ROBUST FORMANT TRACKING FOR CONTINUOUS SPEECH

ROBUST FORMANT TRACKING FOR CONTINUOUS SPEECH WITH SPEAKER VARIABILITY

By

KAMRAN MUSTAFA, BSc.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2003) McMaster University
(Electrical Engineering) Hamilton, ON

TITLE: Robust Formant Tracking for Continuous Speech with Speaker Variability

AUTHOR: Kamran Mustafa, BSc.Eng. (University of New Brunswick)

SUPERVISOR: Dr. Ian C. Bruce

NUMBER OF PAGES: x, 164

ABSTRACT

Exposure to loud sounds can cause damage to the inner ear, leading to degradation of the neural response to speech and to formant frequencies in particular. This may result in decreased intelligibility of speech. An amplification scheme for hearing aids, called Contrast Enhanced Frequency Shaping (CEFS), may improve speech perception for ears with sound-induced hearing damage. CEFS takes into account across-frequency distortions introduced by the impaired ear and requires accurate and robust formant frequency estimates to allow dynamic, speech-spectrum-dependent amplification of speech in hearing aids.

Several algorithms have been developed for extracting the formant information from speech signals, however most of these algorithms are either not robust in real-life noise environments or are not suitable for real-time implementation. The algorithm proposed in this thesis achieves formant extraction from continuous speech by using a time-varying adaptive filterbank to track and estimate individual formant frequencies. The formant tracker incorporates an adaptive voicing detector and a gender detector for robust formant extraction from continuous speech, for both male and female speakers in the presence of background noise. Thorough testing of the algorithm using various speech sentences has shown promising results over a wide range of SNRs for various types of background noises, such as AWGN, single and multiple competing background speakers and various other environmental sounds.

ACKNOWLDGEMENTS

I would like to thank my supervisor Dr. Ian Bruce for his unwavering support and guidance throughout my Masters candidature. I will always value the excellent scientific, engineering and ethical qualities that Ian has tired to instil in me, through leading by example. I am grateful to Ian for always being accessible and for providing a friendly and stimulating environment to conduct research. I consider him to be more than just my supervisor, I consider him to be a friend.

I wish to thank my friends and colleagues, Jeff Bondy, Jennifer Ko and Andrea Mucci. I am indebted to them for their valuable suggestions, ideas and last minute help in proofreading this manuscript. I hope I can one day return the favour. I also want to thank Siddharth Das for his help in coding other formant tracking algorithms, for comparison purposes. A special thanks to my fiancée, Lubna Javed, for her love, support and understanding throughout the past two years. I am also grateful to all my other family and friends for their love and support. I want to specially thank my mother, without her prayers, encouragement and guidance I would not be here.

Finally, I would like to thank God for continuing to bless me with amazing family and friends, and providing me with countless opportunities throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

1. INTRODUCTION

Formant frequencies vary with time in speech as the vocal tract configuration changes. In order to implement Contrast Enhanced Frequency Shaping (CEFS) amplification in hearing aids for continuous speech, the second formant frequency (F2) needs to be accurately estimated for voiced speech. Accurate formant estimation for continuous speech (in real life noise environments) is a challenge because formant frequencies are not simple to track in such a dynamic environment. The formant estimation algorithm needs to be robust and be able to operate in a wide range of real-life noise scenarios. It must also be able to recover quickly if it encounters any problems and after periods of silence.

1.1. Traditional Formant Estimation Techniques and Limitations

Development of accurate formant estimation algorithms began in the 1950s. Since then numerous techniques have been proposed for formant analysis. Most of the work can be classified as frequency domain techniques (such as picking peaks in the short-time frequency spectrum) or parametric techniques (also called "analysis by synthesis") in which one generates a best match to the incoming signal based on a model of speech production. The traditional approaches to formant frequency estimation are misled by spectral peaks in unvoiced speech and perform very poorly in transient background noise. Also, these traditional algorithms are not robust and are unable to recover quickly after periods of silence. These problems limit the possible use of the traditional techniques for estimation of the second formant frequency (F2) for CEFS amplification.

Three algorithms that represent the best known formant analysis techniques have been implemented in MATLAB in order to test and compare their performance under

different conditions. Brief introductions to each of these three formant estimation techniques are presented below. Details about the algorithms, implementation, results and comparisons will be presented in later sections of this thesis.

1.1.1. Analysis by Peak Picking from Cepstrally Smoothed Spectrum

This is a frequency domain method based on analysing and picking peaks from the cepstrally-smoothed frequency spectrum of the speech signal. Cepstral smoothing (or homomorphic filtering) is a nonparametric method that attempts to remove the effects of glottal pulsing on the frequency spectrum to obtain the spectral envelope corresponding to the vocal tract frequency response [2]. The algorithm that was implemented is based on a paper by Schafer and Rabiner [8]. In this algorithm, formant frequencies are estimated from the smoothed speech spectrum by adding constraints on the formant frequency ranges and relative levels of the smoothed spectrum peaks in those frequencies ranges. The three highest peaks of the Fast Fourier Transform (FFT) of the log cepstrum envelope are typically classified as the first three formants in the short-time speech spectrum. Additional constraints allow the detection of formants where two formants are very close to each other in frequency. These peaks that are close in frequency are resolved by using the chirp Z-transform (CZT) algorithm which allows discrimination by enhancing spectral resolution at the cost of the temporal resolution [8]. More details of this algorithm are provided in Section 5.1.

The algorithm is designed to estimate the first three formant frequencies for male speakers but is restricted in the types of speech sounds from which formants can be reliably extracted. The performance of the algorithm is acceptable only for highly voiced segments of speech such as vowels. Cepstral smoothing techniques are also not robust in Additive White Gaussian Noise (AWGN) and perform very poorly in the presence of background speakers. The poor performance occurs because the presence of noise

(AWGN or from a background speaker) in frequency ranges close to the formants of the actual speech can lead to the picking of erroneous peaks. The algorithm also performs poorly for female speakers (even without noise) whose formant frequencies are spread out more than those of male speakers. A large number of logic operations are needed to constrain and refine the formant frequency estimates.

1.1.2. Linear Predictive Coefficient Analysis

Purely voiced speech signals can be modeled using an all-pole (AR) vocal tract model as described earlier. Linear prediction fits an all-pole model to voiced speech signals. The parameters of the model are indicative of formant positions, hence this is a parametric formant estimation technique. The solution of linear prediction is a difference equation which expresses each sample of the original signal as a linear combination of the preceding samples. This difference equation is called the linear predictor and the coefficients of the equation are called the linear predictive coefficients (LPC). In the implemented algorithm proposed by McCandles [9], the first three formant frequencies are estimated from the peaks of the linear prediction spectra of the speech signal. A detailed discussion of this algorithm is presented in Section 5.2.

The LPC based formant tracker is designed to track formants only in heavily voiced sounds. It performs reasonably well for purely voiced and sustained vowel-like sounds. However, in order for the algorithm to work for both male and female speakers, some of its parameters have to be modified manually. The problem of merging formant peaks still persists. When two peaks approach one another until they are sufficiently close together, the formant frequencies that are estimated might take on the same values. This problem is once again tackled using the CZT which enhances the frequency resolution for the merged peak areas. The performance of the algorithm is poor for non-vowel like sounds such as nasals and all unvoiced speech segments. It also performs poorly in the presence

of AWGN and background speakers even at relatively high Signal-to-Noise Ratios (SNRs). This occurs because the presence of noise can lead to additional peaks in the spectra that are close to the actual formant frequencies. During peak picking of the spectra, additional peaks caused by the background noise can be erroneously selected as the formant peaks and lead to incorrect formant estimates. A large number of logic operations are again needed to constrain and refine the formant frequency estimates.

1.1.3. Formant Analysis using Physiological Models of the Ear

The success of the human hearing system in understanding speech in the presence of noise and other adverse conditions has been an inspiring feature that clearly exceeds the ability of current speech recognition systems. A formant tracking algorithm has been proposed by Metz et al. based on a human auditory model which was expected to perform better in noisy conditions than the traditional speech processing techniques [10]. Spectral estimation using auditory models has shown to be efficient and robust but the success of the system depends on the accuracy and robustness of the auditory model used. The auditory model consists of stages for the outer, middle and inner ears. The output of the auditory model is the ensemble interval histogram (EIH) which shares similarities to the auditory nerve response of the mammalian ear. A detailed analysis of this algorithm is presented in Section 5.3.

The algorithm that was implemented uses the peaks of the EIH for estimating formant frequencies from voiced sounds. The three highest peaks of the EIH for each short-time speech segment are designated as the three formant frequencies for that segment. The algorithm performs very well for sustained vowel-like sounds with and without AWGN and is able to estimate the formant frequencies accurately for these types of phonemes. However, it is not accurate for any non-sustained vowel-like sound and performs very poorly in continuous speech even without any AWGN. The performance of the algorithm is very poor for all types of sounds in the presence of a background speaker.

1.2. Adaptive Pre-Filtering and the Rao & Kumaresan Approach

Rao and Kumaresan have proposed a new algorithm for formant estimation [11]. This approach involves pre-filtering speech using a bank of adaptive band-pass filters prior to spectral estimation and peak picking in each of the bands. By limiting the region of spectral estimation, the algorithm tries to reduce the effects of the other formant frequencies and the surrounding additive noise on formant frequency estimation. The peaks picked in each of the four bands correspond to the first four formant frequencies of the speech signal.

