaural Speech Separation Using Source-Adapted Models. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* New Paltz, NY.

Wiem, B., Messaoud, M. A., & Aïcha, B. (2016). Single channel speech separation based on sinusoidal modeling. *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*.

Wohlmayr, M., Peharz, R., & Pernkopf, F. (2011). Efficient implementation of probabilistic multi-pitch tracking. *Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic.

Wu, M., Wang, D., & Brown, G. J. (2003). A multipitch tracking algorithm for noisy speech. *IEEE Transactions on Speech and Audio Processing, 11*(3), 229–241.

Yousefian, N., & Loizou, P. C. (2011). A Dual-Microphone Speech Enhancement Algorithm Based on the Coherence Function. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(2), 599–609.

Ziolko, M., Ziolko, B., & Samborski, R. (2009). Dual-Microphone Speech Extraction from Signals with Audio Background. *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Kyoto, Japan.

Zue, V., Seneff, S., & Glass, J. (1990). Speech database development at MIT: TIMIT and beyond. *Speech Communication, 9*, 351–356.

Curriculum Vitae