IDENTIFICATION AND ELIMINATION OF CROSSTALK IN AUDIO RECORDINGS

Ning Ma
Supervised by Dr Guy Brown

Department of Computer Science
University of Sheffield

Submitted in Partial Fulfilment of the Requirements
for the Degree of Master of Science in Advanced Computer Science

Sheffield, UK

August 2003

All sentences or passages quoted in this dissertation from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations which are not the work of the author of this dissertation have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this dissertation and the degree examination as a whole.

Name:

Signature:

Date:

Abstract

Cochannel interference of speech signals is a common practical problem in speech transmission and recording. Ideally, cochannel speaker separation is desirable to recover one or both of the speech signals from the composite signal. Although the human auditory system is adept at resolving the speech of one talker amongst many (the cock tail party effect), this task still appears very difficult. When voices interfere over a monophonic channel (such as the telephone), separation is much more difficult as one voice may mask the other.

There have been many algorithms proposed that eliminate background noise or other interference from cochannel speech signal. The most successful family of cochannel speaker separation approaches operate on the notion that spectral harmonics of each speaker are separated depending on a pitch estimate. In this project, we implement an automatic system of cochannel speaker separation. The system is based on a frame-by-frame speaker separation algorithm that exploits the pitch estimate of the stronger speaker derived from the cochannel speech signal. The idea is to recover the stronger talker's speech by enhancing their harmonic frequencies and formants give a pitch estimate. The weaker talker's speech is then obtained from the residual signal where the harmonics and formants of the stronger talker are suppressed. The performance of our system has been evaluated at target-to-interferer ratios (TIR's) of 15 dB to -15 dB in human listening tests.

Acknowledgments

This dissertation would not have been possible without the support and encouragement of my supervisor, Dr Guy J. Brown.

I also owe a tremendous debt of gratitude to many of my friends, who help me finish my testing and evaluation and gave me lots of valuable opinions.

Thanks to Dr. Stu Wrigley for supplying me the TIMIT database and TIDIGITS database.

Finally, I would like to thank my girlfriend Chunjie Bi, without whom none of this would be possible. She is an unbelievable inspiration to me. Her boundless love and support has truly been a rock to me.

This dissertation is dedicated to my girlfriend and parents.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Cochannel speech is defined as a signal that is a combination of speech from two talkers. This phenomenon occurs in many common situations: if the microphone at the transmitting end is not acoustically isolated, all background noises including close neighbours' voice would be transmitted along with the primary speaker due to the poor placement of the microphone; or if two people are speaking simultaneously (e.g., when talking on the telephone). The goal of cochannel speech separation is to be able to extract the speech of one or both of the talkers from the cochannel speech and minimise artefacts in the processed speech. This is especially important if the recovered speech is passed to an automatic speech recognition system or a speaker recognition system.

Although the human auditory system is adept at separating the speech of one talker from many (e.g. the cock tail party effect), this task appears to be very difficult as masking is an everyday occurrence; quiet sounds are rendered inaudible by louder sounds. There have been many algorithms proposed that separate background noise or other interference (e.g. competing speech signal) from cochannel speech signal. Some methods assume that the interference is stationary (e.g. stationary background noise) or *a priori* information (e.g. the fundamental frequency of the interfering speech) is available, otherwise they fail. These methods are not suitable for the cochannel speech problem because typically the speech interference is not stationary and *a priori* information is unavailable in realistic cochannel situation. In 1970s a promising new group of cochannel speaker separation algorithms emerged, which do not have these restrictions and achieve great success. These methods operate on the notion that spectral harmonics of each speaker are separated exploiting the pitch estimate of the stronger talker derived from the cochannel signal.

In this project, we attempt to design a system that automatically separates both of the speech signals from cochannel speech signal. This system processes the cochannel signal frame-by-frame and most separation procedure is done in frequency domain. First it uses a YIN pitch estimator (Cheveigne and Kawahara, 2002) to get a pitch estimate of the stronger speech. The pitch estimate is used to construct a pair of filters in the frequency domain, which are then applied to the cochannel signal spectrum to separate the stronger and weaker talkers, respectively. The recovered stronger signal is further processed by enhance energy at frequencies corresponding to its harmonics and formants. On the other hand, the weaker talker's speech signal is obtained from the residual signal created when the harmonics and formants of the stronger talker are suppressed. The recovered stronger and weaker signals are then re-synthesised using overlap-add techniques (Oppenheim and Schafer, 1989).

For experimental purposes, cochannel speech signals are generated by linearly adding two digitised speech signals. Cochannel signals created in this manner are sufficient for experimenting with algorithms in an ideal mixing environment. The performance of our technique is evaluated by listening tests. Two widely used speech databases, TIMIT and TIDIGITS, are used in generating cochannel speech. In human listening tests, twelve digits strings were selected from TIDIGITS as target signal and two sentences (one male-spoken and one female-spoken sentence) were selected from TIMIT as interferer. They are mixed with TIR's -12 dB and -18 dB. The system was also tested informally with other TIR's.

This dissertation is organised as follows: Chapter 2 examines the cochannel speech separation problem in more details and reports some previous work in the related research field by other researchers. Chapter 3 gives the requirement of the project and presents a testing scheme. In Chapter 4 we provide detailed descriptions of the voiced/unvoiced detection, the pitch estimation, speaker separation and algorithms comprising our cochannel separation system, step by step. Chapter 5 describes some implementation details and tests conducted to evaluate our system. In Chapter 6 we present the testing results and discuss the results. Chapter 7 summarises our conclusion and suggests future directions for research.

Chapter 2 Literature Survey

2.1 Cochannel Separation Systems Review

The presence of interference causes the quality or intelligibility of speech to degrade. A noisy environment reduces the listener's ability to understand what is said. Many speech enhancement algorithms were proposed to reduce background noise, improve speech quality, or suppress channel or speaker interference in past two decades. Many of these algorithms are quite successful and made the speech enhancement of many applications. Figure 2.1 illustrates some typical applications of speech enhancement.

The study of speaker separation algorithms began with the development of speech enhancement since it is just the case where the interferer is a competing speaker. There are many classes of approaches and each of these classes has its own set of assumptions, advantages, and limitations. One grouping is depending on whether a single-channel or dual-channel (or multi-channel) approach is used. For single-channel applications, only a single microphone is available. Initial work addressing this problem evolved from several techniques based on multi-microphone processing, such as speech enhancement and blind separation. Among the techniques using two-microphone speech acquisition, the classical approach to speech enhancement, adaptive noise cancelling (ANC), was first formulated by Widrow (1975) and has been widely used. This technique was based on a least mean square (LMS) criterion and has the major advantage of requiring no *a priori* knowledge of the noise signal. However, it focuses on restoring only the primary signal, and has difficulties when the primary signal is also picked up by the reference microphone. A more general technique than Widrow's LMS algorithm is proposed by Weinstein *et al.* (1993). This algorithm separates speech signals via the adaptive decorrelation filtering (ADF) between two simultaneously acquired cochannel signals.

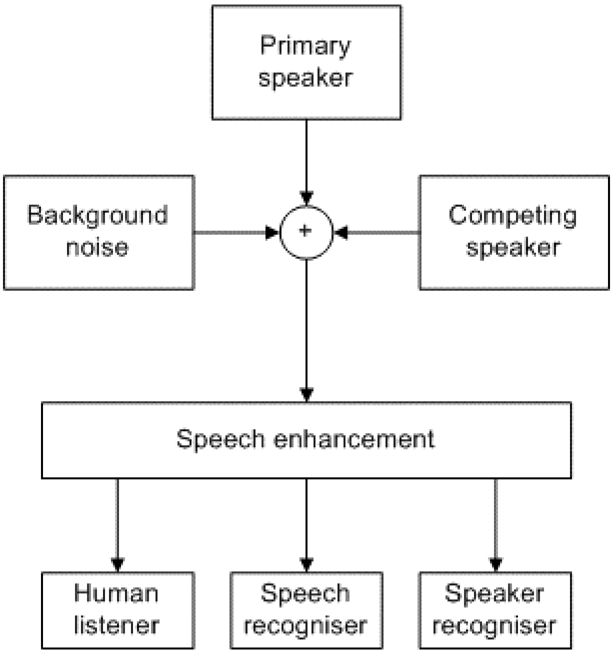


Figure 2.1 Typical applications of speech enhancement

Another enhancement technique is based on adaptive comb filtering (ACF) (Lim, *et al*. 1978). Since voiced speech is quasi-periodic, its magnitude spectrum contains a harmonic structure. The essence of comb filtering is to build a filter that passes the harmonics of speech while rejecting interfering frequency components between the harmonics. It is clear that the magnitude response contains enough energy to allow the accurate estimation of the fundamental frequency component. A filter can be implemented that "passes the fundamental frequency plus harmonics while rejecting frequency components between harmonics" (Deller, *et al.*, 1993). Ideally, spacing between each "tooth" in the comb filter should correspond to the fundamental frequency in Hz and should remain constant throughout the voiced section of speech. Unfortunately, speakers normally vary their pitch and therefore require the comb filter to adapt as data are processed.

If the noise source is a competing speaker, then an enhancement technique similar to comb filtering can be formulated in which spectral harmonics of each speaker are separated based on pitch estimates. Parsons (1976) proposed such a method by means of harmonic selection. The difference between harmonic selection and ACF is that harmonic selection moves the noise under the pitch harmonics, while comb filtering seeks to filter out the noise in the gaps between harmonics.

Hanson and Wong (1984) proposed a harmonic magnitude suppression (HMS) technique to suppress the stronger talker, which is used in Lee and Childers's system (1988). Their system is a two-stage scheme for cochannel speech separation. It uses the HMS as a front-end to make initial spectral estimates of each talker, and then a second stage uses a spectral tailoring technique to obtain better spectral estimation. The system can recover the weaker speech signal with significantly reduced interference. However, the quality of the original speech is not retained.

The important work of McAulay and Quatiery (1986) on sinusoidal modelling of speech provided analysis-synthesis techniques that have been subsequently applied to co-channel separation of speech. Hanson and Wong's method (1984) utilised the envelope of the estimated spectrum and an LPC synthesiser (Gold and Morgan, 2000). In recent work, Morgan *et al* (1997) proposed the use of an overlap-add synthesiser (Oppenheim and Schafer, 1989) to smooth the reconstructed speech. Compared to taking the IFFT of the estimated spectrum directly, both of them recovered the desired speech with less interference and more naturalness. Thus using a synthesiser to reconstruct the separated speech is preferred both for human listeners and further processing.

Beginning with the work of Parsons (1976), Hanson and Wong (1984), Naylor and Boll (1987), cochannel speaker separation algorithms have attempted to exploit pitch information to separate the two talkers. They first try to estimate the pitch of at least one of the talkers, and then to enhance the stronger speech or suppress the interfering talker based on the pitch harmonics. However, these algorithms focus on enhancing voiced speech. Therefore, they generally perform poorly for unvoiced speech. Since vowels (voiced speech) usually possess larger amounts of energy, broadband noise degradation ends to mask unvoiced sections more than voice, thus causing decreased intelligibility. Employing a technique that attempts to improve quality in voiced sections may in fact decrease overall speech quality and intelligibility. For this reason, these methods are not normally used to attenuate broadband additive noise. Instead, their main area of successful application has been in reducing the effects of a competing speaker, where distinct fundamental frequency contours can be identified.

For stationary and well-defined noise sources, effective solutions exist (Widrow, 1975; Lim *et al.*, 1978; Parsons, 1976). However, it is often difficult to formulate a model for speech-like noise sources such as crosstalk. Some researchers proposed solutions addressing this problem that require the use of *a priori* information (Lee and Childers, 1988; Quatieri and Danisewicz, 1990; Arslan and Hansen, 1997). Typically they assumed the position, or amplitude of the true spectral harmonics is either available or not required. In these methods, one successful and widely used technique is Quatieri and Danisewicz (1990). They explored the use of sinusoidal modelling using a least-squares estimation algorithm to determine the sinusoidal components of each of the talkers, based on the Widrow LMS technique (1975). Naylor and Porter (1991) proposed a speech separation algorithm that requires no *a priori* information, based on estimating the pitch of the weaker speech signal and modelling the complex spectrum of the cochannel speech. However, the harmonic location error is generally worse at higher frequencies, thus this approach is sensitive to additive noise.

Some recent investigations conducted on co-channel speaker separation are Benincasa and Savic (1997), Morgan *et al* (1997), Yen and Zhao (1999) and Huang, *et al.*(2000). These methods achieved a very good success. Benincasa and Savic focused on separating overlapping voiced speech signals using constrained nonlinear optimisation. Their work is unique in that it looks to optimise all three parameters, frequency, phase and amplitude for the harmonics of both speakers. Morgan *et al* presented a harmonic enhancement and suppression (HES) system to address the cochannel speech problem. It exploits the pitch estimate of the stronger talker to recover the stronger talker’s speech by enhancing their harmonic frequencies and formants. The weaker talker’s speech is obtained from the residual signal created when the harmonics and formants of the stronger talker are suppressed. A silence/unvoiced detection algorithm was also proposed in their work, which therefore made their system more robust to unvoiced speech and noise. Yen and Zhao’s method is a hybrid of accelerated adaptive decorrelation filtering (ADF) (Naylor and Porter, 1991) and Widrow’s LMS algorithm (1975). A switching between the two algorithms is made depending upon the active/inactive status of the cochannel signal sources. Huang, *et al.* (2000) proposed a sub-band based ADF to address the issues of high computational complexity and slow convergence of ADF.

As the same time, determining the effect of speaker interference on speaker identification is of considerable interest. The development of an effective target speaker extraction technique, which would provide for major improvement of co-channel speech, would also be a very useful tool. They are discussed in Yantorno (1998) and Lewis and Ramachandran (2001).

Humans are very good at distinguishing competing audio streams from each other (the cocktail party problem). This ability has motivated extensive research into the perceptual segregation of sound. The research has resulted in much theoretical and experimental work in so-called Auditory Scene Analysis (ASA) by Bregman (1990) and others, which led to the development of early computational models of the auditory system. The more recent work in this field has been done by Brown and Cooke (1994) and Wang (1996). Brown and Cooke utilise various features derived from grouping and transition cues to separate and organize the individual elements of an auditory map, which is a symbolic CASA architecture. Wang’s model is a neural oscillator architecture, which represents auditory activity within a time-frequency grid. Each point in the grid is associated with a neuron that has an oscillating response pattern. The time dimension is created by a system of delay lines. Both of these two models employed grouping principles. The primary goal of ASA is to model the segregation of sound in the auditory system as accurately as possible.

Another solution to these kinds of problems is called Blind Source Separation (BSS). Blind source separation attempts, as the name states, to separate a mixture of signals into their different sources. The word "blind" is used because we have no prior knowledge about the statistics of the source or the mixing process (e.g. assume mixed linearly) in general. One popular information theoretical approach for BBS is the independent component analysis (ICA) (Hyvarinen and Oja, 1999), which is used to separate independent sources from unknown linear mixtures of the statistically independent source signals. To make sure that ICA works properly, some constraints have to be made to the cochannel speech signals. First the number of sources should not be greater than the number of sensors, in this case, microphones. Second only very small sensor noise is allowed. Thus the ICA method is not suitable for the cochannel speaker separation problem as usually cochannel speech is recorded using only one microphone.