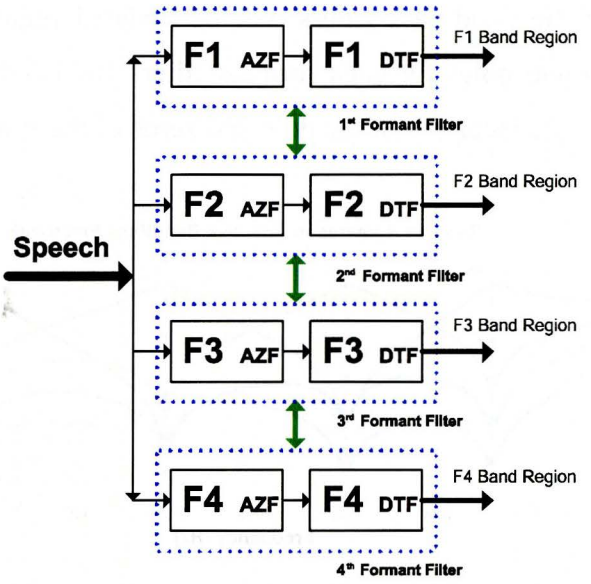

Figure 1 - 1 – The Rao and Kumaresan Adaptive Filterbank

The bank of adaptive band-pass filters is made of four filters whose frequency responses are updated regularly so that each filter's passband approximately covers a spectral region corresponding to that of one formant frequency. Each of the filters is made up of an All-Zero Filter (AZF) and an all-pole filter called a Dynamic Tracking Filter (DTF) as shown in Figure 1-1. The AZFs are each made up of three zeros whose frequency locations correspond to the other formant frequencies estimated previously for

the other band regions. The DTFs are each made up of a single pole whose location corresponds to the formant frequency estimated previously for the corresponding passband region. The combined effect of the AZF and DTF is the creation of a band-pass filter centered on the corresponding formant frequency estimated previously in that band region. The zeros from the AZF (placed at the other three formant frequency locations) help make the fall-off of the filter sharper, thereby limiting the effects of the other formant frequencies. Figure 1-2 shows the frequency and phase responses of the four band-pass filters at a given time. The formant frequencies around which the filters are centered are 700 (F1), 1500 (F2), 2500 (F3) and 3600 (F4). It is important to note that the estimates of the formant frequencies will change with time, such that the locations of the centre frequencies of the band-pass filters will be updated regularly by changing the locations of the zeros and poles for each formant filter. The bandwidth of each filter is kept constant and only the locations of the pole and zeros of the filters are changed.

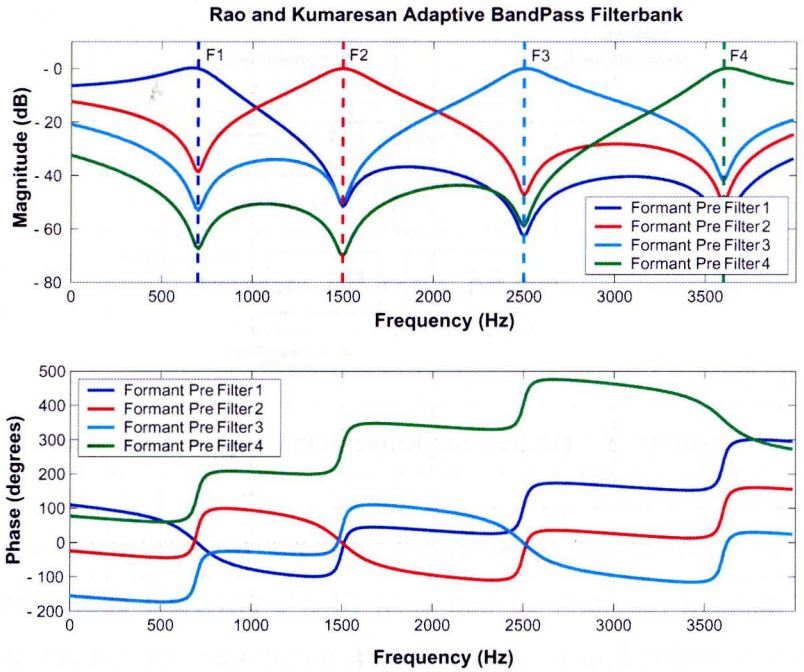

Figure 1 - 2 – Frequency response of the Rao and Kumaresan Adaptive Filterbank

The Rao and Kumaresan approach works well for estimating formant frequencies from highly voiced segments of speech and performs acceptably in AWGN. However, the algorithm is not easy to implement for real-time applications and is also not robust. It does not recover quickly after periods of silence and performs poorly during unvoiced speech segments. These factors make the Rao and Kumaresan algorithm unsuitable for implementing formant frequency estimation for CEFS amplification.

1.3. Improvements to the Rao & Kumaresan Approach

Bruce et al. [12] have proposed some improvements to the Rao and Kumaresan approach in order to overcome the problems that the algorithm encounters during unvoiced speech and periods of silence. Figure 1-3 shows the block diagram of the Bruce et al. algorithm which incorporates a voicing detector and an energy detector into the adaptive band-pass filterbank. The voicing detector allows the algorithm to detect which speech segments are voiced so that it only attempts formant estimation during the voiced speech segments. The energy detector prevents the algorithm from trying to estimate formant frequencies through spectral estimation during periods of silence or when a particular formant frequency has insufficient energy for reliable spectral estimation.

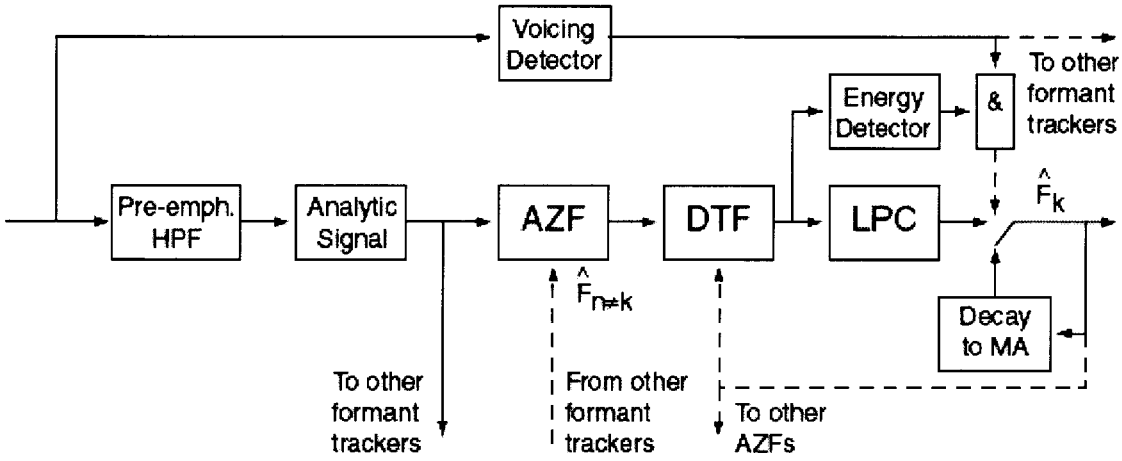

Figure 1 - 3 – The Bruce et al. Formant Estimation Algorithm (Reprinted from Bruce et al. [12])

For regions of unvoiced speech or when there is low formant energy, the algorithm uses the moving average of the previously-estimated formant frequencies instead of the current spectral estimate (obtained using 1st-order LPC). This allows the algorithm to provide acceptable formant frequency estimates for male speech in the presence of AWGN and in the presence of a single competing background speaker.

However, the Bruce et al. algorithm does not work well for continuous speech applications and for female speakers in general. The algorithm is not robust and does not recover well if there are fast formant frequency transitions e.g., when there is a switch in the speaker during a conversation or for some types of phonemes. Such scenarios cause the formant tracker to wander far away from the actual formant values. The algorithm is never able to recover after such an event. The algorithm does not perform well in fluctuating background energy levels and requires adjustments to certain parameters to work properly in different environments. Due to these issues, this algorithm cannot be used for formant frequency estimation for CEFS amplification in a real-time environment.

1.4. Contributions of this Thesis

The formant tracking algorithm being proposed in this thesis has several improvements upon the Bruce et al. scheme. These changes make it more robust and accurate in continuous speech and mitigate the effects of speaker variability and different background noises. This allows the algorithm to operate independently and provide reliable formant frequency estimates for CEFS amplification and other applications. Implementation details and a thorough analysis of the proposed algorithm are presented in Section 3 of this thesis. Figure 1-4 shows a block diagram of the Formant Tracker being proposed.

The speech signal is first pre-emphasized using a high-pass filter to equalize the energy and remove the spectral tilt of the speech signal. An approximate, analytic version of the signal is then calculated to increase spectral accuracy for the formant estimates through an approximate Hilbert transformer [11]. The analytic signal is then filtered into four different bands using a bank of adaptive band-pass filters (called *Formant Filters*). Each of the four formant filters (F1, F2, F3 and F4) in the filterbank is made up of an All-Zero Filter (AZF) and a Dynamic Tracking Filter (DTF). The zeros of each of the AZFs are set to the latest estimate of the formant frequencies from the other three bands. The DTF provides the single pole located at the latest estimate of the formant frequency for that band. This cascade arrangement results in each of the filters having a pole around its own formant frequency and zeros at the other formant frequency locations. Each of the four band-pass filters allows only the signal around the frequency region of the desired formant to pass through and suppress the other frequency regions. The formant filterbank proposed in this thesis has a fundamental modification that was not included in the Rao and Kumaresan or the Bruce et al. papers. The F1 filter of the filterbank has an added zero at the pitch frequency (F0) for further suppression of the region below the F1 frequency (the pitch region). This decreases the effects of the pitch on the F1 estimate

and improves the accuracy of the F1 estimate. The pitch estimate is provided to the F1 filter by the Gender Detector.

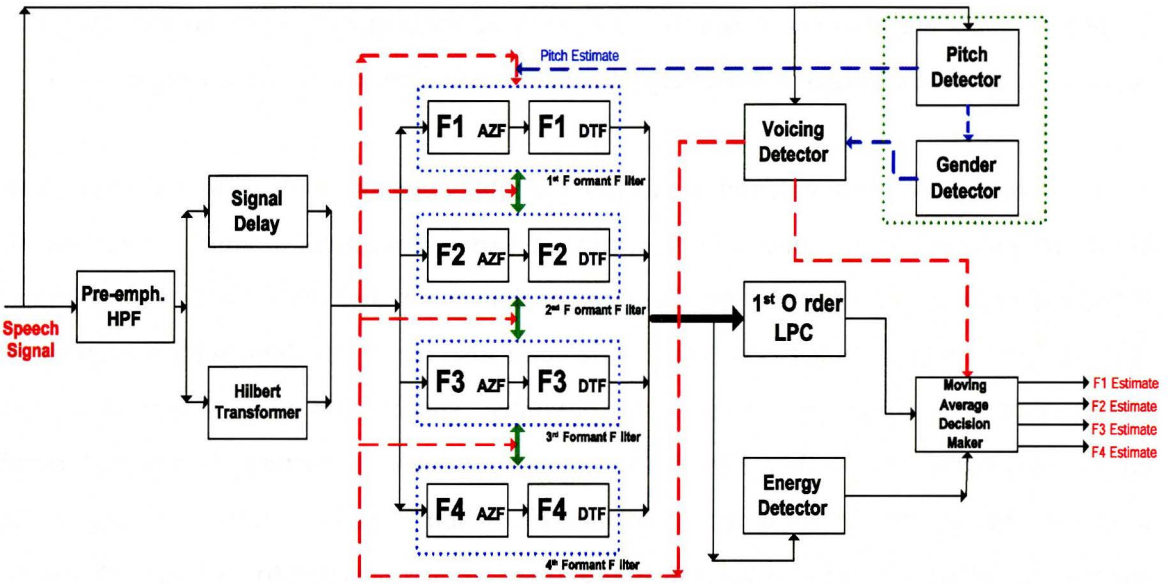

Figure 1 - 4 – The new Formant Tracker

A first-order Linear Prediction Coefficient (LPC) is then calculated for the analytic signal in each of the four bands. From each of these coefficients a formant frequency estimate is obtained. As the value of the four formant frequencies vary with time, the formant pre-filters are modified to track them by changing their pole and zero locations. Due to the band-pass pre-filtering of each formant frequency region prior to LPC, the frequency estimates provided by LPC are more accurate and the algorithm is less susceptible to errors due to background noise.