Although separation of the target speaker from cochannel speech was richly researched in past years, it has been still very difficult. Therefore, to make the problem more manageable, it is worthwhile to understand what the final use of the target speech is. If the final goal were that a human listener would use the speech, then intelligibility and quality would be important characteristics of the extracted speech. However, if the extracted speech is to be used for speaker identification, then one would be concerned with how much and what type of target speech is need to perform “good” speaker identification, i.e., voiced and unvoiced speech or just voiced speech. It is generally known that humans and speech recognizers process information in different ways; a perceptual improvement does not necessarily translate into an improvement in recognition accuracy. Seltzer (2000) pointed out that after performing the separation algorithm proposed by Morgan *et al* (1997), the reduction of word error rate for speech recognition is quite small.

2.2 Pitch Estimation

As some pitch-based systems are not very robust with respect to pitch estimation errors, a good pitch estimate algorithm for cochannel speech is required. The research about single-talker pitch estimate is quite rich and this is well documented in the literatures (Wise, *et al.*, 1976; Droppo and Acero, 1998; Rosier and Frenier, 2002; Cheveigne and Kawahara, 2002). Many authors tried to estimate fundamental frequencies in time domain, using extensions of techniques based on the autocorrelation method or the Maximum Likelihood principle. In a real cochannel situation, one would be faced with the need to obtain a pitch estimate of one or both of the speakers as a starting point for separating the two speech signals. Recent experiments (Morgan, *et al.*, 1997) have showed that the use of a single-talker pitch detector proved satisfactory to determine the pitch of the stronger talker in cochannel speech.

Previous work has demonstrated that the ML pitch detector (Wise, *et al.*, 1976) outperforms cepstral, harmonic matching and auditory synchrony based pitch detectors (Naylor and Boll, 1987). The ML technique performs well in the presence of additive noise, which is important in noisy conditions or when the interfering speech is unvoiced. One of the drawbacks of the ML pitch detector is that it outputs an integer pitch estimate rather than a fractional one. This is sometimes not adequate in accuracy for some tasks, e.g. for the cochannel speech separation. A possible solution to this problem is to employ a multi-resolution search to determine a fractional pitch period, proposed by Morgan *et* al (1997). However, this method will also increase the amount of computation, which is a concern in real-time system. Recently a new pitch determination algorithm called YIN has been described in a journal publication by Cheveigne and Kawahara (2002). The YIN algorithm is an approach to pitch determination that is based on autocorrelation, a well-known time-domain approach to the problem. YIN improves upon a simple autocorrelation scheme in a number of ways and gives a fractional pitch estimate. The results have showed that its error rates are about three times lower than the best competing methods. The YIN algorithm outperforms the ML approach in the way that it outputs a fractional pitch period estimate directly, and it is also relatively a simple and effective algorithm to implement. Some other familiar schemes such as median smoothing (Rabiner and Schafer, 1978) can be included in the post-processing procedure. These techniques can further improve the robustness of the pitch estimate methods.

In the case of the cochannel speech separation, some algorithms were proposed to estimate two-talker pitch simultaneously. Naylor and Porter’s method is based on the modified covariance (MC) spectrum estimator, which tries to detect the pitch of a speech signal that is being masked by a much louder speech signal. Other multi-talker pitch estimators are based on extensions of single-talker maximum likelihood (Chazan, *et al.*, 1993). However, previous experiments showed that two-talker pitch estimator seems computationally intensive and no better performance (Morgan, *et al.*, 1997).

2.3 Testing and Evaluation

When we consider speech enhancement, we normally think of improving a signal-to-noise ratio (SNR). However, this may not be the most appropriate performance criterion for speech enhancement. All listeners have an intuitive understanding of speech quality, intelligibility, and listener fatigue. These aspects are not easy to quantify in most speech enhancement applications since they are based on subjective evaluation of the processed speech signal. However, many tests have been developed that assess the speech intelligibility by measuring the speech reception threshold (SRT) (Plomp and Mimpen, 1979; Nilsson *et al.*, 1994; Versfeld *et al.*, 1999). A SRT is the lowest intensity and equally weighted two syllable word is understood approximately fifty percent of the time. The pure tone average and speech reception threshold should be within 7 dB of each other. Comparison of the speech reception threshold and the pure tone average serves as a check on the validity of the pure tone thresholds. Discrepancies between these measures may suggest a functional or non-organic hearing loss.

2.4 Related Software

In this project, most of the implementation and experiments are done in MATLAB. MATLAB is a high-performance language for technical computing. It integrates computation, visualisation, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. It is also an interactive system that allows us to solve many technical computing problems, especially those with matrix and vector formulations. Thus MATLAB is ideally suitable for the speech processing work as every discrete-time signal can be represented by a vector or matrix. MATLAB has many build-in mathematics functions, such as Fourier transform functions, which are frequently used in signal processing. These functions have been optimised for years, thus have very high performance. Another benefit of MATLAB is that it is very convenient to visualise the result. MATLAB supplies a big family of plotting and data visualisation functions, which operate with other matrix manipulating functions naturally and easily.

However, MATLAB is not designed to develop a separate speech processing system, as it cannot be compiled to an independent program and is difficult to build a function library to be used by other programs. Java is first developed by Sun in early 1990s and designed to be an object-oriented, robust, portable, and high-performance language. All Java source codes are compiled into byte-codes, which can then be executed by Java virtual machine (JVM) on every platform. Many speech research works were done with Java and several Java speech libraries were developed for speech processing, known as Java Speech API (JSAPI).

Chapter 3 Requirements and Analysis

3.1 Introduction

As in any engineering problem, it is useful to have a clear understanding of the objectives and the ability to measure system performance in achieving those objectives. The goal of our work is to develop an automatic cochannel speaker separation system that would be capable of separating the cochannel signal without requiring *a priori* information that is unavailable in realistic cochannel situation, would minimise artefacts in the processed speech, and would emphasise software engineering approaches. The project is based around some audio recordings in which two speakers are conversing, and separate recordings of the two conversing speakers are required. Thus continuous real-time throughput is not necessary. To make the problem more manageable, the final use of the recovered recordings is targeted for human transcribing. So it is desirable to reconstruct the speech with less interference and more naturalness. Final performance of the approach will be evaluated by listening tests.

3.2 System Overview

From previously research work, we were able to make use of proven techniques and avoid known pitfalls. There are many classes of solutions addressing the cochannel speaker separation problems. In Chapter 2, we reviewed several approaches including Blind Source Separation (BSS) and some pitch-based methods. The BSS approach requires the number of sources should not be greater than the number of microphones, and this, in the case of cochannel speaker separation, means at least two microphones are needed. This is sometimes impractical because audio recordings might be recorded using only one microphone (or, e.g., the case of telephone recordings). Furthermore, BSS will fail if the variance between the two microphones varies according to time, but in realistic world, people always move their heads when speaking, causing non-stationary variance. These restrictions make BSS inappropriate for our system.

The methods based on fundamental frequency (F0, or pitch) tracking do not have these restrictions. These techniques capitalise on the property that waveforms during voiced passages are periodic. The separation takes advantage of differences in fundamental frequency contours. We found the work of Morgan *et al* (1997), who proposed a harmonic enhancement and suppression (HES) technique to address the cochannel separation problem, is quite promising. The HES approach is based the pitch estimate of the cochannel speech and considers it as that of the stronger talker. This assumption can be satisfied in a realistic situation as normally the cochannel speech contains enough energy of the stronger talker for a pitch estimator to get an accurate estimate. Our system is based on the principle informed by the approach described in Morgan *et al* (1997). It avoids the need to jointly estimate the pitch of both talkers, which is a main problem in previous methods as reliably estimating pitch of one talker in the presence of another is a very difficult task. Instead estimating the pitch of the stronger talker is sufficient to achieve separation and make an estimate of the pitch of the weaker talker during subsequent processing.

The basic strategy of our system is to process the signals frame-by-frame and the stronger talker’s speech is recovered by discarding the energy not associated with the harmonic frequencies of the stronger talker in frequency domain, given a pitch estimate of the stronger talker. The weaker talker's speech is obtained from the residual signal where the energy at the harmonics is suppressed. The recovered stronger and weaker signals are then assigned to the target or interfering talker and re-synthesised using overlap-add techniques. A voiced/unvoiced detector is also implemented to assist in smoothing the pitch track. The following sections contain more detailed analysis of each step.

3.3 Modules Analysis

A. Voiced/Unvoiced Detection

The need for deciding whether a given segment of a speech waveform should be classified as voiced speech, unvoiced speech, or silence arises in many speech analysis systems, for example, fundamental frequency estimation, formant extraction or syllable marking. In our system, the voiced/unvoiced decision is used in smoothing the pitch estimate. The S/UV detection algorithm classifies each analysis frame of the cochannel signal as either voiced or S/UV.

A variety of approaches have been described in the literature for making this decision. The most classic approach is short-time average zero-crossing rate. In the context of discrete-time signals, a zero-crossing is said to occur if successive samples have different algebraic signs. A reasonable generalisation is that if the zero-crossing rate is high, the speech signal is unvoiced, while if the zero-crossing rate is low, the speech signal is voiced. However, an accurate decision is not possible based on short-time average zero-crossing rate alone because we have not said what is high and what is low. For this reason, nowadays a combination of several features was used to classify voiced/unvoiced signal and achieved very high accuracy.

B. Pitch Estimate

As our system is based on tracking the fundamental frequency contour (pitch contour), the pitch estimate step is the most fundamental and important step. In the case of cochannel speaker separation problem, one speaker's speech is often contaminated by the interfering speech. Thus a robust and accurate pitch estimator is required.

Morgan *et al* (1997) pointed out that previous work (Naylor and Boll, 1987) has demonstrated that ML pitch detector (Wise *et al*, 1976) outperforms cepstral, harmonic matching and auditory synchrony based pitch detectors. However, one of the drawbacks of the ML pitch detector is that it provides an integer estimate of pitch period, which is found inadequate in further processing. Thus a more accurate pitch period estimate is needed. Morgan *et al* proposed a technique that a multi-resolution search is conducted to determine a fractional pitch period. This technique can give some helps addressing the problem to some extent, but at the same time it also introduces new computational burden, which is key concern in a real-time system.

Recently Cheveigne and Kawahara (2002) developed a pitch determination algorithm, YIN, which combines the well-known autocorrelation method (Licklider, 1951) and Average Magnitude Difference Function (AMDF) methods (Ross, 1974) with a set of incremental modifications that combine to improve the overall pitch estimation. Cheveigne and Kawahara (2001) described a methodology for evaluation of pitch estimation algorithms and provided results for a set of methods. The evaluation was performed over an extensive laryngographlabeled database aggregated from several sources comprising speech from a total of 38 speakers. The results showed the YIN method was uniformly more effective than others and the error rates are about three times lower than the best competing methods. In the case of cochannel speaker separation we found the performance of YIN method is very good and steady.

C. Speaker Recovery

The speaker recovery is the main step of our whole system. It is in this step where the cochannel speech is separated into a stronger signal and a weaker signal. If we consider the weaker speech as the interfering signal such as noise, there are several methods available addressing the problem. Adaptive comb filtering (ACF) (Lim, 1979) is the most famous method. Since voiced speech is quasi-periodic, its magnitude spectrum contains a harmonic structure. If the noise is non-periodic, its energy will be distributed throughout the spectrum. The essence of comb filtering is to build a filter that passes the harmonics of speech while rejecting noise frequency components between the harmonics.

If the degrading noise source is a competing talker, then an enhancement technique similar to comb filtering can be formulated in which spectral harmonics of each talker are separated based on external pitch estimates. Parsons (1976) proposed such a method in which a short-term spectrum is used to separate competing speakers. All processing is performed in the frequency domain. An alternative to frequency domain harmonic selection is time domain harmonic scaling (TDHS). This is a time domain technique that requires pitch-synchronous block decimation and interpolation. Our system is based on the Parsons method.

After modifying the spectrum in frequency domain, we must transform it back to time domain. There are two distinctly different methods for reconstructing a signal from its short-time spectrum. One is filter bank summation (FBS) method and the other is overlap-add method. Both methods have been shown capable of reconstructing the original signal exactly.

D. Speaker Assignment

Since in independently conversational speech the talkers can randomly appear as either the stronger or weaker talker as their speech signals evolve over time, a speaker assignment algorithm is required. Speaker assignment is needed to assign the recovered stronger speech and weaker speech to correct output signals.

Several strategies have been investigated for assigning the recovered signals to the correct talkers. One of the most popular is two-talker pitch tracking (Zissman, and Seward, 1992; Wu *et al*, 2003), which is used when both talkers are voiced. Other techniques use distortion metrics to compare the spectra of the recovered signals.

However, our cochannel speaker separation system does not cover the speaker assignment stage. This is because:

1. The assignment error rate is too high, causing the output worse than input.
2. For many cochannel speeches, the need for speaker assignment is not quite often.

3.4 Software Engineering Methodology

Our system is designed according to the principle of software engineering methodology. The system is divided into several manageable modules and each functional module, or logical module, is independent of each other.

One advantage of this approach is that as we break the problem down into manageable steps, we are able to work on each separate module to enhance its performance. Another advantage is we can perform testing step by step. Furthermore, we can also compare our system with other techniques at a module level. Thus we can replace some module by a better one without influencing others as well as the whole system. However, the relationship between two adjacent modules is still very important to make sure the whole system works better and is also a key concern to decide how each module should work together.

3.5 Testing and Evaluation Scheme

When we consider speech enhancement, we normally think of improving a signal-to-noise ratio (SNR). However, this may not be the most appropriate performance criterion for speech enhancement. All listeners have an intuitive understanding of speech quality, intelligibility, and listener fatigue. These aspects are not easy to quantify in most speech enhancement applications since they are based on subjective evaluation of the processed speech signal. However, many tests have been developed that assess the speech intelligibility by measuring the speech reception threshold (SRT) (Plomp and Mimpen, 1979; Nilsson *et al.*, 1994; Versfeld *et al.*, 1999). SRT is a very reproducible test to determine the lowest sound intensity level at which fifty percent or more of the test words are repeated correctly. For the SRT a sentence that is masked by noise is presented to a listener. The listener has to recall the sentence precisely. If the listener produces a correct answer, the next sentence is presented with an increased noise level of 2 dB. This continues till the response of the subject is incorrect, than the noise level will be decreased by 2 dB. After a number of presentations, a noise level is obtained for which 50 % of the sentences are responded correctly.

Running the SRT is complex because all the utterances have to be balanced for difficulty. Normally in a speech database some utterances (e.g., very nasalised voices) are easier to recognise than others. Hence, an SRT measured using this database would be invalid. To use this database, one needs to scale their level in a pre-test to make each interval of the same difficulty. However, we can extract the concept of the SRT to form an evaluation scheme for our system. Our system was subjected to human listening tests on linearly added speech signals. The goal of the human listening tests is to transcribe a target speech signal when contaminated by a stronger interfering speech signal, and use these transcriptions to determine the performance variance between processed and unprocessed cochannel speech signals.

Two standard speech databases were used for testing our speaker separation algorithm: TIMIT and TIDIGITS. The TIMIT corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. It contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. TIDIGITS is a speaker-independent connected-digit database. This dialectically balanced database consists of more than 25 thousand digit sequences spoken by over 300 men, women, and children. The data were collected in a quiet environment and digitized at 20 kHz.