The formant estimation is further refined by adding an adaptive voicing detector to detect voiced and unvoiced speech segments. LPC estimates for the formant frequencies are only used during the voiced segments of speech. During the unvoiced speech segments or when the signal energy of a particular formant frequency region (determined by the adaptive energy detector) is below a set threshold level, the formant frequency

estimates are assigned their moving average value. This approach ensures that the formant tracker is able to recover quickly and with minimum error to the formant estimates, after unvoiced or low-energy speech segments [13]. The energy detector threshold levels are also made adaptive for each of the formant filters so that they can adjust to long term changes in the energy levels of each formant frequency region.

The voicing detector calculates the log ratio between the energy in the lower and higher frequencies of the speech to determine if a speech segment is voiced or unvoiced. If there is more energy in the lower frequencies than the higher ones, the speech segment is classified as being voiced. The voicing detector also has a threshold with hysteresis to ensure that switching from voiced to unvoiced speech (or vice versa) does not erroneously occur too quickly. Finally, an autocorrelation-based energy test is performed to ensure that voicing is not detected erroneously when there is no actual voicing in the speech but sufficient energy is present in the lower frequencies due to 'coloured Gaussian noise' (or other background noises) [13]. The voicing detector provides a sample by sample decision on whether a segment is voiced or unvoiced. The formant tracking algorithm only attempts to estimate formant frequencies using spectral estimation (LPC) if the entire previous 20 ms window of the signal is voiced.

In order for the voicing detector to work properly for both male and female speakers, various parameters of the voicing detector need to be modified. The main purpose of the gender detector is to determine the gender of the speaker and pass this information to the voicing detector so that it is able to modify its parameters. The gender detector uses a pitch based method to classify the gender of the speaker where the pitch is calculated using an autocorrelation based method [2]. The gender detector also provides the pitch estimate to the first formant filter so that an additional zero can be added at the location of the pitch in the AZF of the first formant filter.

Extensive testing of the algorithm has shown that these modifications have made the formant tracker accurate and robust to a wide variety of real-life background noise conditions. The algorithm is able to provide reliable formant frequency estimates from continuous speech for both male and female speakers. It recovers quickly and with minimal error when problems do occur and when there is a switch in speakers. The formant estimates it provides are smooth and can be obtained in real-time for use in a CEFS amplification scheme.

1.5. Thesis Layout

Chapter 2 of this thesis describes some of the material that is considered to be important background information for understanding the discussion and analysis presented in later sections. It includes a brief discussion about the anatomy of speech production followed by a detailed discussion on what formant frequencies are and their importance in speech perception. Also included in chapter 2 is a detailed look at the formant frequency characteristics in different types of sounds and a detailed look at the motivation behind the proposed algorithm. Chapter 3 contains a detailed discussion and analysis of the formant frequency algorithm being proposed including implementation details. The proposed algorithm has also been tested vigorously under various real-life scenarios, and chapter 4 describes the details of the tests cases for which the algorithm was tested as well as their results. Finally, in chapter 5, details of the three traditional formant frequency estimation techniques that have been implemented are presented along with the results and a discussion of their limitations. The thesis concludes in chapter 6 with some closing remarks about the performance of the proposed algorithm as well highlights of the limitation of the traditional formant frequency estimation techniques.

1.6. Related Publications

Parts of this thesis have appeared in the following publication:

> Mustafa, K. and Bruce, I. C. "*Robust formant tracking for continuous speech in speaker variability*". Proceedings of the International Symposium on Signal Processing and its Applications (ISSPA) - 2003, Vol. 2, pp. 623 -624.

A journal article is currently being prepared for submission to the IEEE Transactions on Speech and Audio Processing.

2. BACKGROUND AND MOTIVATION

This chapter contains background information that is required to fully understand the topics and discussions that follow in later sections of this thesis. Most of the information presented here is elementary and is limited to the topics deemed critical to understanding later discussions. It is recommended that for further details on these topics, the reader should refer to the references mentioned throughout the thesis.

2.1. Anatomy of Speech Production

Figure 2-1 shows a simplified view of the different parts of the human body that make up the speech production system. The lungs provide the airflow needed for speech production to the larynx. The larynx modulates the continuous airflow from the lungs into either a periodic or noise-like airflow and then passes it into the vocal tract. The vocal tract is made up of the oral, nasal and pharyngeal cavities and provides spectral shaping to the modulated airflow (periodic or noisy) from the larynx. Sound sources can also be produced within the vocal tract itself by constrictions and relaxations creating an impulsive airflow. Following the spectral provided by the vocal tract to all three sound sources, the lips vary the air pressure of the airflow resulting in traveling sound waves that are perceived as speech.

This description provides an idealized model of the anatomy of speech production however, in reality the sound sources required to produce most sounds are not ideal (periodic, noisy or impulsive) but usually a mixture of these types and change with the environment.

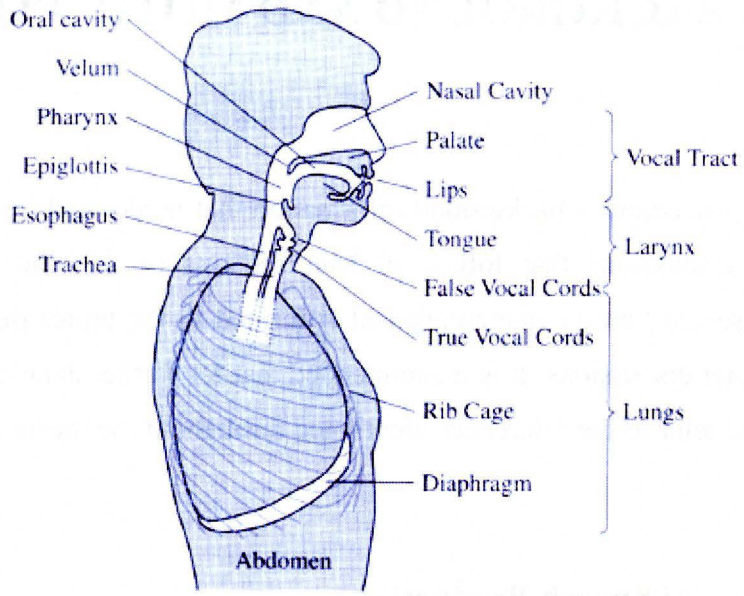

Figure 2 - 1 – Cross-sectional view of the anatomy of speech production (Reprinted from Quatieri [2])

2.1.1. The Lungs

The lungs normally inhale and exhale air in a rhythmic manner for respiration. However, during speaking the lungs override this rhythmic pattern of inhalation and exhalation of air to exhale air more slowly. Usually the period of exhalation roughly coincides with the length of the sentence being spoken. A steady and slow contraction of the rib cage provides a timed exhalation from the lungs during which the air pressure in the lungs is maintained to be roughly constant. This allows the lungs to provide a steady airflow to the larynx for the entire duration of the sentence being spoken. Note that even though the sound source provides a steady airflow to the larynx, the properties of the larynx and the vocal tract allow the pressure of the airflow being produced to vary.

2.1.2. The Larynx

The main purpose of the larynx in the speech production system is to control the *vocal folds*. The vocal folds are a mass of flesh, ligament and muscle that stretch between the back and the front of the larynx. The *glottis* is a slit-like opening between the two vocal folds and the size of this slit can be varied. The tension in the vocal folds can also be varied by the muscle and cartilage around them. The vocal folds (along with the epiglottis) close during eating to prevent food from entering the larynx. The vocal folds have three main states: breathing, voiced and unvoiced. During the breathing state the vocals folds are wide open and the muscles within them are relaxed to allow the air from the lungs to flow through freely. During speaking (for both voiced and unvoiced states) an obstruction to the airflow is provided by the vocal folds.

In the voicing state the vocal folds tense up and are brought close together partially closing the glottis leading to self-sustained oscillations as air passes through the glottis. Figure 2-2 shows the periodic airflow that is observed in the glottis during voicing. The contraction of the lungs results in air flowing through the glottis. As the airflow velocity increases the local pressure in the glottis decreases and the tension in the vocal folds increases. These two factors lead to an abrupt closing of the glottis. This is followed by an air pressure build-up behind the vocal folds causing them to open slowly and allowing air to flow through. The process is then repeated again resulting in the periodic release of puffs of air into the vocal tract. As shown in Figure 2-2, the airflow begins slowly (with the vocal folds opening slowly from their closed position) and builds to a maximum and then abruptly drops to zero as the vocal cords close quickly. The time period during which the vocal folds are closed is called the 'closed phase'. The time period during which there is some airflow before the maxima is reached is called the 'open phase'. The time between the airflow maxima and the total closure of the vocal folds is called the 'return phase'. The time duration for one such complete cycle is called the *pitch period* and its reciprocal is called the pitch frequency or just the *pitch* (or *fundamental*

frequency). The pitch ranges from about 60 Hz to 400 Hz for most phonemes depending on various factors including the gender of the speaker. Adult males typically have lower pitch than adult females because their vocal cords are longer and larger.