Two sentences were selected from the TIMIT database as interfering speech. One is uttered by a male speaker and the other is uttered by a female speaker. Twelve digit strings were selected from the TIDIGITS database, which are uttered by different male speakers. Each TIDIGITS string was considered as a target signal and the TIMIT sentences were considered as interferers. They were linearly added at -12 dB, -18 dB TIR's, forming 48 different cochannel speech signals. Eight untrained listeners were asked to transcribe the digit strings they heard in four types of speech signals: unprocessed cochannel signal at -12 dB, processed weaker signal (digit string signal) at -12 dB, unprocessed cochannel signal at -18 dB, and processed weaker signal at -18 dB. The test case of 0 dB TIR was not evaluated because it is a trivial separation task for humans given the configuration of our experiment. We found human listeners were so good at transcribing digits that even at -10 TIR the accuracy is still very high.

Speech was presented to the listener in the following order: two repetitions of the interferer's clean speech followed by unprocessed cochannel signals and processed signals. Without first hearing the interferer's clean speech, our listener found it too difficult to block out the stronger interferer and focus on the weaker target talker. Table 3-1 is a detailed speech presentation sequence.

Table 3-1 Speech Presentation Sequence

Step 1	Step 2	Step 3	Step 4	Step 5
Clean interferer speech	three unprocessed cochannel signals at -12 dB	three processed signals at -12 dB	three unprocessed cochannel at -18 dB	three processed signals at -18 dB

Chapter 4 System Design

4.1 Framework Design

Briefly, our strategy was to process the cochannel speech frame by frame and the separation process is done in frequency domain. A frame of speech, formally, is defined to be the product of a shifted window with the speech sequence:

$$f_s \overset{def}{=} s(n)w(m-n) \tag{4.1}$$

Practically, a frame is just a “chunk” of speech which perhaps has been tapered by the window. In this project, the system uses an analysis frame length of N = 400 samples, with a Hamming analysis window, and assumes a frame increment of R = 100 samples. A sample rate of F_s = 20 kHz is used, thus the frame length of N (400 samples) corresponds 20 ms.

Figure 4.1 is a block diagram of our cochannel speaker separation system. Referring to Figure 4.1, the system consists four main function parts: a voiced/unvoiced detector, a YIN pitch detector, speaker recovery stage, and speaker assignment stage. For each N-sample analysis frame, the cochannel signal $s_+[n]$ is analysed using a *voiced/unvoiced detector* to get a V/UV decision. A *YIN pitch estimator* is then performed on the current analysis frame, combined with its V/UV decision, to determine the pitch period of the stronger talker, denoted as $\tilde{p}_s$, which corresponds to a radian pitch frequency $\tilde{w}_s = 2\pi / \tilde{p}_s$. The V/UV decision helps smooth the pitch estimate. To make the pitch estimate more accurate, and to avoid the interfering effect of the weaker talker when calculating pitch period, we use a two-pass speaker recovery method. At the first pass, the pitch period $\tilde{p}_s$ is used to drive the *speaker recovery* algorithm, which produces estimates of the stronger speech signal. Then the recovered signal is used to recalculate the pitch period, denoted as $\hat{p}_s$. The recalculated pitch is believed more accurate to the stronger talker’s real pitch than the former one because it is calculated when the interfering weaker speech is depressed. An informal test shows this method can help get the stronger talker’s pitch at the frames where it is strongly interfered by the weak speech.

As seen in Figure 4.1, *speaker recovery* is a two-stage algorithm that exploits the pitch estimate of the stronger talker $\hat{p}_s$ and the cochannel signal $s_+[n]$. The first stage is *spectral recovery*, in which a filter pairs are applied in the frequency domain to separate cochannel speech into recovered stronger signal $\tilde{s}_s[n]$ and weaker signal $\tilde{s}_w[n]$, respectively. The second stage of speaker recovery is *spectral enhancement*, where the recovered cochannel speech is further enhanced by nonlinear post-processing steps.

All these inputs are then passed to the final stage: the *speaker assignment*. The speaker assignment algorithm uses a pitch-based algorithm to produce two output recovered signals, $\hat{s}_1[n]$ and $\hat{s}_2[n]$, one for the target and the other for the interferer.

This approach avoids the need to jointly estimate the pitch of both talkers, which is a main problem in previous methods as reliably estimating pitch of one talker in the presence of another is a very difficult task. Instead estimating the pitch of the stronger talker is sufficient to achieve separation and make an estimate of the pitch of the weaker talker during subsequent processing. The following sections contain more complete descriptions of each stage shown in Figure 4.1.

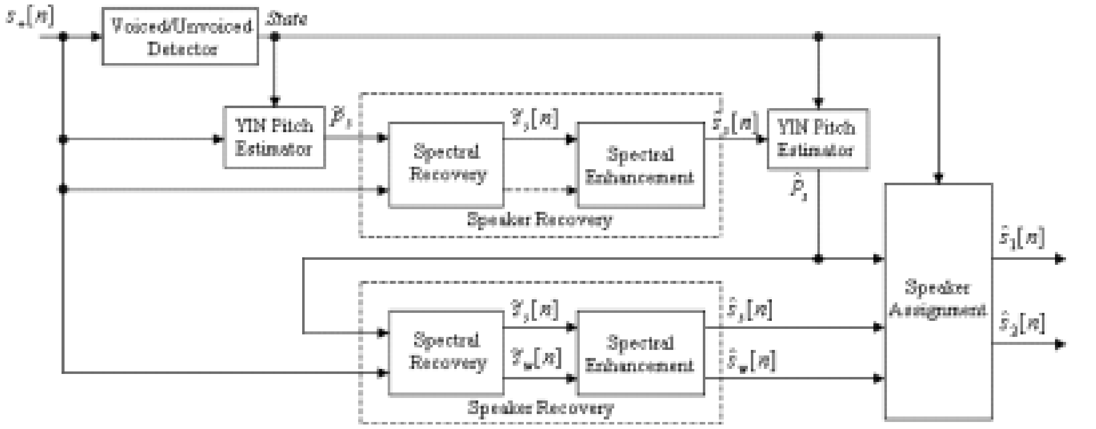


Figure 4.1 Block diagram of the cochannel speaker separation system.

4.2 Voiced/Unvoiced Detector

A. Introduction

The need for deciding whether a given frame of a speech waveform should be classified as voiced speech, unvoiced speech, or silence arises in our system. The decision is used to smooth the pitch estimate according to several rules described in section 4.3C and is stored in the pitch estimate by setting pitch estimates of unvoiced/silence frames to a specific (invalid) value.

A variety of approaches have been described in the literature for making this decision. In our system, we use a pattern recognition approach for classifying a given speech segment. The pattern recognition approach provides an effective method that combines the contributions of five features, which individually may not be sufficient to discriminate between classes. The method is essentially a classical hypothesis testing procedure based on the statistical decision theory. In this method, for each of the two classes, a non-Euclidean distance measure is computed from a set of measurements made on the speech frame and the segment is assigned to the class with minimum distance. The detector generates a binary voicing decision, which we label V[n], where V[n]=1 corresponds to a voiced classification.

The success of a hypothesis-testing depends upon the measurements or features which are used in the decision criterion. Five features were selected in our system based on the experimental evidence:

- Energy of the signal
- Zero–crossing rate of the signal
- First predictor coefficient
- Energy of the prediction error
- Autocorrelation coefficient at unit sample delay

A block diagram of the analysis and decision algorithm is shown in the following figure.

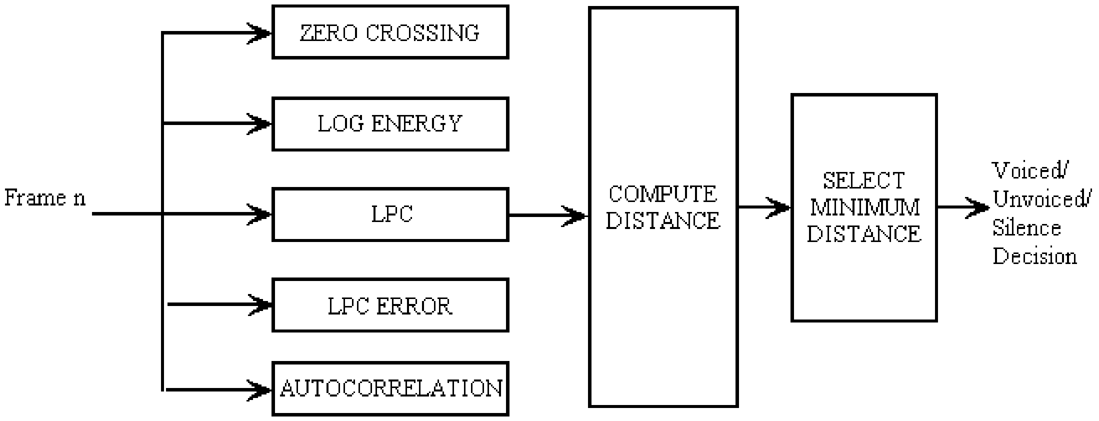


Figure 4.2 Block diagram of the Voiced/Unvoiced decision algorithm

B. Feature Measurements

1. Zero-crossing count N_z, the number of zero-crossing in the frame

   The zero crossing count is an indicator of the frequency at which the energy is concentrated in the signal spectrum. Voiced speech is produced as a result of excitation of the vocal tract by the periodic flow of air at the glottis and usually shows a low zero crossing count (usually 14 per 10 msec).. Unvoiced speech is produced due to excitation of the vocal tract by the noise-like source at a point of constriction in the interior of the vocal tract and shows a high zero crossing count (usually 49 per 10 msec). The zero-crossing count of silence is expected to be lower than for unvoiced speech, but quite comparable to that for voiced speech.

2. Log energy E_s is defined as

$$E_s = 10\log(\varepsilon + \frac{1}{N}\sum_{n=1}^{N} S^2(n))$$

   where ε is a small positive constant added to prevent the computing of log of zero. Generally speaking, E_s for voiced data is much higher than the energy of silence. The energy of unvoiced data is usually lower than that for voiced sounds but higher than that for silence.

3. The first predictor coefficient $\alpha[1]$ computed from a 12-order linear prediction analysis.

4. Log of the prediction error normalised by the first autocorrelation coefficient, $\log(E^{(p)} / R[1]+1)$.

5. Normalized autocorrelation coefficient at unit sample delay, C_1 which is defined as

$$C_1 = \frac{\sum_{n=1}^{N} s(n)s(n-1)}{\sqrt{(\sum_{n=1}^{N} s^2(n))(\sum_{n=0}^{N-1} s^2(n))}}$$

   This parameter is the autocorrelation between adjacent speech samples. Due to the concentration of low frequency energy of voiced sounds, adjacent samples of voiced speech waveform are highly correlated and thus this parameter is close to 1. On the other hand, the correlation is close to zero for unvoiced speech.

4.3 YIN Pitch Estimator

A. Introduction

The YIN pitch estimator is an implementation of a method developed by Cheveigne and Kawahara (2002). The method combines the well-known autocorrelation method (Licklider, 1951) and Average Magnitude Difference Function (AMDF) methods (Ross, 1974) with a set of incremental modifications that combine to improve the overall pitch estimation. These modifications focus on two different issues: one is how to get a periodic lag function less sensitive to amplitude change and imperfect periodicity; the other is how to extract the correct period from it.

The approach is very much an engineering solution to the problem of imperfectly period signals rather than a purely theoretic one. Cheveigne and Kawahara (2001) described a methodology for evaluation of pitch estimation algorithms and provided results for a set of methods. The results showed the YIN method was nevertheless uniformly more effective than others. YIN performs very well especially in noisy conditions and the error rates are about three times lower than the best competing methods. This feature is very important in the case of cochannel speech as the unvoiced part in interfering speech may act like noise interferer. YIN is robust to get the periodic information in either target speech or interfering speech, whichever is stronger. Furthermore, the algorithm is relatively simple and may be implemented efficiently and with low latency. Thus YIN is ideal for our cochannel speech separation system.

Figure 4.3 is the block diagram of our pitch estimator.

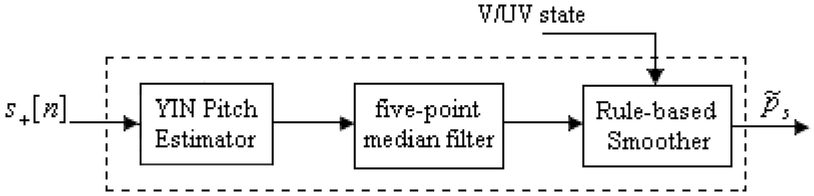


Figure 4.3 Block diagram of pitch estimator

B. Pitch Estimation

YIN divides an audio signal into a series of overlapping windows, on each of which functions are calculated to get the pitch. The fundamental function of the YIN algorithm is the autocorrelation function (ACF), which is given by Equation 4.2:

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau} \tag{4.2}$$

where $r_t(\tau)$ or $r_t'(\tau)$ is the autocorrelation function of lag τ calculated at time index t, and$W$ is the window size. The autocorrelation function chooses the highest non-zero-lag peak within a window, but it is quite sensitive to amplitude changes (Hess, 1983), which may cause the algorithm to choose a higher-order peak. A difference function (Equation 4.3) is then introduced to make the system less susceptible to the problem.

$$d_t(\tau) = \sum_{j=t+1}^{t+W} (x_j - x_{j+\tau})^2 \qquad (4.3)$$

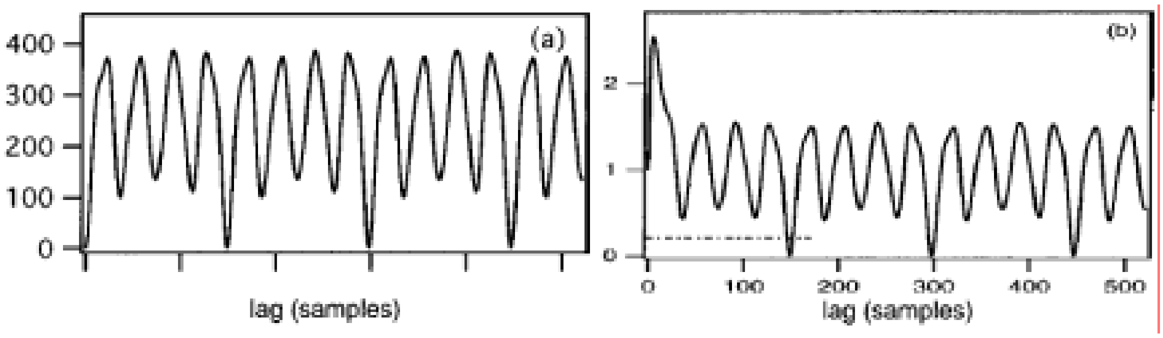


Figure 4.4 (a) Difference function . (b) Cumulative mean normalized difference function. Note that the function starts at 1 rather than 0 and remains high until the dip at the period.