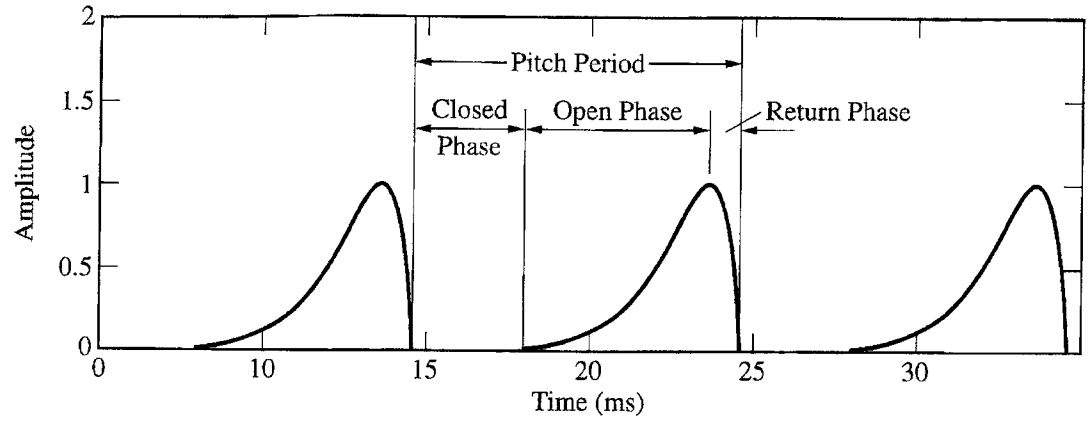

Figure 2 - 2 – Periodic Glottal Airflow (Reprinted from Quatieri [2])

The Fourier transform of the periodic glottal waveform is characterized by harmonics and the spectral envelope of the harmonics has an approximately -12 dB/octave roll off. The exact value of the roll off depends on the speaker and can change slightly. The lower frequencies of the speech spectrum contain more energy than the upper frequencies. This is because the lower frequencies contain the glottal pulsing and radiation from the lips. This causes the speech to have a *spectral tilt*, which needs to be compensated prior to speech processing, to allow equalisation of the energy distribution in the speech spectrum and obtain better spectral estimation in the higher frequency regions.

During the unvoiced state, the shape of the vocal folds is similar to that in the breathing state and there is no vocal fold vibration. However, the vocal folds are closer together during un-voicing than in the breathing state and this leads to some amount of turbulence being caused at the folds, as the air passes through. This turbulence is called *aspiration* and sounds produced through aspiration are sometimes called 'whispered'

sounds because this turbulence is also created during whispering (without any oscillations of the vocal folds). Aspiration can also occur with voicing leading to a 'breathy' voice. The vocal folds can also move in a form that does not fall clearly in any of the three states defined above. These includes a 'creaky' voiced state where the vocal folds are tense but only a short portion of the vocal folds are actually in oscillation resulting in a harsh sounding voice with a very high and irregular pitch [2].

2.1.3. The Vocal Tract

The vocal tract is made up of the oral cavity from the larynx to the lips and the nasal passage coupled with the oral passage (through the vellum). The oral tract can take on various different configurations depending upon the shape and movement of the tongue, mouth, teeth, lips, jaw, etc. The vocal tract provides frequency shaping to the output from the larynx and also generates new sources for sound production (impulsive source).

Under certain circumstances, the vocal tract can be modeled as a linear filter with resonances. The resonance frequencies of the vocal tract are called *formant frequencies* or just *formants*. The formant frequencies change with different vocal tract configurations. The peaks of the vocal tract response correspond roughly to its formant frequencies. If the vocal tract is modeled as a time-invariant, all-pole linear system, then each of the conjugate pole pairs corresponds to a formant frequency (resonance frequency). Generally, as the length of the vocal tract increases the formant frequencies decrease, so the formant frequencies of adult males are somewhat lower than those of adult females, for the same sound.

When using the time-invariant, all-pole linear system model of the vocal tract, the speech waveform can be obtained through the convolution of the glottal flow with the vocal tract impulse response. It is important to discriminate between the formant

frequency and the *harmonic frequency*. Formant frequencies correspond to the vocal tract frequency response poles while harmonic frequencies arise from the periodicity of the glottal source. When the vocal tract vellum is lowered, the nasal passage is introduced into the vocal tract and the oral tract closes resulting in the acoustic waves propagating though the nasal cavity, this produces 'nasal' sound such as 'm'. These sounds are often dominated by the lower frequency formants due to the large volume of the nasal cavity. When the vellum is lowered while keeping the oral cavity open the resulting sounds are referred to as 'nasalized speech'. The effect of the nasal passage on the vocal tract is to broaden the formant bandwidths (due to greater loss of energy in the nasal passage) and to introduce anti-resonances (zeros) into the vocal tract system model due to coupled resonances.

It should be noted that the time-invariant vocal tract model can only be applied when the vocal tract configuration is steady and constant. As mentioned earlier, the vocal tract changes its shape with time so the time-invariant model can only be applied over short time periods or for sounds with a long, repetitive duration, such as sustained vowels, with a temporal windowing heuristic.

2.1.4. Categorization of Speech Sounds by Source – Voiced and Unvoiced Speech

The broadest way to categorize sounds is by the source to the vocal tract that produces the sound. As described earlier there are three main sound sources: periodic, noisy and impulsive. In a broad sense, sounds produced due to a periodic glottal source are called *voiced* sounds, and sounds produced otherwise are called *unvoiced* sounds. Generally, voiced speech has more low-frequency energy and is quasi-periodic (such as steady state vowels) requiring vibration of the vocal cords. On the other hand, unvoiced speech has more high-frequency energy, is noisy in nature and does not require the vibration of the vocal cords. Figure 2-3 shows an example of a typical waveform for a

voiced speech segment displaying the lower frequency and quasi-periodic characteristics of voiced speech sounds. The waveform is for the vowel sound /a/ (as in 'father').

There are a variety of unvoiced sounds. Those that are created due to a noise source at an oral constriction are called *fricatives* because the noise is created by friction of the air moving against the constriction. The sound of 'th' in the word 'thin' is a fricative with the friction being provided between the tongue and the upper teeth. The waveform for 'th' is shown in Figure 2-4 and shows the typical higher frequency and noise-like characteristics of unvoiced speech. Another unvoiced sound class is the *plosives* (such as 't' and 'p') created with an impulsive airflow from the vocal tract as the sound source. When the barrier to the airflow is provided by partially closed vocal folds a new class of unvoiced sounds is produced called *whispers* (such as 'h'). Sometimes the sound source is from a combination of voiced and unvoiced sources such as in the case of the sound 'z' where there is friction as well as simultaneous voicing; this class of sounds are therefore called *voiced fricatives*. Similar in concept are *voiced plosives* which occur due to simultaneous impulsive and voiced sources as in the sound 'b'.

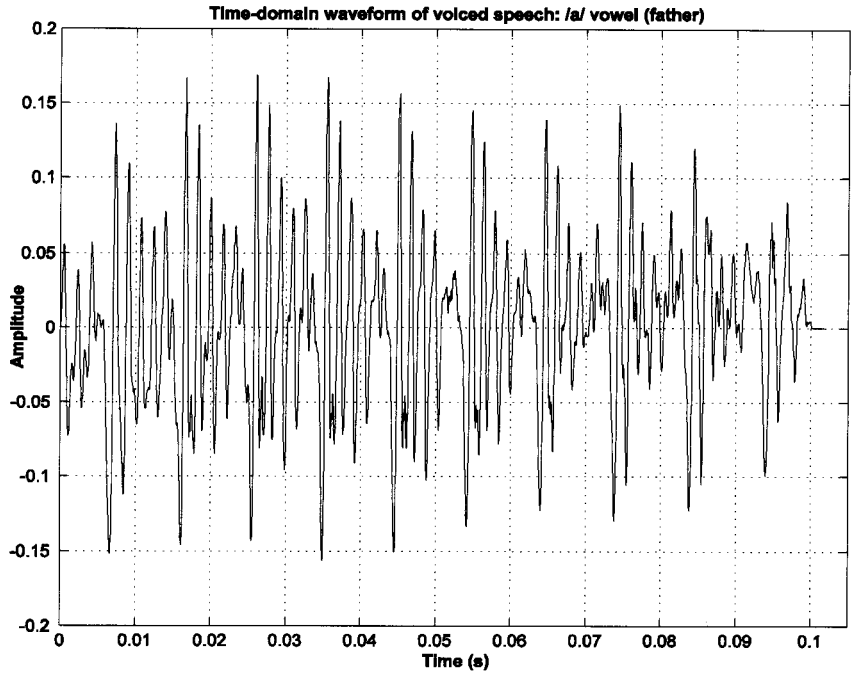

Figure 2 - 3 – Waveform of a Voiced Speech segment (for /a/ as in 'father')

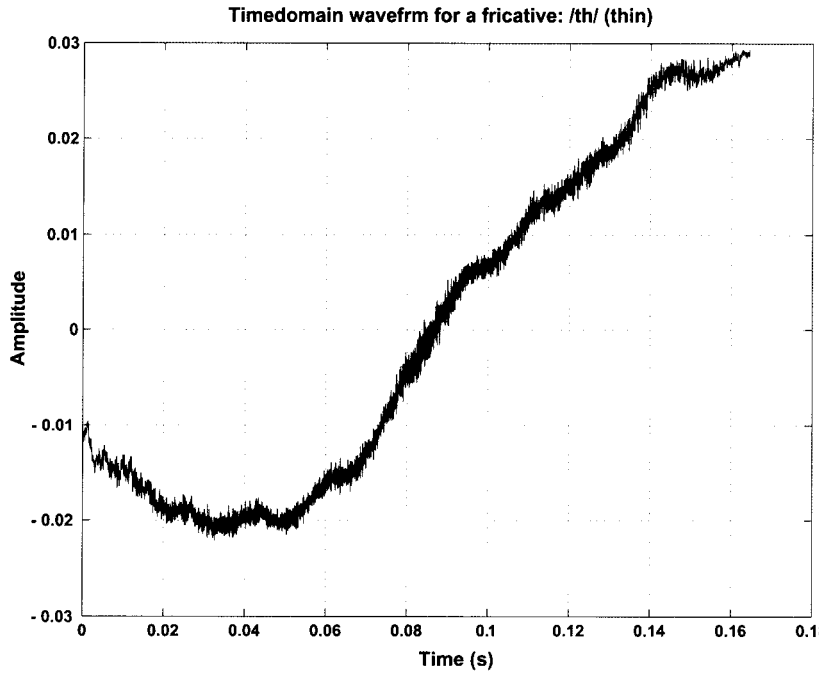

Figure 2 - 4 – Waveform of a fricative sound (for /th/ as in 'thin')

2.2. Formant Frequencies

As mentioned earlier, the resonance frequencies of the vocal tract are called the *formant frequencies* or *formants*. In this section, the origins and characteristics of formants are explored further in terms of their behaviour in different types of sounds and the problems that are caused in extracting the formants from these sounds.

2.2.1. Vocal Tract Filtering and Formant Frequencies

Figure 2-5 shows a complete discrete-time speech production model for periodic, noisy and plosive speech. G(z) is the Z-transform of the glottal flow input, $R_g(z)$ is the radiation impedance modeled by a single zero and V(z) is the stable all-pole vocal tract transfer function. A_v, A_n, and A_i are the gains that controls the loudness of the sound for periodic, noisy and plosive sources respectively. $R_l(z)$, in H(z), models the radiation impedance of the lips.

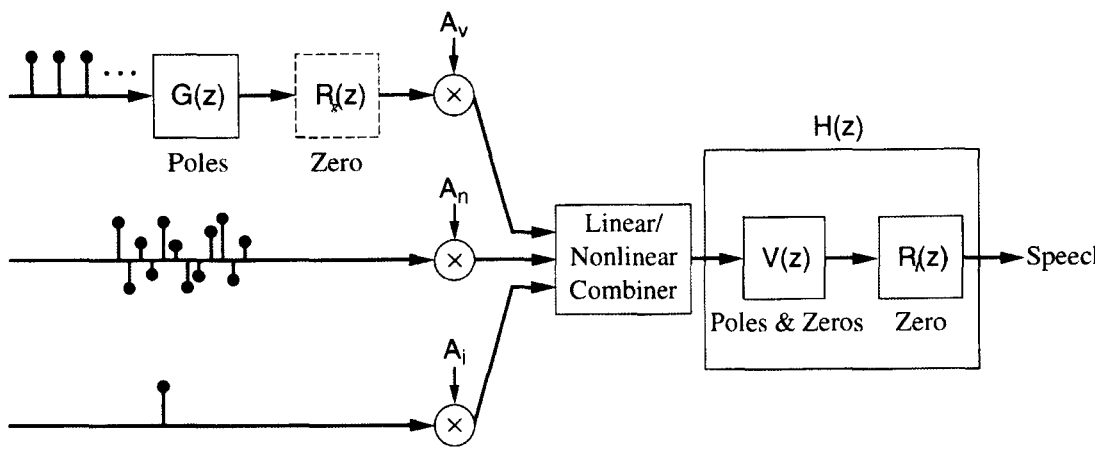

Figure 4.20 Overview of the complete discrete-time speech production model.

Figure 2 - 5 – Discrete-Time Model of Speech Production (Reprinted from Quatieri [2])

The vocal tract transfer function V(z) varies with the type of sound produced and also depends on the speakers and their speaking style. The formant frequencies vary with different vocal tract configurations and therefore, formant frequencies vary in speech with time as the vocal tract changes its shape. The peaks of the vocal tract response in each configuration correspond roughly to its formant frequencies. The first resonance of the vocal tract is called the *first formant frequency* (or F1), the second resonance of the vocal tract is called the *second formant frequency* (or F2), and so on. For perfectly voiced and periodic speech (as in sustained vowels) the vocal tract can be accurately modeled by the stable all-pole model for V(z). However, in order to model other types of sounds, zeros are also added to V(z) in order to model the nasal cavity of the vocal tract. The resonances or peaks of the vocal tract transfer function (poles of the V(z) transfer function) correspond roughly to the formant frequencies of a particular sound. The characteristics and behaviour of formant frequencies change in different types of sound and estimating formants in continuous speech is a challenging task.

2.2.2. Phonemic Classification of Speech and Formant Behaviour in Phonemes

In this section, the behaviour and characteristics of formant frequencies in different types of sounds are explored in greater detail to understand the problems associated with estimating formant frequencies in such cases. In general, formant frequency regions have more energy than the other frequencies in the speech spectrum and can easily be visually identified in spectrograms.

Phonemes are the fundamental distinctive units of sound. Each distinct and identifiable sound in a language forms a phoneme. Figure 2-6 shows all the different phonemes in American English grouped together by their phoneme class. Formant frequencies for each phoneme vary and will also depend on the speakers and their individual speaking style. However, formant frequencies for a particular phoneme class

(for a particular speaker) have similar characteristics and behaviour. The formant frequency behaviour and characteristics of some of these phoneme classes are discussed below.

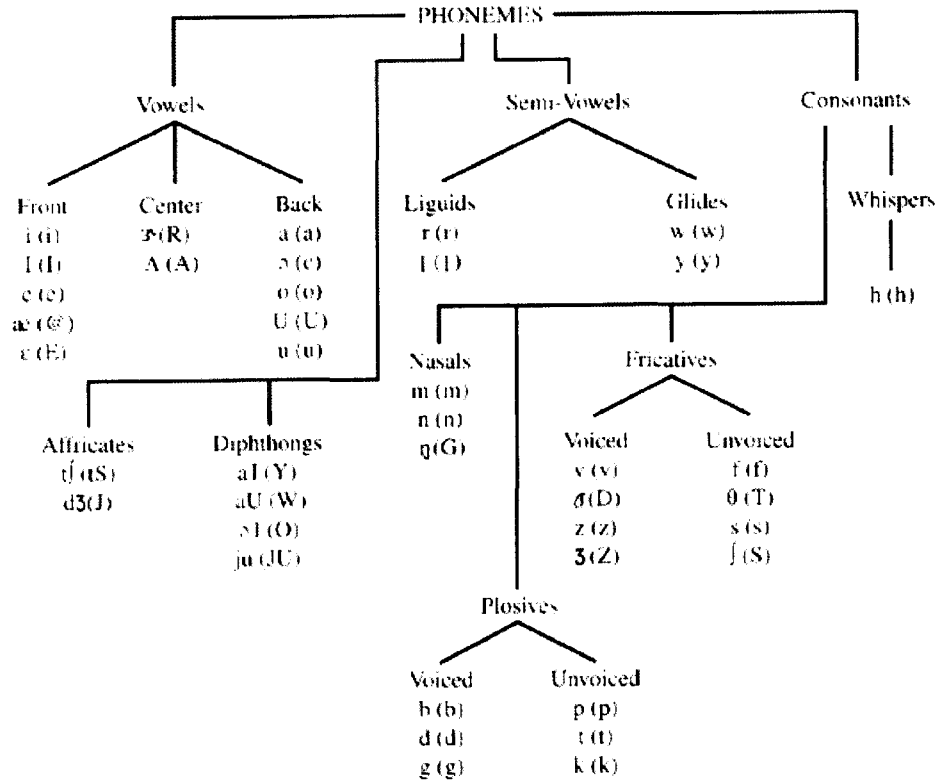

Figure 2 - 6 – Classification of Phonemes in English (Reprinted from Quatieri [2])

Vowels

Vowels make up the largest class of phonemes. The three types of vowels are grouped according to the tongue hump position required to make the sound (front, central or back). Vowel sounds are produced by quasi-periodic airflow through the glottis that vibrates the vocal folds at a certain fundamental frequency. The nasal tract remains closed in vowel sound production so the vocal tract does not contain the effects of the nasal cavity. The lips can contribute to the vocal tract configuration through their degree of opening and rounding. The position of the tongue (front, centre or back) determines

the phoneme produced, e.g. /a/ ('father') and /i/ ('eve') are differentiated primarily through the position of the tongue hump. Figures 2-7 and 2-8 show the waveform and spectrogram of the vowel sound /i/ ('eve'). The quasi-periodic nature of the vowel can be seen from its waveform and from the spectrogram it can be observed that the formant frequency regions have concentrated energy, these features are common to all vowels. The strong energy of the formant regions and the periodic nature of the waveform make it relatively easy to extract formant frequencies of pure and sustained vowel-like sounds. Despite general similarities between different vowel sounds it is important to remember that the exact formant frequencies of different vowels differ from each other and depend on a wide variety of parameters including the speakers and their speaking style.

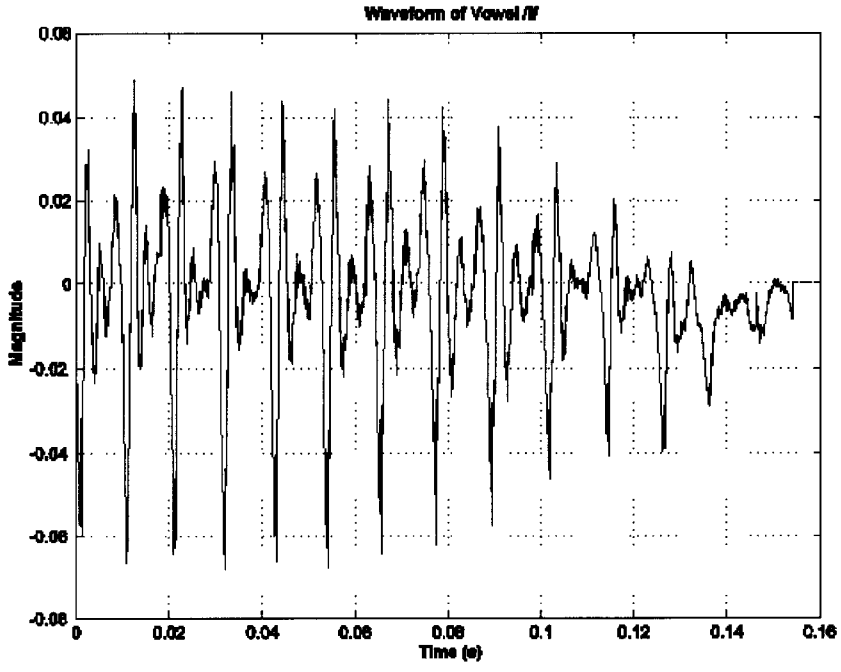

Figure 2 - 7 – Waveform of vowel /i/ ('eve')

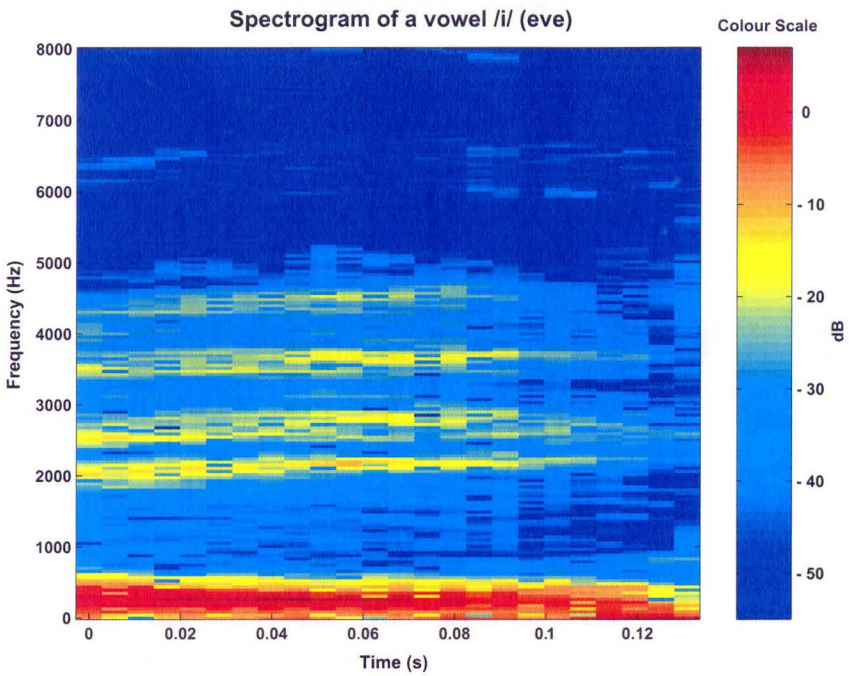

Figure 2 - 8 – Spectrogram of vowel /i/ ('eve')

Fricatives

There are two main types of *fricative* consonants: unvoiced and voiced. Unvoiced fricatives are generated through turbulence in the airflow being provided at some point in the oral tract without any vocal fold vibration e.g. /f/ ('father'). The constrictions provided through the hump in the tongue, lips, teeth, etc., help separate the rear and front oral cavity regions. The primary source of spectral shaping is the front of the oral cavity however, anti-resonances that are provided by the rear of the oral cavity also have an effect on the overall spectral shaping provided by the vocal tract. The transfer function of the vocal tract is made up of primarily higher frequency resonances that vary with the location of the vocal tract constrictions. Figures 2-9 and 2-10 show the waveform and the spectrogram of the unvoiced fricative /f/. From these figures it can be seen that unvoiced fricatives have 'noisy' waveforms as expected.