The difference function of Figure 4.4 (a) is zero at zero lag and often nonzero at the period because of imperfect periodicity. Unless a lower limit is set on the search range, the algorithm must choose the zero-lag dip instead of the period dip and the method must fail. Even if a limit is set, a strong resonance at the first formant might produce a series of secondary dips, one of which might be deeper than the period dip. The solution is to replace the difference function by the “cumulative mean normalized difference function” (CMND), which is defined in Equation 4.4. Figure 4.4 (a) and (b) shows the difference between the two functions.

$$d'_t(\tau) = \begin{cases} 1, & if\, \tau = 0, \\ d_t(\tau) \Big/ \left[(1/\tau) \sum_{j}^{\tau} d_t(j) \right] & otherwise \end{cases} . \qquad (4.4)$$

Based on the Equation 4.4, some other steps are employed to make the algorithm further robust to noise. Rather than rely on the minimum value over a window, YIN recognises that even the CMND function is liable to produce multiples of the period with lower values. Instead an absolute threshold is set. The threshold scheme marks the smallest lag in the function that falls below the threshold as the pitch period.

Finally, a best local estimate is chosen instead of original value. This step is effectively a smoothing of the final pitch-time function. After calculating the estimated pitch for all time frame, YIN then looks again at each point chosen to see if any local point (in a window we define to be the original window size) were chosen with “better” (i.e. lower) values of the CMND function. These pitches are then chosen instead. The “best local estimate” method is reminiscent of median smoothing, but differs in that it takes into account a relatively short interval and bases its choice on quality rather than mere continuity.

Following all these steps, YIN appears to be quite a robust estimator of pitch and performs very well in either a noisy condition or a cochannel speech case.

In our system, a window size of W = 25 ms is used when calculating pitch period. With a sample rate of F_s = 20 kHz, this window size corresponds 500 samples. This window size was selected so that fundamental frequencies as low as 40 Hz could be identified.

Figure 4.5 illustrates how the YIN pitch estimator tracks the pitch of the stronger talker for two linearly added sentences “*Don't ask me to carry an oily rag like that*” and “*She had your dark suit in greasy wash…*”. The target-interferer-ratio (TIR) for these complete sentences is 6 dB, although the TIR's between different analysis windows can vary significantly. The target (stronger) speech is uttered by a female speaker and the interfering (weaker) speech is uttered by a male speaker. Both sentences are selected from TIMIT database and contain unvoiced speech, some silence, vowels, and low-energy voiced consonants. In Figure 4.5, the upper part is the waveform of the mixed speech, and the middle part is its corresponding pitch track. In the lower part the darker curve is pitch track for target (female) speech while the dotted curve is that for interfering (male) speech. The result is almost perfect. YIN tracks the pitch for all the voiced part of the target speech as well as the pitch for some voiced part of the interfering speech when the target is unvoiced/silence at those frame. A formal evaluation about pitch tracking performance is included in chapter 6.

Figure 4.6 shows the pitch track for the mixed speech (the solid line) superimposed on the *a priori* pitch tracks for both talkers (dotted line and dashed line) at 6 dB TIR. In this example both sentences are uttered by female (different) speakers; so their fundamental frequencies are not so different as in Figure 4.5. YIN performs still very well in this case, although at some voiced part of the target speech it tracks the pitch of the interfere speech. This is because the cochannel pitch estimate is not always the pitch of one talker; it is the pitch of the stronger talker. It tends to swap from one talker's pitch track to the other over time, and at all times the cochannel pitch corresponds closely to the pitch of one talker or the other.

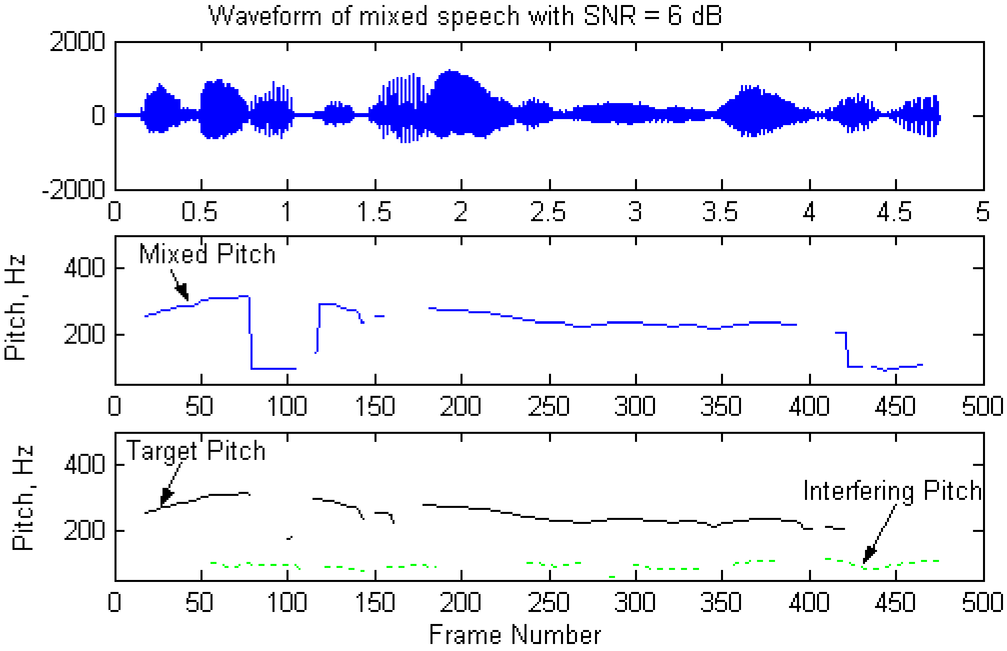


Figure 4.5 Pitch tracking for the mixed speech, target speech, and interfering speech.

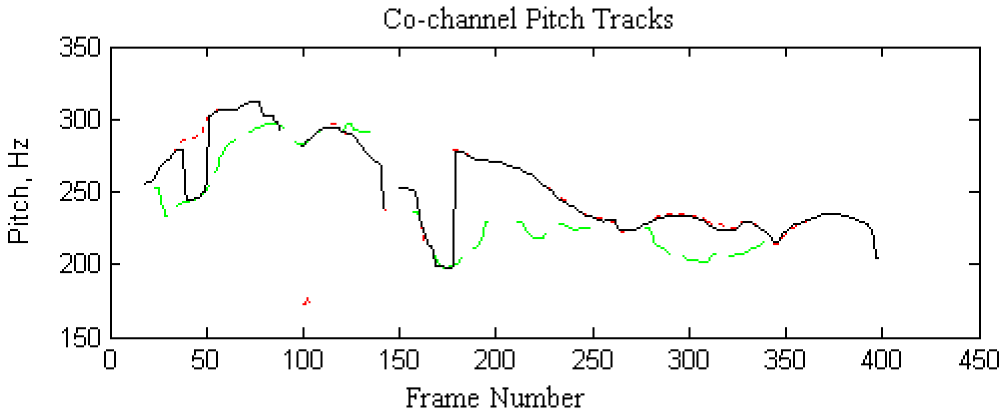


Figure 4.6 Pitch track of the stronger talker superimposed on the *a priori* pitch tracks of both talkers at 6 dB SNR.

C. Post-Processing Methods

Although the YIN performs very well in pitch estimation, we still need some post-processing methods to correct errors in pitch estimation. A popular technique is median-smoothing (Tukey, 1974). The concept of median-smoothing is very simple but it is quite useful. This method looks at a sequence of final pitch estimates and tries to find the median of this sequence. Then the centre of the sequence is replaced by the median. For example, in the sequence "*5, 6, 12, 7, 8*", the median is 7; thus the centre of the sequence is replaced by 7, so the new sequence becomes "*5, 6, 7, 7, 8*". In this example, the outlier 12 was replaced.

In many cases, median-smoothing is preferable to a linear filter, for which the effect of an outlier would spread to other samples. In the case of pitch estimation, for each window of N points, where N is an odd integer, the value of point (N+1)/2 in the window is set equal to the median of the points in the window. The window then is stepped along by one sample point, and the same operation is repeated.

In our system, two median-smoothing filters are applied to the output of the YIN estimator to compensate for gross errors and to smooth the estimate when the pitch may not be stationary. We cascade a three-point median filter with a five-point median filter. Furthermore, before performing median-smoothing we set all the inappropriate pitch estimates (e.g. too high or too low) to zero, which represents unvoiced. For example, in a sequence of "*200, 201, 0, 210,* 205", the zero gets changed to a "*201*" and the modified sequence is "*200, 201, 201, 210,* 205". By this means, apparently pitch errors and unvoiced gap errors can be fixed by median-smoothing.

After being smoothed by two median filters, the output pitch estimate is then processed by a rule-based smoothing algorithm (Seltzer, 2000) for each analysis frame. The rule-based smoother combines the voiced/unvoiced state from a Voiced/Unvoiced detector and the pitch estimate value to smooth the pitch track. Following rules are used to smooth the estimated pitch contour:

- A voiced segment of speech must consist of at least three successive frames. Any voiced segments less than three frames in length are relabelled as unvoiced and corresponding pitch estimates are set to a specific invalid value, e.g. "NaN" in Matlab.
- An unvoiced segment must also last at least three frames. Any unvoiced segment that is less than three frames is considered incorrectly labelled and changed to voiced

speech. The pitch estimates for these frames (formerly set to an invalid value) are determined by linearly interpolating between the pitch estimates of the bounding adjacent voiced frames.

- At voiced/unvoiced or unvoiced/voiced boundaries, the candidate pitch estimates of the unvoiced regions are re-evaluated. If the unvoiced frames at the boundary have a pitch estimate that is within a fixed threshold of the neighbouring voiced frames, those frames are relabelled as voiced. This threshold was empirically set at 8Hz.

The smoother enables the pitch estimates to contain the voiced/unvoiced information. After these rules are applied, a signal frame can be considered as a voiced frame if the frame has a corresponding valid pitch estimate value. Unvoiced frame will hold an invalid pitch value. Combining all these post-processing methods, the YIN pitch estimator is believed to work very well in the cochannel speech separation system.

4.4 Speaker Recovery

A. Introduction

The speaker recovery step is a main part of our system as most separation work is done in this stage. The speaker recovery algorithm operates on each analysis frame and attempts to recover the speech of both the stronger and weaker talkers. For voiced sounds, the basic strategy behind this algorithm is to recover the stronger talker by enhancing their formants and pitch harmonics. The recovered weaker talker, at the same time, is the residual signal obtained by suppressing the stronger talker’s formants and pitch harmonics. For the unvoiced or silence frames, we simply pass them through our speaker recovery algorithm unprocessed. However, a scaling term (which is typically in the range of 0.3-0.6) is applied because applying the algorithm to voiced sounds reduces the energy present.

The algorithm involves two stages: the first one is spectral recovery stage and the other is spectral enhancement stage. Figure 4.7 is a block diagram that shows the steps in the speaker recovery algorithm. Ideally, the outputs of this system are the recovered speech signals of the stronger talker and the weaker talker, which are then passed to the speaker assignment stage.

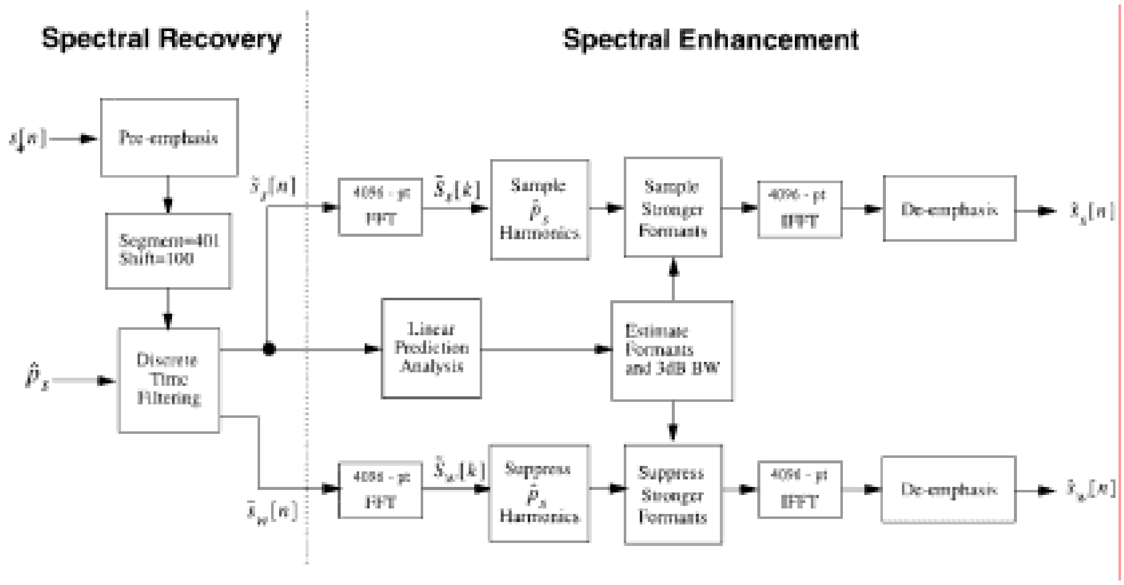


Figure 4.7 Block diagram of the speaker recovery system.

B. *Spectral Recovery*

Speech signals have a spectrum that falls off at high frequencies. In our cochannel speaker separation system, it is desirable that this high-frequency falloff be compensated by "pre-emphasis". A simple and widely used method of pre-emphasis is linear filtering by a "first difference" filter of the form:

$$y[n] = x[n] - \alpha \cdot x[n-1]$$

where $x[n]$ is the input speech signal and $y[n]$ is the output "pre-emphasised speech" and α is an adjustable parameter.

In the first stage of speaker recovery, the cochannel signal is first pre-emphasised with a factor of .95 to flatten the spectrum and help emphasise higher pitch harmonics. Pre-emphasise can also help to minimise the possibility of close formants merging in spectral enhancement stage. The speech signal is then processed 401 points at a time with a 100 points advance by a discrete-time filter. Each frame is Hamming-windowed for future re-synthesis procedure. Figure 4.8 is the block diagram of the operations involved in the discrete-time filtering procedure. The discrete-time filter begins as a simple delay and add to recover the stronger talker, and delay and subtract to recover the weaker talker. Given the pitch period estimate denoted by $\hat{p}_s$, the following transfer functions (Equation 4.5) proposed by Morgan *et al* (1997) are constructed to recover the two talkers' spectra.

$$\begin{aligned} H_+(z) &= (1+\alpha z^{-\hat{p}_s})/(1+\alpha), \\ H_-(z) &= (1-\alpha z^{-\hat{p}_s})(1-\alpha z^{\hat{p}_s})/(1+\alpha)^2. \end{aligned} \tag{4.5}$$

where α = .99 is the factor introduced to avoid $H_+(z)$ = 0 at $z = e^{jk2\pi\hat{p}_s}$. Morgan *et al* pointed that the second-order filter formulation for $H_-(z)$ was found to provide a broader notch at harmonic frequencies, and produced superior perceptual results.