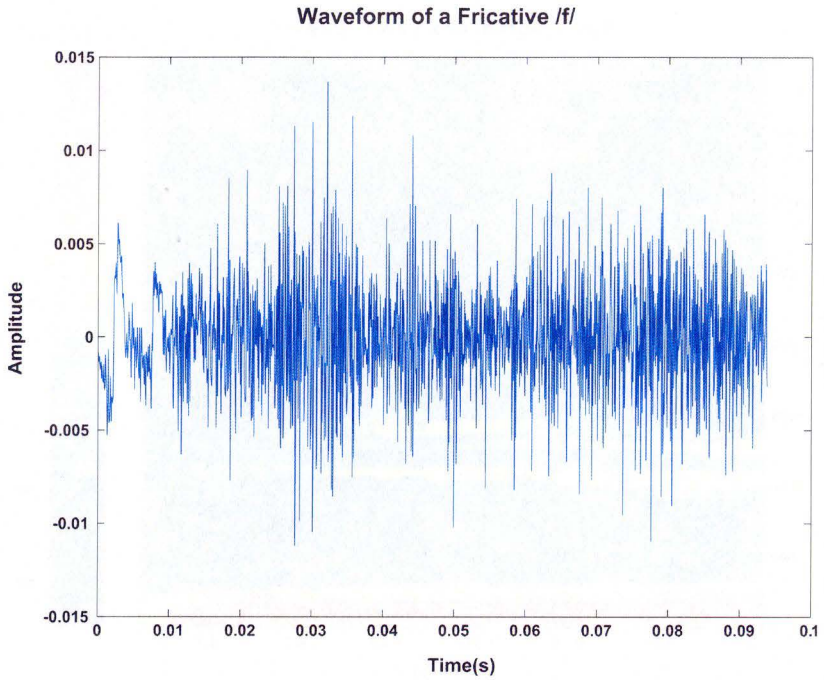

Figure 2 - 9 – Waveform of unvoiced fricative /f/ ('father')

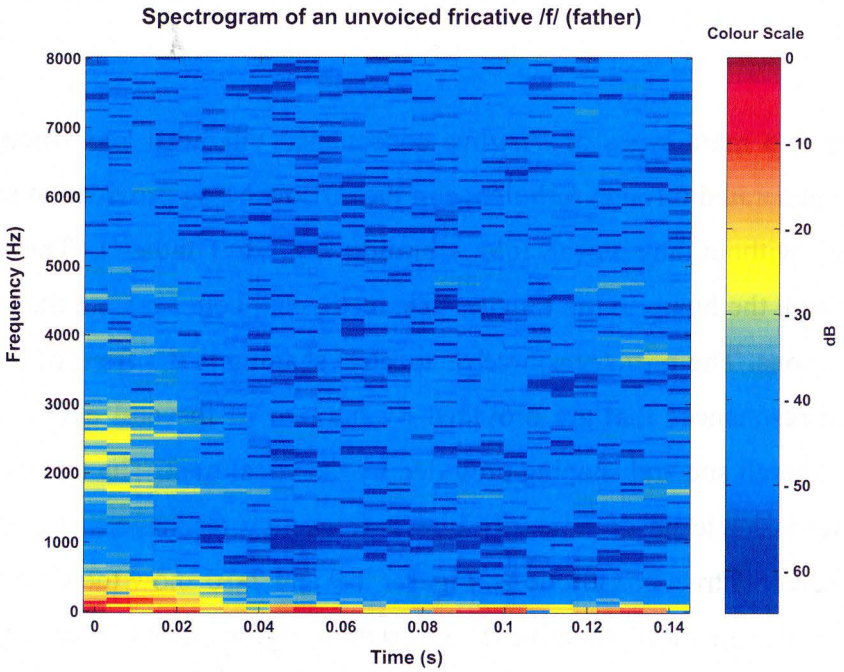

Figure 2 - 10 – Spectrogram of unvoiced fricative /f/ ('father')

Voiced frication also occurs due to turbulence in the airflow provided from within the oral tract but it is often accompanied by some vocal fold vibration as in /v/ ('vote'). The vibration of the vocal folds means that the airflow in the vocal tract is periodic and frication only takes place when the periodic airflow has reached a certain minimum level. This leads to frication being roughly synchronized with the glottal airflow velocity. Voiced fricatives can be differentiated from unvoiced fricatives through the onset of voicing. The formant transitions from fricatives to vowels also serve as a cue to distinguish between voiced and unvoiced fricatives. In voiced fricatives the voicing occurs sooner in the transitions than for unvoiced fricatives. Figures 2-11 and 2-12 show the waveform and spectrogram for the voiced fricative /v/. The figures show that in voiced fricatives the noise in the waveform is super-imposed on a quasi-periodic envelope. The spectrogram shows characteristics of both noisy and periodic signals.

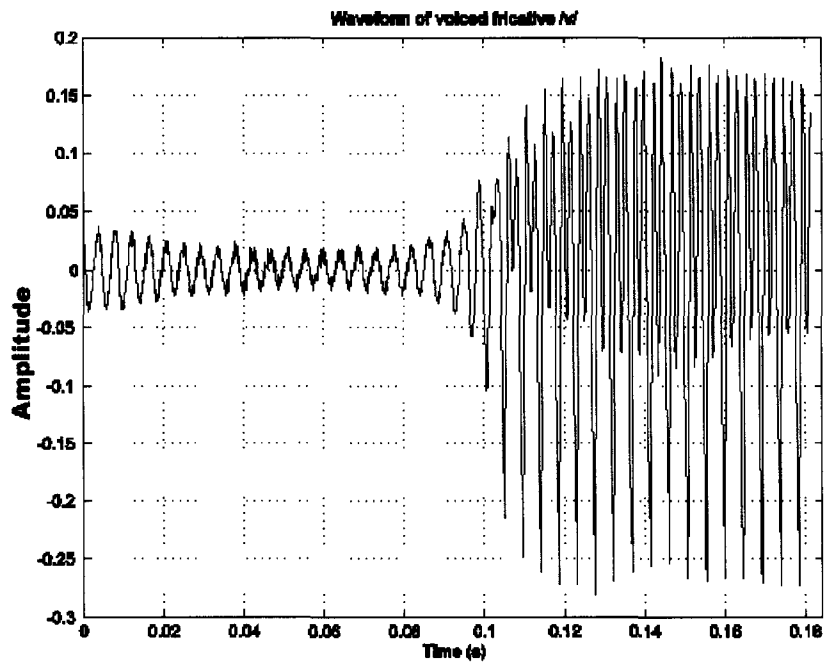

Figure 2 - 11 – Waveform of voiced fricative /v/ ('vote')

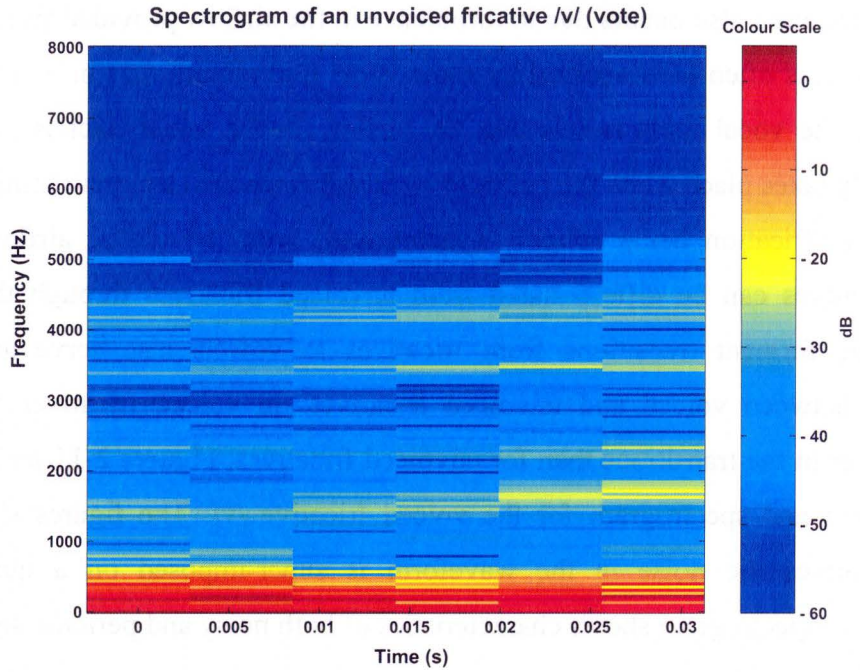

Figure 2 - 12– Spectrogram of voiced fricative /v/ ('vote')

Nasals

Nasal consonants are produced from a source similar to that used for producing vowels – semi-periodic airflow through the vocal tract that vibrates the vocal folds. For nasals, the vellum is lowered and air is mainly radiated through the nostrils because the oral cavity is constricted. Due to the large volume and low resonance of the nasal cavity, nasals are dominated by lower frequency energy with the first formant frequency usually being the most prominent in the spectrogram. The formant transitions that follow the release of the constriction into the steady state vowel position are used to perceptually differentiate between the different nasal consonants. Figures 2-13 and 2-14 show a waveform and spectrogram of the nasal consonant /m/ ('me'). It can be seen from the spectrogram that the nasal sound is dominated by lower frequency energy and the first formant has the highest energy.

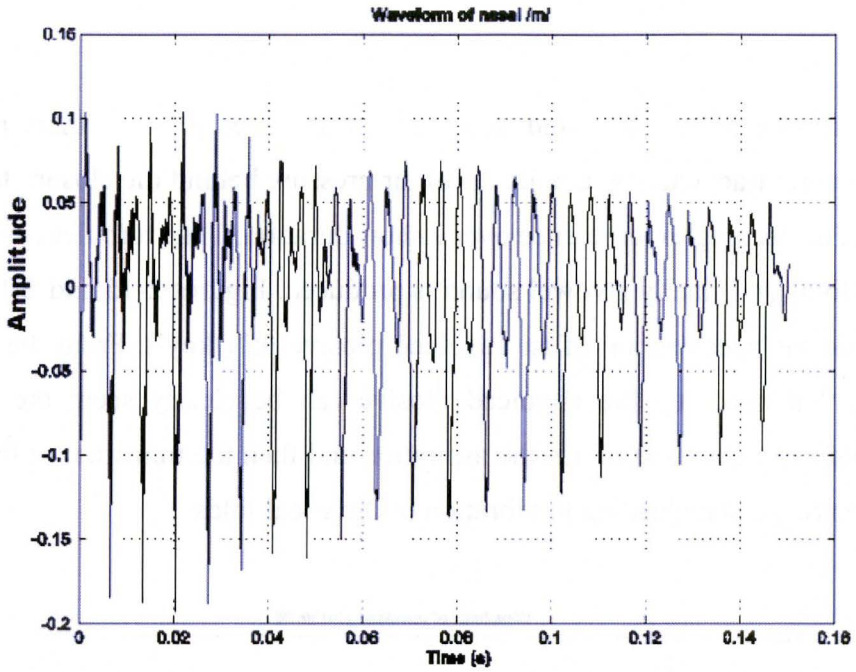

Figure 2 - 13 – Waveform of Nasal /m/ ('more')

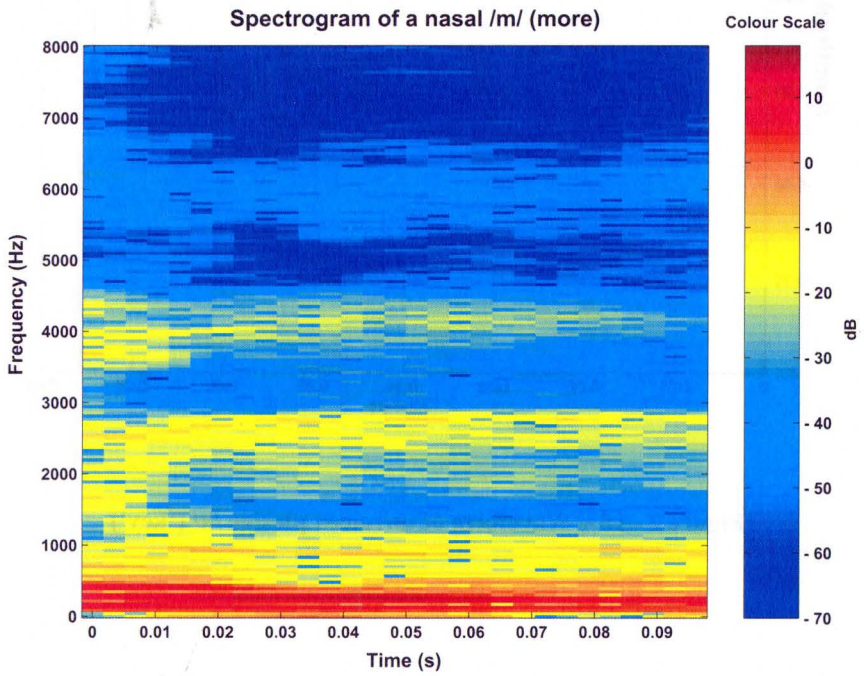

Figure 2 - 14 – Spectrogram of Nasal /m/ ('more')

Plosives

Plosives can be both voiced and unvoiced. In unvoiced plosives there is complete closure of the oral tract causing a build-up of air pressure behind the closure followed by the release of air leading to turbulence over a short duration. Then turbulence is generated at the vocal folds and finally a vowel sound is produced. Figures 2-15 and 2-16 show the waveform and the spectrogram of an unvoiced plosive /k/ ('key'). From the figures the main stages that make up the unvoiced plosive can be clearly seen: the silence (as pressure builds up), the burst of air, the aspiration and then the transition of the oral tract from the constricted state leading to vibration of the vocal folds.

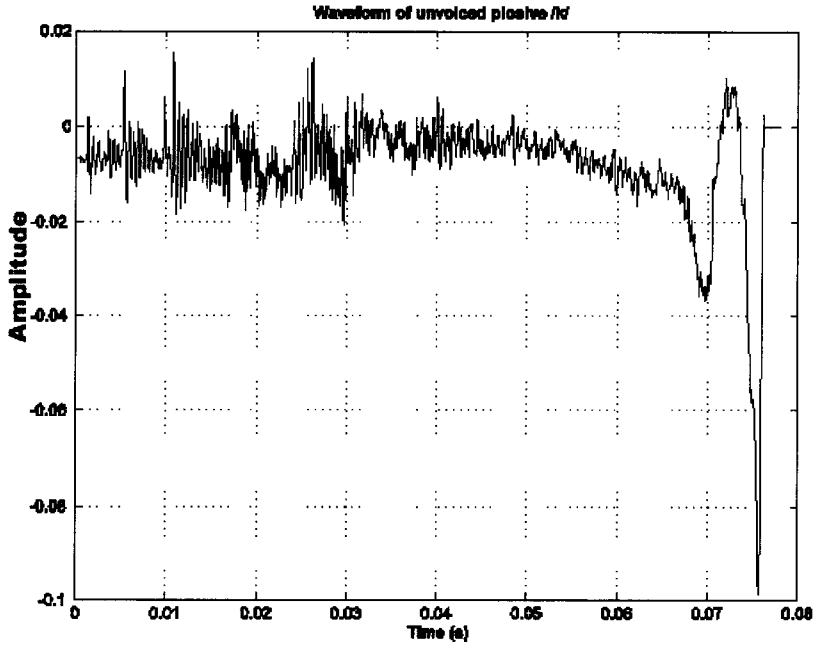

Figure 2 - 15 – Waveform of unvoiced plosive /k/ ('key')

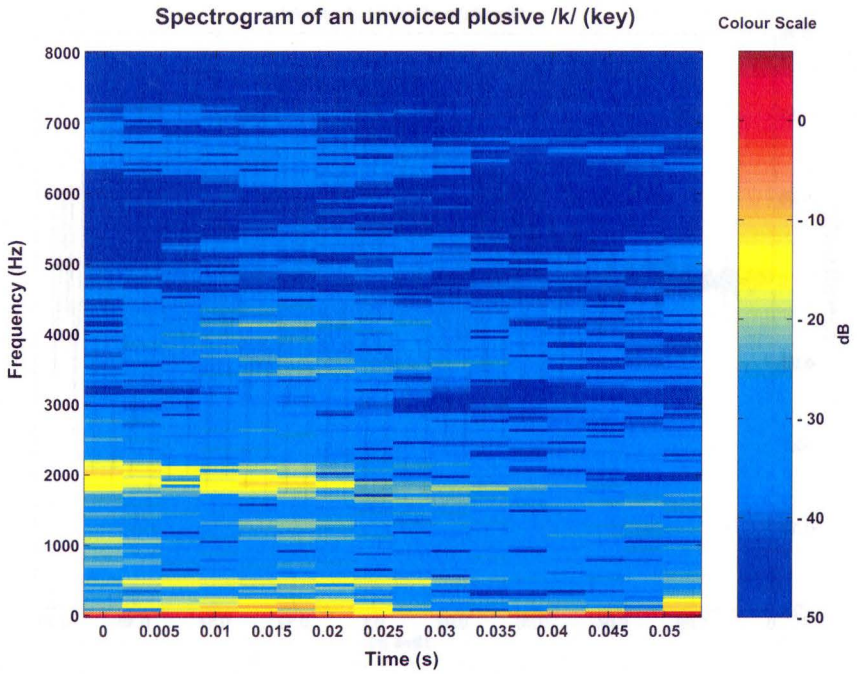

Figure 2 - 16 – Spectrogram of unvoiced plosive /k/ ('key')

Voiced plosives are generated by a mechanism similar to that of unvoiced plosives. However, in voiced plosives there is vocal fold vibration during the pressure build-up stage. This is called the voice bar and it is generated due to the low-frequency vibration of the walls of the throat. Also, after the release of air there is no aspiration and the start of the transition to the vowel occurs much faster than in unvoiced plosives. Figures 2-17 and 2-18 show the waveform and spectrogram of the voiced plosive /g/ ('go'). The low frequency voice bar can be seen in the spectrogram and the waveform. Most of the energy in both voiced and unvoiced plosives is lower frequency and so the first formant is very strong compared to the other formants.