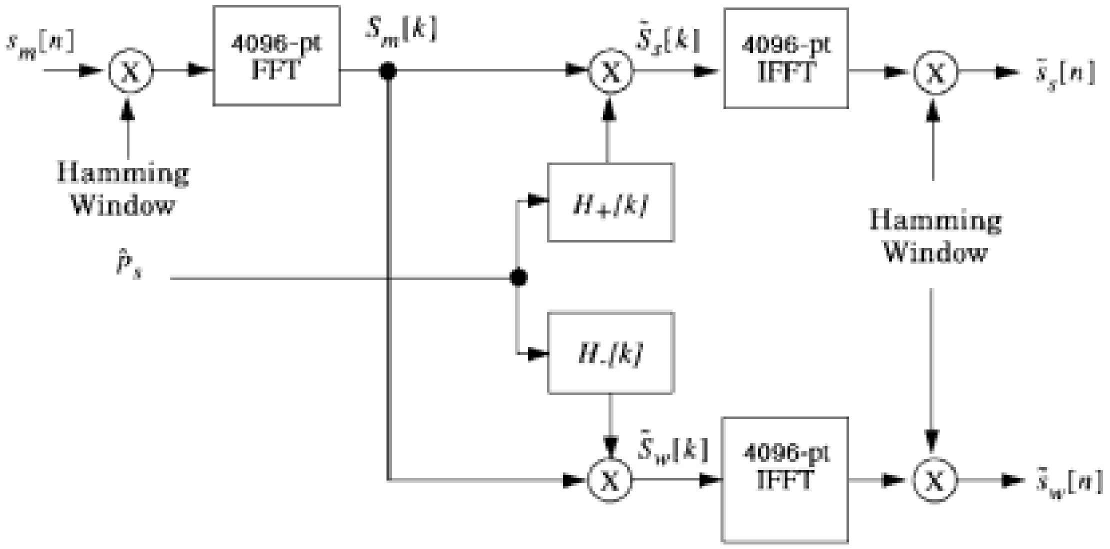


Figure 4.8 Block diagram of the operations involved in the discrete-time filtering procedure.

The discrete-time filter is in fact an adaptive comb filter (ACF) (Lim, 1978). Since voiced speech is quasi-periodic, its magnitude spectrum contains a harmonic structure. If the weaker speech does not have the same harmonic structure, its energy will be distributed between the harmonics throughout the spectrum. The essence of the discrete-time filtering is then to build a filter that passes the harmonics of the stronger speech while rejecting interfering speech or noise frequency components between the harmonics. The weaker speech recovery filter, on the other hand, rejects the harmonics of the weaker speech while passing the interfering speech frequency components. The technique is best explained by considering Figure 4.9 and Figure 4.10. In Figure 4.9 (a), the magnitude spectrum of a voiced (periodic) speech frame is shown. The transfer functions for both stronger speaker recovery and weaker speaker recovery filters are displayed in Figure 4.9 (b) and (c), respectively. The "stronger" filter has large values at the specified fundamental frequency F0 (in this case 244Hz) and its harmonics, and low values between. The "weaker" filter looks just like a vertical flip of the stronger one; so it has low values at the f0 and its harmonics, and large values between. Figure 4.10 is the result applying these filters to the spectrum shown in Figure 4.9 (a).

When implemented in frequency domain, the two transfer functions become:

$$\begin{aligned} H_+(k) &= (1+\alpha e^{-(j2\pi/M)k\hat{p}_s})/(1+\alpha), \\ H_-(k) &= (1-\alpha e^{-(j2\pi/M)k\hat{p}_s})(1-\alpha e^{(j2\pi/M)k\hat{p}_s})/(1+\alpha)^2. \end{aligned} \tag{4.5}$$

where M is the point of used DFT. We could see that the linear phase factor $e^{-(j2\pi/M)k\hat{p}_s}$ determines the value of the function. Since for non-integer $\hat{p}_s$ values, $e^{j2\pi\hat{p}_s} \neq 1$ in general, to delay a real signal by a non-integer value of $\hat{p}_s$, the linear phase factor must therefore be implemented differently. For an even value of M, the linear phase factor is:

$$\begin{cases} e^{-(j2\pi/M)k\hat{p}_s} & if\, 0 \le k \le M/2 \\ e^{-(j2\pi/M)(k-M)\hat{p}_s} & if\, M/2+1 \le k \le M-1 \end{cases}$$

When a 4096-point FFT is used and the speech sample rate is 20 KHz, we get a frequency resolution $\Delta f = fs/M = 4.88Hz$. So each sample in frequency domain represents the frequency

$$f_0 = index\_f_0 \times \Delta f = index\_f_0 \times fs/M$$

where $index\_f_0$ is the corresponding index of each frequency in frequency domain. We can then deduce:

$$index\_f_0 = M/(fs/f_0) = M/\hat{p}_s \text{ or}$$

$$\hat{p}_s/M = index\_f_0 \tag{4.6}$$

where $\hat{p}_s$ is the pitch period estimate in samples. Combining Eq. (4.5) and (4.6) we can get:

$$\begin{aligned} H_+(k) &= (1+\alpha e^{-j2\pi k\cdot index\_f_0})/(1+\alpha), \\ H_-(k) &= (1-\alpha e^{-j2\pi k\cdot index\_f_0})(1-\alpha e^{j2\pi k\cdot index\_f_0})/(1+\alpha)^2. \end{aligned} \tag{4.7}$$

From Eq. (4.7) we can see $H_{+}(k)$ achieves maximal value at multiple $index\_f_{0}$ positions, which correspond to the harmonics positions. Meanwhile, $H_{-}(k)$ achieves minimal value at these positions.

Ideally, spacing between each "tooth" in the discrete-time filter should correspond to the fundamental frequency in Hz and should remain constant throughout the voiced section of speech. Unfortunately, speakers normally vary their pitch and therefore require the filter to adapt as data are processed. For each analysis frame, we use corresponding pitch estimate to construct the transfer functions for the filters. As the frame advances, we have to reconstruct the transfer functions according to each pitch estimate.

Referring to Figure 4.8, we perform an 4096-point FFT to each analysis frame to get $S_{m}[k]$, where m is the first sample in the frame. Given a pitch period $\hat{p}_{s}$, the stronger and weaker talkers' spectra are recovered by constructing $H_{+}(k)$ and $H_{-}(k)$ according Eq (4.7) and multiplying by $S_{m}[k]$ to produce $\tilde{S}_{s}[k]$ and $\tilde{S}_{w}[k]$.

One of the disadvantages of the discrete-time filter approach is that it is extremely dependent upon an accurate estimate of the stronger talker's true pitch. We found that estimates of the pitch harmonics were often off by more than one DFT bin at higher frequencies. Thus we need an accurate pitch estimate to ensure that the location of higher order harmonics is accurate to within one DFT bin. Our YIN pitch estimate can supply a fractional pitch estimate, i.e., the precision is less than 1Hz. This precision is adequate enough to ensure the location of higher order harmonics is within one DFT bin.

Since the discrete-time filter can only be used to enhance voiced speech, a method must be available with which to handle unvoiced speech or silence section. Two approaches are typical. First, we can pass the unvoiced speech through the filter unprocessed. In this case, a scaling term (typically in the range of 0.3-0.6) is necessary because applying the discrete-time filter to voiced speech reduces the energy present. Failure to apply attenuation in unvoiced or silence sections results in unnatural emphasis of unvoiced speech sounds with respect to voiced sounds. The second method for processing unvoiced speech is to maintain a constant pitch period, obtained from the last voiced speech frame, and process the unvoiced sounds or silence as if they were voiced. Deller *et al*. (1993) pointed out that the first method is more successful than the second. In our system, we choose a scaling term 0.6 for stronger speaker recovery and 0.3 for weaker speaker recovery. This is because more energy is suppressed in the weaker speaker recovery procedure.

After all the modifications to the spectrum are completed, the recovered signals are then converted back to the time domain. A 4096-point IFFT is then applied to produce the estimate of both the stronger speech and the weaker speech using overlap-add synthesis. This technique results in a great improvement of the perceptual quality of the reconstructed speech.

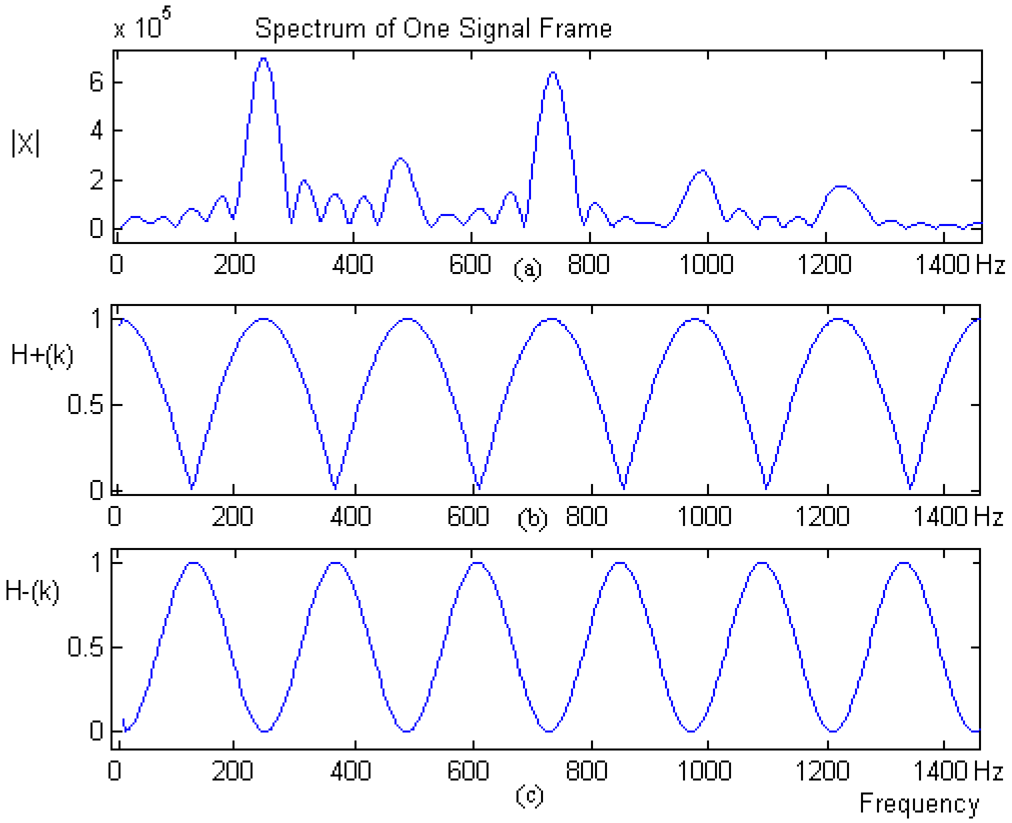


Figure 4.9 Plotting of transfer functions of the discrete-time filters

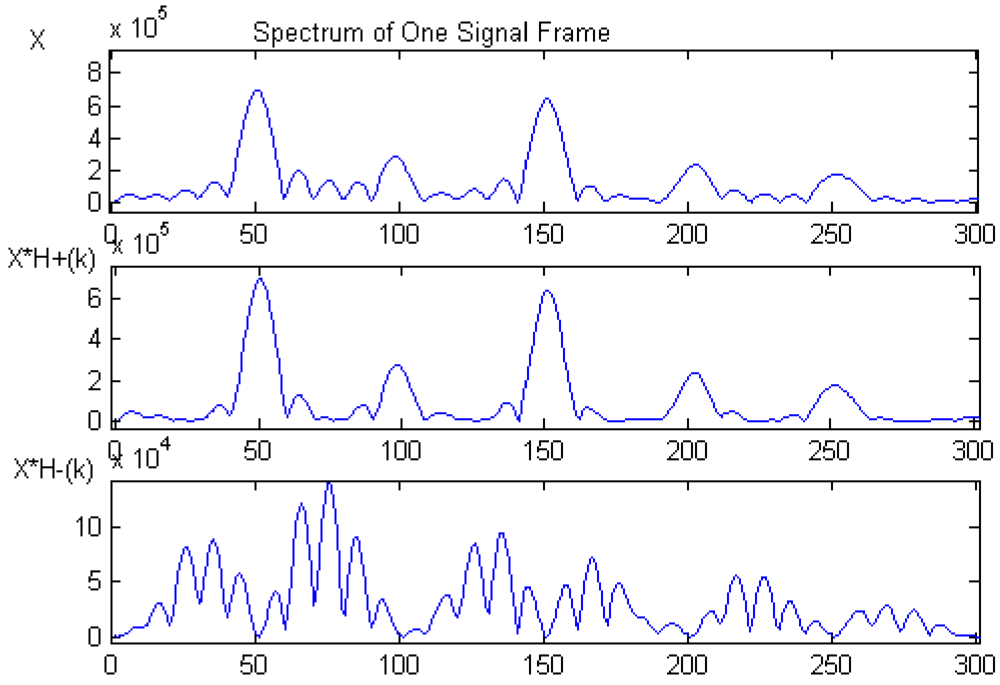


Figure 4.10 Spectra of the same frame after applying the discrete-time filters

C. *Spectral Enhancement*

Referring to Figure 4.7, having recovered the raw stronger and weaker talker signals, we perform several nonlinear post-processing steps to further enhance the recovered signals. A zero-phase Hamming window is first applied to make sure the frequency response of the window is “totally real even.” This prevents the window from influencing the phase response of the recovered signals during subsequent processing. To further reduce the residue effects of weak talker interference, the stronger signal $\tilde{s}_s[n]$ is enhanced by “sampling” the harmonics. A 4096-point FFT is used to obtain a frequency resolution of 4.88 Hz (assuming the sample rate is 20 kHz). The bandwidth of each sampled harmonic is empirically set to 53.7 Hz, that is, only the energy associated with the frequency bins closest to the harmonics, and at the five adjacent bins on either side is retained in frequency domain. The energy not associated with these bins is discarded.

The energy associated with the formant frequency bins is also retained. Twelfth-order linear prediction analysis is employed to determine the location of the stronger talker’s first three formants. Formants can be estimated from the predictor parameters in one of two ways. The most direct way is to factor the predictor polynomial and, based on the roots obtained, try to decide which are formants, and which correspond to spectral shaping poles (Markel, 1973). The alternative way of estimating formants is to obtain the LPC spectrum, and choose the formants by a peak picking method (McCandless, 1974). A distinct advantage inherent in the linear predictive method of formant analysis is that the formant centre frequency and bandwidth can be determined accurately by factoring the predictor polynomial. Since the predictor order *p* is chosen *a priori*, the maximum possible number of complex conjugate poles that can be obtained is *p/2*. Since 12-order linear prediction analysis is performed, we can use the method proposed by Markel (1973) to determine the location of the stronger talker’s first three formants.

The energy in the bins not associated with the formants is then discarded. A similar process is also used to enhance the weaker speech. Residue signal is used to enhance weaker signal by suppressing the stronger energy at harmonics and formants. In this case the energy associated with the harmonics and stronger signal’s formants is discarded. However, we found that the same bandwidth of each harmonic and formant frequency is not wide enough for the weaker speech enhancement. This is because the energy of the stronger speech may spread as wide as 100 Hz at each harmonic. Discarding energy in only half of these bins can still leave more energy of stronger speech than weaker speech, causing the recovered weaker speech less perceptual. A bandwidth of 107.4 Hz is used instead for the weaker speech enhancement.

An IFFT is then used to reconstruct one talker’s speech signal using overlap-add synthesis, which result in an improvement of the perceptual quality of the reconstructed speech.

It is known that when two equal bandwidth signals are added, such a separation is not possible as one voice may mask the other. Even if the two signals do not have equal bandwidth, there is still a problem if they have overlapping harmonics. Harmonics overlapping is illustrated in Figure 4.11. Figure 4.11 (a) is the spectrum of one frame in a linearly added speech with TIR = 12 dB. Figure 4.11 (b) and (c) is the spectra of same frames in stronger signal and weak signal. The fundamental frequencies of these two signals are 306.58 Hz and 117.57 Hz, respectively, produced by our YIN pitch estimator. We should note the second harmonic of the stronger signal is 613 Hz and the five harmonic of the weaker signal is 588 Hz. Both these harmonics contain more energy than other frequencies. Because their variance is only 25 Hz, when we sample the harmonics of stronger signal using a bandwidth 50 Hz, we also retain the energy of the weaker signal. Meanwhile, when the energy associated with these frequency bins is removed, both talkers' speech signals lose the energy (remember the original weaker speech also contains much energy at these frequency bins). Thus the recovered speech is of less naturalness. Figure 4.11 (d) and (e) illustrate this phenomenon. (d) is the spectrum of recovered stronger signal and (e) is the spectrum of recovered weaker signal. We can see in (d) that the information around 600 Hz is lost, which is very strong in original spectrum (c).

To solve this problem pure signal processing technique might be inadequate. A solution to this might be considered to require source-specific knowledge, but this is often impossible in realistic situation. Cooke and Brown (1993) proposed a computational auditory scene analysis (CASA) (Brown, 1992; Cooke, 1993; Brown and Cooke, 1994) which exploits principles of perceived continuity. This method uses cues provided by primitive grouping processes such as harmonicity to restore the missing harmonic fragment. However, at the same time the method also raises another problem. As it is done according to a principle of pitch contour similarity, to restore the missing harmonic fragment in the recovered weaker signal we will need the pitch information of the weaker signal. Wu, *et al.* (2003) proposed a new multi-pitch tracking algorithm which uses a hidden Markov model for forming continuous pitch tracks. We could use this algorithm to get pitch contours of both stronger signal and weaker signal at the beginning. Our system did not involve these approaches; so we do not know their performance in the case of cochannel speaker separation. The harmonics-overlapping problem remains unresolved in our system.

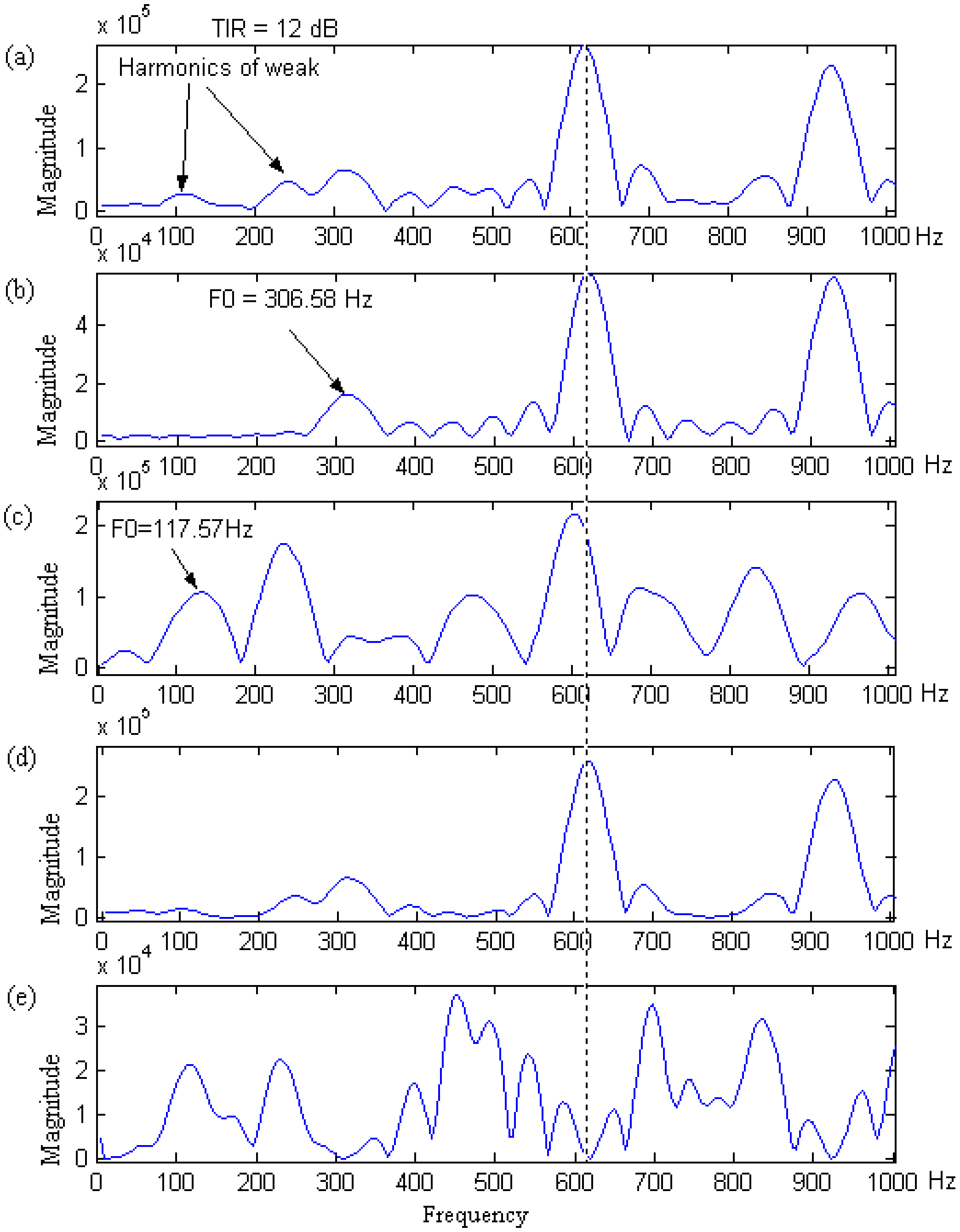


Figure 4.11 Illustration of harmonic overlapping in cochannel speech separation

(a) Spectrum of a linearly added signal with a TIR = 12 dB. (b) Spectrum of the stronger signal. (c) Spectrum of the weaker signal. (d) Spectrum of the recovered stronger signal. (e) Spectrum of the recovered weaker signal.

4.5 Speaker Assignment

The final stage in our system is the speaker assignment algorithm. This algorithm receives the stronger and weaker recovered signals $\hat{s}_s[n]$ and $\hat{s}_w[n]$ and assigns them to output signals $\hat{s}_1[n]$ and $\hat{s}_2[n]$. Assignment is driven on a frame-by-frame basis by a binary sequence denoted as A[n]. We term the case when A[n]=0 as the “no swap” state, in which case $\hat{s}_s[n]$ is assigned to $\hat{s}_1[n]$ and $\hat{s}_w[n]$ is correspondingly assigned to $\hat{s}_2[n]$. A[n]=1 is defined as the "swap" state, in which case the assignment is reversed.

Morgan *et al.* (1997) proposed a Maximum Likelihood Speaker Assignment (MLSA) algorithm for assigning the recovered signals, which exploits an ML formulation and allows for error recovery using the Viterbi algorithm. One advantage is that it uses several frames to make a decision and ties swapping decisions directly to an objective hypothesis testing method. Thus it is relatively robust to the effects of additive noise and pitch tracking errors. However, its biggest disadvantage is that when a mistake is made, it is propagated infinitely. For example, if talker 1 is placed on channel 1, and an assignment error occurs, talker 1 will appear on channel 2 until another error occurs.

As stated in Section 3.3D, currently the speaker assignment is not included in our system. This is because the performance of speaker assignment algorithm is not good enough and largely depends on the pitch tracks of both stronger and weaker talkers. One way to track the weaker talker's pitch is to use two-talker pitch tracking algorithms (Zissman and Seward, 1992; Rosier and Grenier, 2002; Wu *et al*, 2003). Another way is to estimate the pitch of the recovered weaker speech, and then consider it as the weaker talker's pitch. This method is proposed by Morgan *et al* (1997) and works not very well because the periodic structure in recovered weaker speech is often extremely suppressed as the energy of stronger speech is removed. Whichever method is used, the pitch-tracking task is rather difficult, and causing the performance of speaker assignment algorithm is very unsteady.

However, we found without the speaker assignment the outputs of our system are still acceptable. Only if there exist some speech segments where the target speaker is silence and the interfering speaker is voiced, the speaker assignment is necessary. In final tests, we could use the prior pitch to provide optimal assignment of the speech.

Chapter 5 Implementation and Testing

5.1 Matlab Implementation

For easy demonstration, the whole system is implemented in Matlab with signal processing toolbox. A simple graphic user interface (GUI) is available. Testing is also done in Matlab. A sample rate of 20 kHz is assumed. Frame size is 400 samples; corresponding to 20 ms. Frame increment is 100 samples. All speech signals are re-sampled to 20 kHz before being processed. A 4096-point FFT is used, giving a frequency resolution as low as 4.88 Hz.

The cochannel signal is first read into memory and is padded with zeros so its length is exactly multiple frame sizes. The processing procedure is multi-passed, that is, we finish each step in a separate pass rather than a same pass. However, in each pass, the signal is processes frame-by-frame with same frame size and frame increment.

A. Mixing Cochannel Signals

The cochannel signal is linearly added given a TIR. Following code is used:

```
function mix = addSignals(s,n,snr)
%ADDSIGNALS adds signal s to signal n at specified SNR. If n is not long
% enough, replicate it

s = s(:); n = n(:);
while length(n) < length(s)
  n = [n;n];
end
n = n(1:length(s));
es = sum(s.^2);  ns=sum(n.^2);
k = sqrt((es/ns)*(10.^(-snr/10)));
mix = s+k.*n;
```

B. Pre-emphasis of Speech

Speech signals have a spectrum that falls off at high frequencies. In our cochannel speaker separation system, it is desirable that this high-frequency falloff be compensated by "pre-emphasis". A simple and widely used method of pre-emphasis is linear filtering by a "first difference" filter of the form:

$$y[n] = x[n] - \alpha \cdot x[n-1]$$

where $x[n]$ is the input speech signal and $y[n]$ is the output "pre-emphasised speech" and α is an adjustable parameter.

We can use the Matlab function `filter()` to implement the pre-emphasis filter.

```
y = filter([1 -a],1,x);
```

where, a is the adjustable parameter, and x is the input speech signal. We use a = 0.95 in our system.

C. YIN Pitch Estimator

We implement the YIN algorithm described in Cheveigne and Kawahara (2002) step by step. Though our system is just for demonstration, we found Matlab is too slow when calculating the pitch estimates, especially if we set the pitch search upper limit to a very large value (e.g. a quarter of the sample rate). Benefiting Matlab's powerful functionality, we can call compiled C code from Matlab (as a dynamic link library). The entire core computing functions are implemented in C code as Matlab Mex files because of the high cost of computation. We found the C code is ten times faster than the same Matlab code.

The following parameters are used in the implementation of YIN pitch estimator:

- Minimum search frequency = 30 Hz
- Maximum search frequency 500 Hz
- Aperiodicity ratio threshold = 0.15
- Computation buffer size = 5000 samples
- Window shift = 100 samples
- Integration window size = 25 ms, but at least 300 samples

The meaning of these parameters can be found either in previous section or in the original YIN paper (Cheveigne and Kawahara, 2002).

5.2 Testing

A. Human Listening Tests

The goal of the human listening tests was to transcribe a target digit string speech when jammed by a stronger interfering speech signal, and use these transcriptions to determine the performance difference between processed and unprocessed cochannel speech. Eight untrained listeners were asked to transcribe twelve digit strings with different TIR at -12 dB and -18 dB.

Two sentences were selected from the TIMIT database as interfering speech. One is uttered by a male speaker and the other is uttered by a female speaker. The whole sentences are:

Female:	Don't ask me to carry an oily rag like that
Male	She had your dark suit in greasy wash all year

Twelve digit strings were selected from the TIDIGITS database, which are uttered by different male speakers. Each TIDIGITS string was considered as a target signal and the TIMIT sentences were considered as interferers. They were linearly added at -12 dB, -18 dB TIR's, forming 48 different cochannel speech signals. Speech was presented to each listener in the following order: two repetitions of the interferer's clean speech to let the tester get familiar with the interferer. Three unprocessed cochannel signals followed by three processed signals at -12 dB TIR are then presented. Each time the listener was asked to input the heard digits into the testing program. The valid input is "123456789oz", in which 'z' means zero. After that, the other three unprocessed and three processed signals at -18 dB are played to the listener.

The testing program will save the testing result to hard disk. The result consists of two columns: one for the listener's transcript and the other for the correct answers. The result is then passed to a scoring program for analysis. The scoring program compares the listener's transcript with correct answers to get its accuracy. There are three types of error a tester can make: insertion (*ins*), substitution (*sub*), and deletion (*del*). The correct transcribed digit is counted as *hits*. The accuracy is calculated according following formula:

$$accuracy = 100 \times (hists - ins)/(hits + dels + subs)$$

B. Pitch Estimation Tests

The pitch estimate of cochannel speech was compared with the estimated pitch of the stronger speech within appropriate regions. The appropriate regions are defined as following: pitch estimate between 50 – 400 Hz is considered as voiced. Only the regions where the pitch estimates of both speech signals are voiced are considered as appropriate regions. At each frame if the pitch variance between target and cochannel speech signals is less than 2% of target pitch value, the pitch of cochannel speech is considered correct. Since the TIR's vary significantly between different frames, and the pitch detector has its own error, there cannot be 100% correct pitch estimate.

C. Voiced/Unvoiced Detection Tests

In order to make our voiced/unvoiced detector work, a training set of data is required to obtain the mean vector and the covariance matrix for each class (Silence, Unvoiced and Voiced). The raw training set was pre-processed (e.g., high-pass filters and data/shift windows). The training set is then analyzed by manually segmenting natural speech into regions of silence, unvoiced speech, and voiced speech. The speech segments are sub-divided into 10-ms blocks (i.e., 200 samples in each block on speech of 20K sampling rate), and the set of 5 different measurements defined in previous section is made on each block of data.

Chapter 6 Results and Discussion

6.1 YIN Pitch Estimation

To examine how close the correct pitch estimate of the cochannel speech is to the target one, we also produced the average variance between them. Table 6-1 and Table 6-2 show the results. The test results are also showed in Figure 6.1, where the pitch track for the cochannel speech is superimposed on the one for the target speech. The pitch contour for the target digit string (o8o6255a.wav) is drawn in solid line, and the pitch contour for the mixed speech (with the female spoken sentence) is drawn in dashed line.

Table 6-1 The Accuracy of YIN Pitch Detector for Cochannel Pitch Tracking in Voiced Region. Interferer is a female spoken sentence n8.

Target / n8.Z	Average Variance of Correct Pitch / Average 2% Variance (Hz)									
(male)	0dB		5dB		10dB		15dB		20dB	
o8o6255a.wav	0.31	2.18	0.23	2.18	0.17	2.17	0.10	2.17	0.06	2.16
1871986a.wav	0.27	2.92	0.21	2.91	0.12	2.90	0.11	2.91	0.08	2.91
19z96z8a.wav	0.24	2.80	0.19	2.80	0.18	2.82	0.13	2.83	0.08	2.83
19974a.wav	0.35	2.29	0.24	2.27	0.17	2.28	0.18	2.32	0.12	2.33
2567184a.wav	0.30	3.07	0.26	3.05	0.19	3.03	0.14	3.01	0.07	3.00
27984a.wav	0.28	2.39	0.23	2.39	0.27	2.39	0.11	2.42	0.10	2.45
34187a.wav	0.27	2.18	0.20	2.15	0.14	2.13	0.15	2.11	0.12	2.10
42o45a.wav	0.25	2.09	0.23	2.06	0.20	2.03	0.17	2.05	0.12	2.05
4379315a.wav	0.34	2.86	0.24	2.87	0.20	2.87	0.17	2.86	0.12	2.87
7551322a.wav	0.32	2.30	0.23	2.31	0.18	2.31	0.15	2.33	0.12	2.34
9oo6o43a.wav	0.28	2.26	0.24	2.29	0.17	2.30	0.11	2.27	0.09	2.32
92o1648a.wav	0.31	2.32	0.23	2.30	0.15	2.28	0.11	2.27	0.08	2.26
Average	0.29	2.47	0.22	2.46	0.18	2.46	0.13	2.46	0.09	2.46

Table 6-2 The Accuracy of YIN Pitch Detector for Cochannel Pitch Tracking in Voiced Region. Interferer is a female spoken sentence n9.

Target / n9.Z	Average Variance of Correct Pitch / Average 2% Variance (Hz)									
(female)	0dB		5dB		10dB		15dB		20dB	
o8o6255a.wav	0.25	2.18	0.20	2.19	0.12	2.20	0.07	2.17	0.05	2.16
1871986a.wav	0.33	2.97	0.33	2.98	0.22	2.94	0.15	2.94	0.10	2.94
19z96z8a.wav	0.24	2.73	0.17	2.77	0.13	2.80	0.15	2.81	0.12	2.79
19974a.wav	0.42	2.24	0.38	2.27	0.33	2.31	0.18	2.31	0.13	2.31
2567184a.wav	0.39	3.02	0.19	3.02	0.11	3.00	0.13	3.04	0.09	3.03
27984a.wav	0.33	2.38	0.26	2.38	0.18	2.38	0.11	2.39	0.09	2.42
34187a.wav	0.14	2.10	0.18	2.13	0.14	2.15	0.10	2.12	0.12	2.11
42o45a.wav	0.25	2.07	0.16	2.09	0.09	2.08	0.07	2.05	0.06	2.05
4379315a.wav	0.39	2.76	0.23	2.82	0.14	2.82	0.09	2.82	0.07	2.83
7551322a.wav	0.36	2.30	0.35	2.30	0.30	2.30	0.18	2.33	0.14	2.34
9oo6o43a.wav	0.34	2.25	0.37	2.32	0.22	2.32	0.15	2.34	0.10	2.35
92o1648a.wav	0.27	2.35	0.20	2.34	0.14	2.31	0.10	2.30	0.08	2.30
Average	0.30	2.44	0.25	2.46	0.17	2.46	0.12	2.46	0.09	2.46

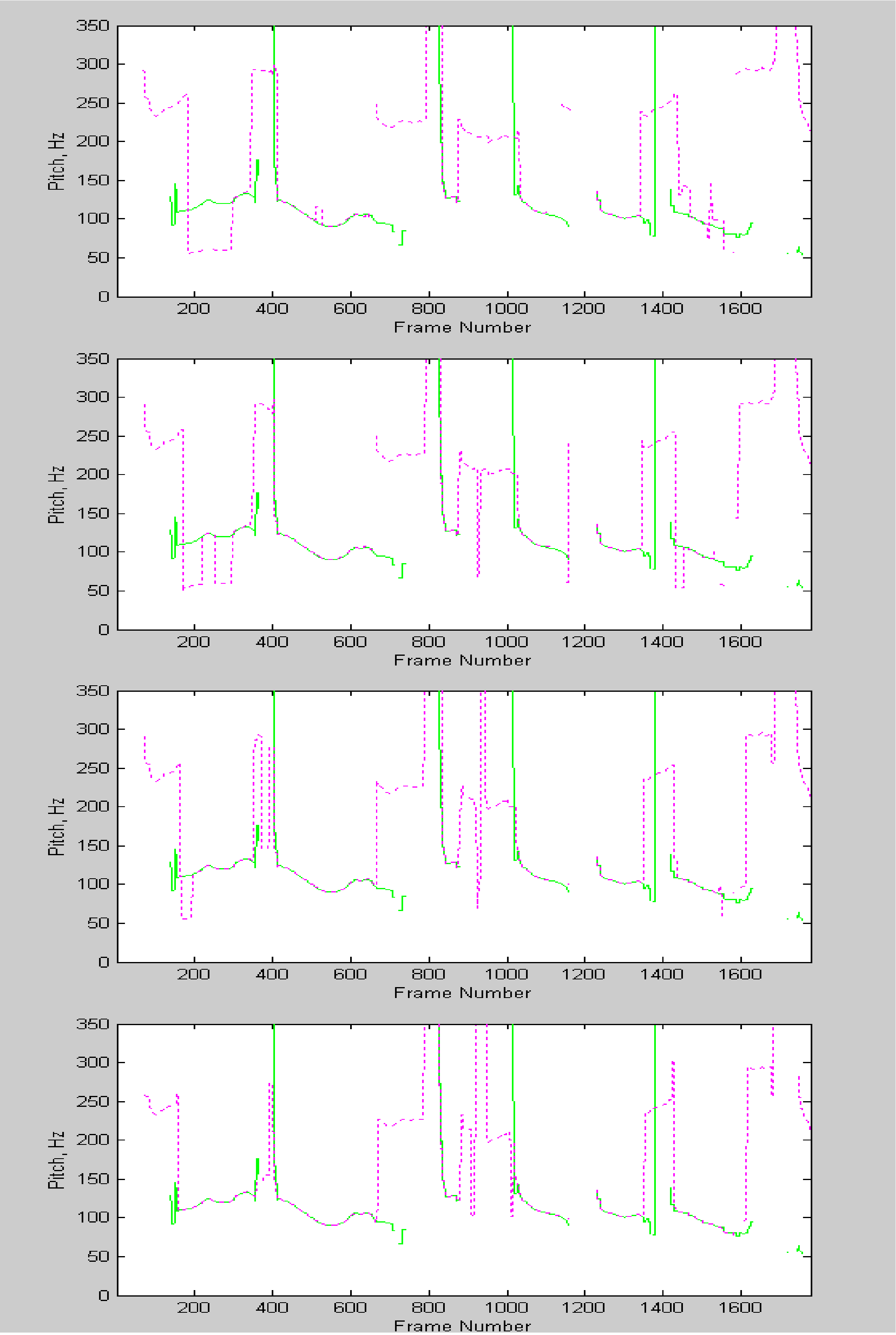


Figure 6.1 YIN pitch detector tracking the cochannel pitch superimposed on the pitch tracks of the target talkers at 0–15 dB TIR’s, respectively. Solid line is target pitch, dashed line is cochannel pitch

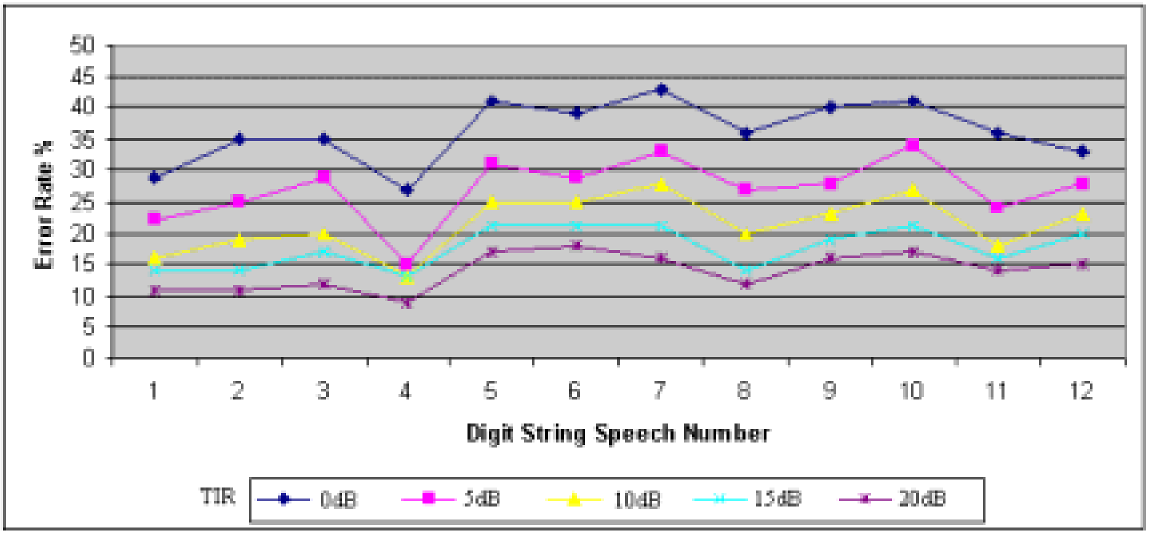


Figure 6.2 The error rate of YIN pitch detector for cochannel pitch tracking at different TIR's with a male spoken sentence as the interference.

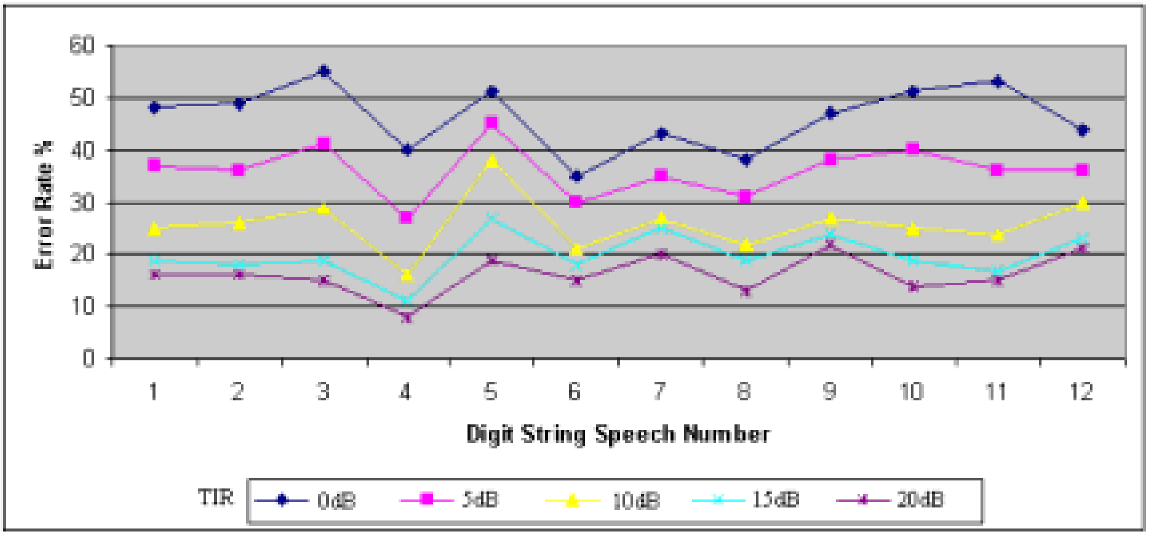


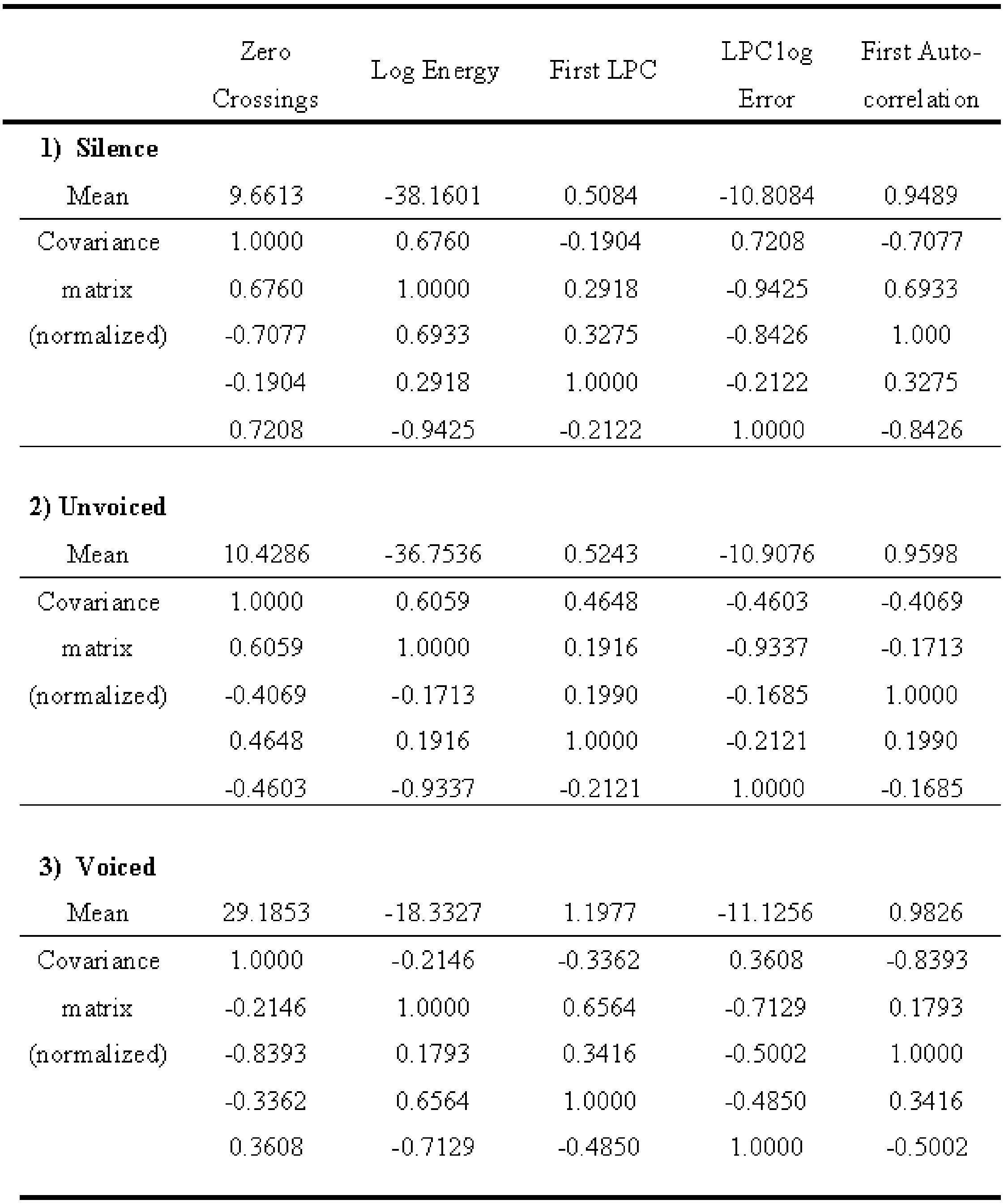
Figure 6.3 The error rate of YIN pitch detector for cochannel pitch tracking at different TIR's with a female spoken sentence as the interference.

The result showed YIN is robust in noisy conditions and capable to estimate the pitch period of the stronger talker from the cochannel speech. When the TIR is high (e.g., above 6 dB), YIN is able to exactly estimate the pitch of the stronger talker. When the TIR is low (e.g. below 6 dB), YIN tends to track the pitch of either the target talker or the interfering talker. This phenomenon is expected and desired, but requires a robust speaker assignment algorithm as the same time.

6.2 Voiced/Unvoiced Detection

Table 6-3 shows the means, and the normalized covariance matrices for the three classes for a typical set of training data. The columns in Table 6-3 correspond to the five feature measurements discussed in Section 4.2. The off-diagonal terms of the covariance matrices are a measure of the correlation between the different parameters. If the measurements were all independent and uncorrelated, then all off-diagonal elements would be 0. It can be seen that the magnitudes of the off-diagonal elements vary from 0.18 to 0.94, indicating varying degree of correlations between the different parameters.

Table 6-3 Means and Covariance matrices for the three classes for the training data



	Zero Crossings	Log Energy	First LPC	LPC log Error	First Auto-correlation
1) Silence					
Mean	9.6613	-38.1601	0.5084	-10.8084	0.9489
Covariance	1.0000	0.6760	-0.1904	0.7208	-0.7077
matrix	0.6760	1.0000	0.2918	-0.9425	0.6933
(normalized)	-0.7077	0.6933	0.3275	-0.8426	1.000
	-0.1904	0.2918	1.0000	-0.2122	0.3275
	0.7208	-0.9425	-0.2122	1.0000	-0.8426
2) Unvoiced					
Mean	10.4286	-36.7536	0.5243	-10.9076	0.9598
Covariance	1.0000	0.6059	0.4648	-0.4603	-0.4069
matrix	0.6059	1.0000	0.1916	-0.9337	-0.1713
(normalized)	-0.4069	-0.1713	0.1990	-0.1685	1.0000
	0.4648	0.1916	1.0000	-0.2121	0.1990
	-0.4603	-0.9337	-0.2121	1.0000	-0.1685
3) Voiced					
Mean	29.1853	-18.3327	1.1977	-11.1256	0.9826
Covariance	1.0000	-0.2146	-0.3362	0.3608	-0.8393
matrix	-0.2146	1.0000	0.6564	-0.7129	0.1793
(normalized)	-0.8393	0.1793	0.3416	-0.5002	1.0000
	-0.3362	0.6564	1.0000	-0.4850	0.3416
	0.3608	-0.7129	-0.4850	1.0000	-0.5002

The algorithm has been tested on training data and testing data respectively. The speech data in the training set consisted of utterance "Don't ask me to carry an oily rag like that" spoken by a female speaker. The testing set was used to evaluate the performance of the algorithm. The speech data in the testing set consisted of utterance "She had your dark suit in greasy wash all year" spoken by a male speaker.

The confusion matrix was used to evaluate how well the algorithm performs with the speech data. The algorithm was first run on the training set itself and then was used on the speech data in the test set. The confusion matrices for the two cases are presented in Table 6-4 and

Table **6-5**. Most of the identifications are correct, a few errors occurred at the boundaries between the different classes. Since the classification was made on the basis of consecutive 10-ms long speech segments, a segment at the boundary often included data from two classes.

Table 6-4 Matrix of incorrect identifications for the three classes for the speech data in the *training* set.

Actual class Identified as	Silence	Unvoiced	Voiced
Silence	61	3	0
Unvoiced	2	56	1
Voiced	0	0	253
Total	63	59	254

Table 6-5 Matrix of incorrect identifications for the three classes for the speech data in the *testing* set

Actual class Identified as	Silence	Unvoiced	Voiced
Silence	23	1	0
Unvoiced	2	35	1
Voiced	0	0	190
Total	25	36	191

6.3 Cochannel Speaker Separation

The human listening test result is listed in Table 6-6. Before being present to listeners, the twelve digit strings are randomly sorted to ensure the digit strings that have been presented to the listener are not repeated again to the same listener. The results show that cochannel processing is helpful at very low TIR's for linearly added speech. We should note the listener expressed a very good ability to catch the digits. Although the results show the accuracy of processed speech transcription is a little higher than that of unprocessed cochannel speech, the tested reported that the processed signal is of less naturalness than the original signal. We believed this is because some useful periodic information of the weaker (target digit string) signal was removed when we removed the energy associated with the stronger talker's harmonics. This is the harmonics-overlapping problem.

Table 6-6 Human Listening Test Result

Test data	Acc	Hit	Sub	Ins	Del
Cochannel Speech –12 dB TIR	94.6%	144	4	4	0
Cochannel Speech –18 dB TIR	85.5%	132	18	2	2
Processed Speech –12 dB TIR	97.4%	148	4	0	0
Processed Speech –18 dB TIR	88.5%	136	11	5	1

We also present the listeners some cochannel speech signals made by linearly adding a male utterance and a female utterance as well as both the recovered stronger signal and weaker signal. They were asked to give their opinions on whether the recovered signals are easier to catch. The results are showed in Table 6-7. The result showed the recovered stronger signal is very good and the weaker interfering signal is strongly eliminated. But the recovered weaker signal was reported of less naturalness again.

Table 6-7 Intelligibility Test Result

Target \ Interferer		Female	Male
TIR 6 dB	Female	Better / Worth	Better / Worth
	Male	Better / Worth	Better / Worth
TIR 12 dB	Female	Better / Worth	Better / Worth
	Male	Better / Worth	Better / Worth

When the TIR is 0 dB, the requirement for a speaker assignment step is obvious. We can hear that the target speech signal occurs in both recovered stronger signal and weaker signal. After manually assigning the recovered speech signals according to a prior pitch track, the result appeared much better.

Chapter 7 Conclusion

This dissertation has presented an automatic cochannel speaker separation system designed to operate on two-talker cochannel speech in both clean and noisy environments. The system has been shown to improve human recognition of processed clean cochannel speech at -12 and -18 dB TIR. The result showed our cochannel speaker separation system is helpful when the TIR is relative high.

While the system has demonstrated its effectiveness for two-talker cochannel enhancement applications, it nonetheless has some inherent problems. When both talkers have the same instantaneous pitch, or have overlapping harmonics, the algorithm will place both talkers on one output and neither talker on the other output. The problem of automatic speaker assignment also remains a significant obstacle to completely “hands-free” operation.

To solve this problem pure signal processing technique might be inadequate. A solution to this might be considered to require source-specific knowledge, but this is often impossible in realistic situation. Cooke and Brown (1993) proposed a computational auditory scene analysis (CASA) (Brown, 1992; Cooke, 1993; Brown and Cooke, 1994) that exploits principles of perceived continuity. This method uses cues provided by primitive grouping processes such as harmonicity to restore the missing harmonic fragment. However, at the same time the method also raises another problem. As it is done according to a principle of pitch contour similarity, to restore the missing harmonic fragment in the recovered weaker signal we will need the pitch information of the weaker signal. Wu, *et al.* (2003) proposed a new multi-pitch tracking algorithm which uses a hidden Markov model for forming continuous pitch tracks. We could use this algorithm to get pitch contours of both stronger signal and weaker signal at the beginning.

Although considerable work remains in developing cochannel algorithms, system achieves an effective balance between processing capability, low delay, relatively low computational complexity, and operation without *a priori* information that makes it attractive for many cochannel speech separation problems.

Reference

Arslan, L.M. and Hansen, J.H.L. (1997), "Speech Enhancement for Crosstalk Interference," IEEE Signal Processing Letters, vol. 4, No. 4.

Benincasa, D.S. and Savic, M.I. (1997), "Co-channel Speaker Separation Using Constrained Nonlinear Optimization," Proc. IEEE ICASSP, pp. 1195-1198.

Bregman, A.S. (1990), "Auditory scene analysis", MIT Press.

Brown, G.J. (1992), "Computational auditory scene analysis: A representational approach, Ph.D. Thesis, University of Sheffield.

Brown, G.J. and Cooke, M.P. (1994), "Computational Auditory Scene Analysis," Computer Speech and Language, vol. 8 no. 4, pp. 297-336.

Chazan, D., Stettiner, Y., and Malah, D. (1993), "Optimal multi-pitch estimation using the EM algorithm for co-channel speech separation," in Proc. ICASSP, Minneapolis, MN, pp. II-728–II-731.

Cooke, M.P. (1993), "Modelling Auditory Processing and Organisation", Cambridge University Press, Cambridge.

Cooke, M.P. and Brown G.J. (1993), "Computational auditory scene analysis: exploiting principles of perceived continuity," Speech Communication, 13, pp. 391-399

de Cheveigne, A. and Kawahara, H. (2001), "Comparative evaluation of F0 estimation algorithms," Eurospeech, Scandinavia.

de Cheveigne, A. and Kawahara, H. (2002), "YIN, A fundamental frequency estimator for speech and music," J. Acoust. Soc. Am., 111(4).

Deller, J.R., Hansen, J.H., and Proakis, J.G. (1993), "Discrete-time processing of speech signals," New York: Mac-millan.

Droppo, J. and Acero, A. (1998), "Maximum *A Posteriori* Pitch Tracking," ICSLP.

Gold, B. and Morgan, N. (2000), "Speech and Audio Signal Processing: Processing and Perception of Speech and Music," New York; Chichester: John Wiley.

Hanson B.A. and Wong, D.Y. (1984), "The harmonic magnitude suppression (HMS) technique for intelligibility enhancement in the presence of interfering speech," in Proc. ICASSP, San Diego, CA, pp. 18A.5.1–18A.5.4.

Hess, W. (1983), "Pitch Determination of Speech Signals", Springer-Verlag, Berlin.

Huang, J., Yen, K., and Zhao, Y. (2000), "Subband-Based Adaptive Decorrelation Filtering for Co-Channel Speech Separation," IEEE Trans. on Speech and Audio Processing, vol. 8, No. 4, pp. 402-406.

Hyvarinen, A. and Oja, E. (1999), "Independent Component Analysis: A Tutorial," Helsinki.

Lee, C.K. and Childers, D.G. (1988), "Cochannel speech separation," J. Acoust. Soc. Am., vol. 83, pp. 274–280.

Lewis, M.A. and Ramachandran, R.P. (2001), "Cochannel speaker count labelling based on the use of cepstral and pitch prediction derived features," Pattern Recognition, 34, pp. 499-507.

Licklider, J.C.R. (1951), ''A duplex theory of pitch perception,'' Experientia 7, 128–134.

Lim, J.S., Oppenheim, A.V., and Braida, L.D. (1978), "Evaluation of an adaptive comb filtering method for enhancing speech degraded by white noise addition," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp. 354–358.

Markel, J.D. (1973), "Application of a digital inverse filter for automatic formant and F0 analysis," IEEE Trans. on Audio and Electroacoustics, vol. 21, no. 3, pp. 149-153.

McAulay, R.J. and Quatiery, T.F. (1986), "Speech Analysis/Synthesis Based on Sinusoidal Representation," IEEE Trans. on Acoust., Speech and Signal Proc., vol. ASSP-34, no. 4.

McCandless, S.S. (1974), "An algorithm for automatic formant extraction using linear predication spectra," IEEE Trans. on Acoust., Speech and Signal Proc., vol. ASSP-22, no. 2, pp. 135-141.

Morgan, D.P., George, E.B., Lee, L.T., and Kay, S.M. (1997), "Cochannel speaker separation by harmonic enhancement and suppression," IEEE Trans. on Speech and Audio Processing, vol. 5, No. 5, pp. 407-424.

Naylor J.A. and Boll, S.F. (1987), "Techniques for suppression of an interfering talker in co-channel speech," in Proc. ICASSP, Dallas, TX, pp. 205–208.

Naylor, J.A. and Porter, J. (1991), "An effective speech separation system which requires no *a priori* information," in Proc. ICASSP, Toronto, Ont., Canada, pp. 937–940.

Nilsson, M., Soli, S.D., and Sullivan, J.A. (1994), "Development of the Hearing in Noise Test for the measurement of speech reception thresholds in quite and in noise," J. Acoust. Soc. Am. 95, pp. 1085-1099.

Oppenheim, A.V. and Schafer, R.W. (1989), "Discrete-Time Signal Processing," Englewood Cliffs, NJ: Prentice Hall.

Parsons, T.W. (1976), "Separation of speech from interfering speech by means of harmonic selection," J. Acoust. Soc. Amer., vol. 60, pp. 911–918.

Plomp, R. and Mimpen, A.M. (1979), "Improving the reliability of testing the speech reception threshold for sentences". Audiology 18, pp. 43-52.

Quatieri, T.F. and Danisewicz, R.G. (1990), “An approach to co-channel talker interference suppression using a sinusoidal model for speech,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, pp. 56–69.

Rabiner, L.R. and Schafer, R.W. (1978), “Digital Processing of Speech Signals,” Prentice-Hall, Englewood Cliffs, NJ.

Rosier, J. and Grenier, Y. (2002), “Two-pitch Estimation for Co-channel Speakers Separation,” in Proc. ICASS.

Ross, M.J., Shaffer, H.L., Cohen, A., Freudberg, R., and Manley, H.J. (1974), “Average magnitude difference function pitch extractor,” IEEE Trans. Acoust., Speech, Signal Processing, ASSP, vol. 22, pp.353-362.

Rouat, J., Liu, Y.C., and Morissette, D. (1997), “A pitch determination and voiced/unvoiced decision algorithm for noisy speech,” Speech Communication, vol. 21, pp. 191-207.

Seltzer, M.L. (2000), “Automatic Detection of Corrupt Spectrographic Features for Robust Speech Recognition,” Master’s thesis, Dept. Electr. and Comp. Eng., Carnegie Mellon University.

Shields, V.C. (1970), “Separation of added speech signals by digital comb filtering,” Master’s thesis, Dept. Electr. Eng., Mass. Inst. Technol.

Tukey, J.W. (1974), “Nonlinear (nonsuperposable) methods for smoothing data,” in Proc. Eascon ’74, Washington, D.C., pp. 673.

Versfeld, N.J., Daalder, L. Festen, J.M, and Houtgast, T. (2000), “Method for the selection of sentence materials for efficient measurement of the speech reception threshold,” J. Acoust. Soc. Am. 107(3), pp. 1671-1684

Wang, D. L. (1996), “Primitive auditory segregation based on oscillatory correlation”, Cognitive Science, 20, pp. 409-456.

Weinstein, E., Feder, M., and Oppenheim, A.V. (1993), “Multi-channel signal separation by decorrelation,” IEEE Trans. Speech Audio Processing, vol.1, pp. 405-413.

Widrow, B., Grover, J.R., McCool, J.M., *et al.* (1975), “Adaptive noise cancelling: Principles and applications,” Proc. IEEE, vol. 63, pp.1692-1716.

Wise, J.D., Caprio, J.R., and Parks, T. (1976), “Maximum likelihood pitch estimation,” IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-24, pp. 418–423.

Wu, M., Wang, D.L., and Brown, G.J. (2003), “A multi-pitch tracking algorithm for noisy speech,” IEEE Trans., Speech and Audio Processing, 11 (3), pp. 229-241.

Yantorno, R.E. (1998), “Co-Channel Speech and Speaker Identification Study,” AFOSR Report.

Yen, K. and Zhao, Y. (1999), “Adaptive Co-Channel Speech Separation and Recognition,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 7, No. 2, pp. 138–151.

Zissman, M. and Seward IV, D. (1992), “Two-talker pitch tracking for co-channel interference suppression,” Tech. Rep. 951, Lincoln Labs.,Mass. Inst. Technol.