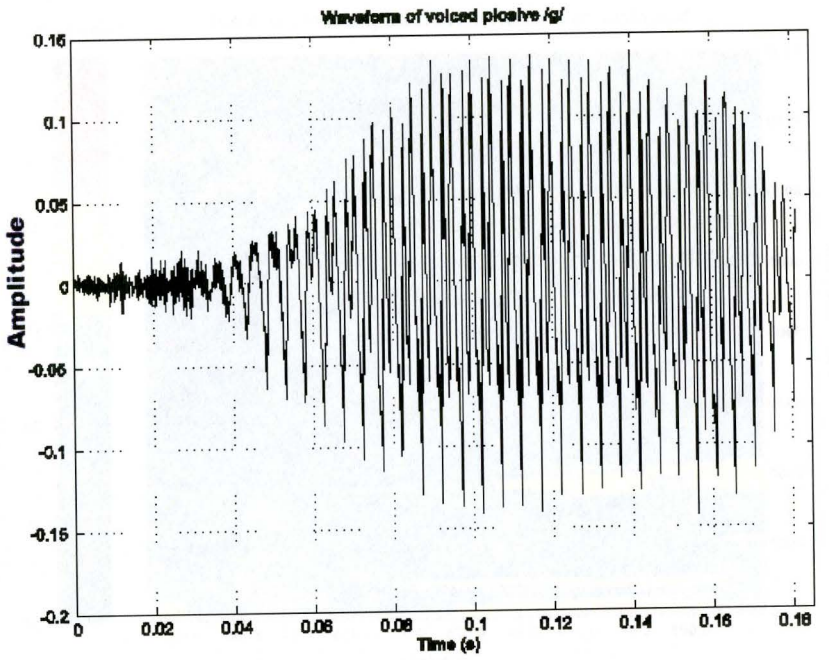

Figure 2 - 17 – Waveform of voiced plosive /g/ ('go')

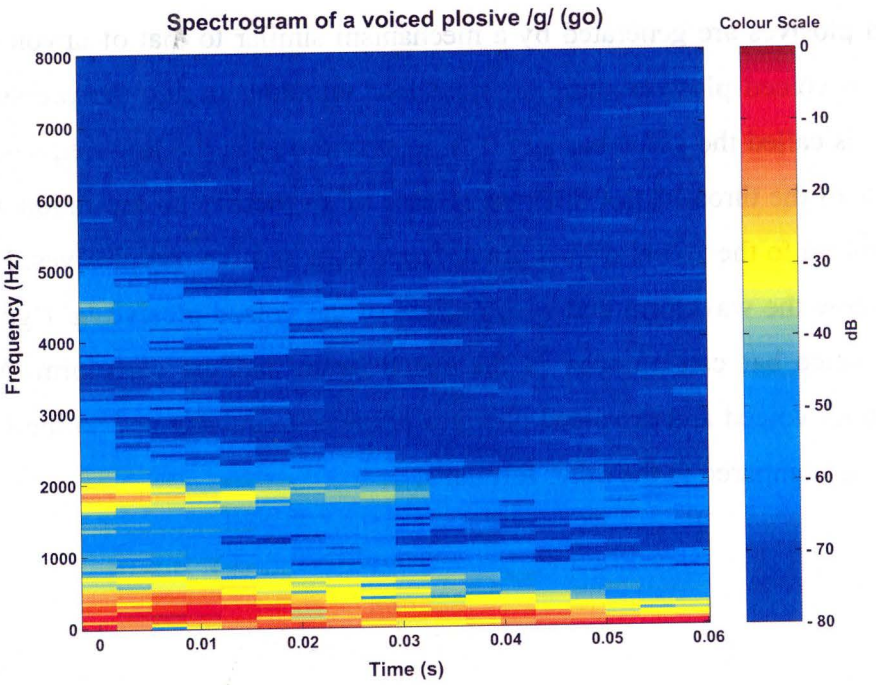

Figure 2 - 18 – Spectrogram of voiced plosive /g/ ('go')

Transitional Speech Sounds

The types of phonemes described earlier have mostly stationary articulators and the resulting formants are fairly constant throughout the duration of the phoneme. However, during speech there are fast transitions from one phoneme to the next so it is important to also consider the effects of these transitions. Transitional articulators occurring during speech are non-stationary. Many sounds are actually defined through their transition stage rather than their stationary stage.

Diphthongs are phonemes that are characterized through their transitional nature. These phonemes are similar to vowels in that they are produced through vibration of the vocal cords. However, unlike vowels diphthongs are not generated through a steady vocal tract configuration but rather through constant transition of the vocal tract between two vowel-like configurations. Figures 2-19 and 2-20 show the waveform and spectrogram of the diphthong /O/ ('boy'). From the spectrogram, it can be seen that the rapid change in the vocal tract configuration is characterized by fast changing formant frequencies, especially F2.

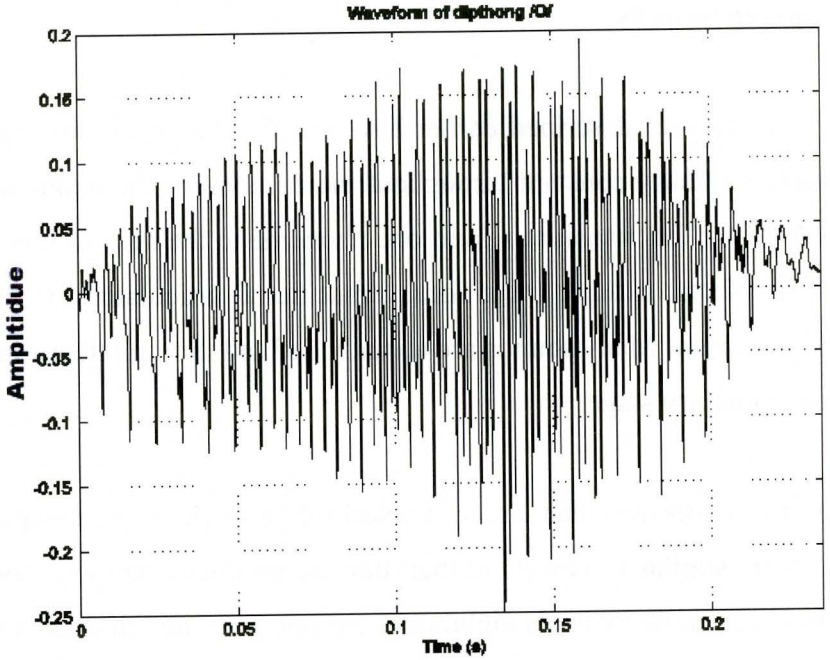

Figure 2 - 19 – Waveform of diphthong /O/ ('boy')

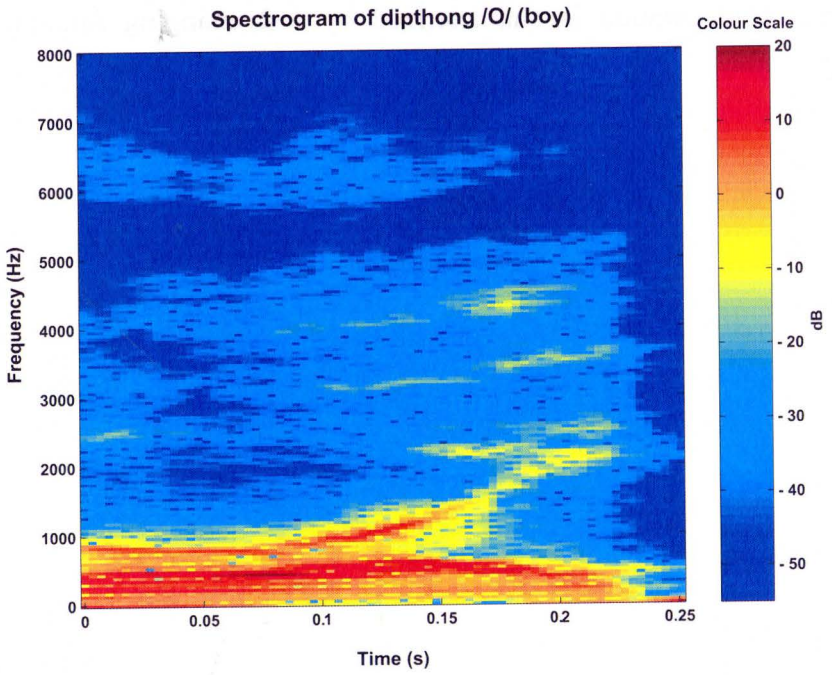

Figure 2 - 20 – Spectrogram of diphthong /O/ ('boy')

Semi-vowels are another phoneme class that are similar to vowels and are characterized through their transitional phases rather than their steady state configuration. There are two kinds of semi-vowels; glides (/w/ as in 'we') and liquids (/r/ as in 'read'). They both have fast transitional configurations and fast moving formant frequencies.

Figure 2-21 shows the variation of the second formant frequency versus the first formant frequency for 10 vowels spoken by 76 different speakers. Each dot on the figure represents the formant frequencies of a single speaker. From the figure it is clear that there are slight differences in the exact formant frequencies of individual speakers even for the same vowel. However, the formant frequencies for a given vowel are usually similar even for different speakers and can be grouped together (shown by circles surrounding a group of dots). This figure shows that despite the fact that the exact formant frequencies for a given vowel depend on the individual speaker, they (the formant frequencies) are often similar across different speakers and even genders.

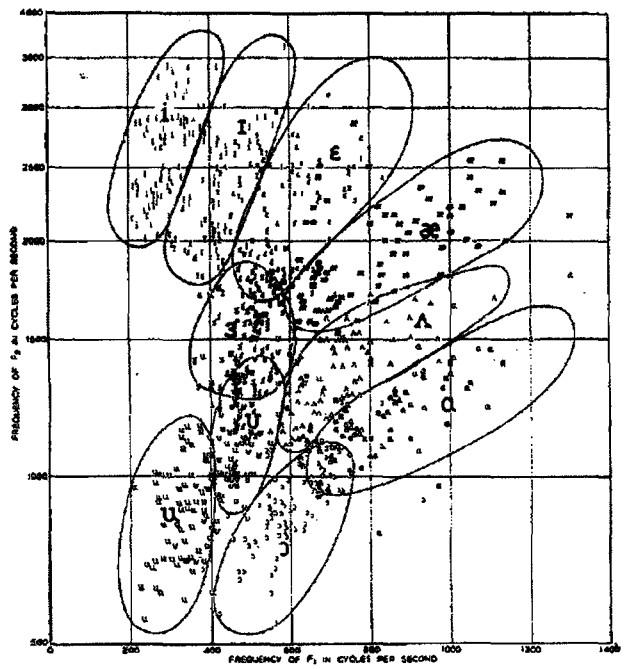

Figure 2 - 21 – Variation of the second formant versus the first formant for vowels of 76 speakers

(Reprinted from Peterson et al. [22])

2.2.3. Importance of Formant Frequencies in Speech Perception

In this section the roles of different features of speech that help differentiate human perception between the phonemes will be discussed. In vowels the primary source of discrimination between the phonemes can be provided by the formant frequencies. It has been shown that F1 and F2 are highly discriminable features for vowel identification and the higher formants also play a smaller role [8]. It is thought that the formant spacing in vowels and vowel-like phonemes are an essential feature for proper identification of vowels [2]. Nasalization is another important feature for vowel identification; it can be checked by observing the increase in bandwidth of F1. Identification of consonants is a more complex problem than identification of vowels. Among the cues used for consonant identification are formant frequency values, formant frequency transitions into the vowel following the consonant, voicing during the consonant production, and the timing of the onset of the vowel following the consonant.

It is clear from the above discussion that formant frequencies play a major role in vowel identification and are also important for consonant identification. Speech processing systems that aim to help restore the ability of the human ear to discriminate between phonemes must provide fairly accurate formant frequency estimates, at least for the lower formant frequencies.

2.3. Motivation - Restoring Normal Auditory Nerve Representation in Ears with Sensorineural Hearing Loss

Liberman et al. [6] and Sachs et al. [1] have shown that sound induced or sensorineural hearing loss causes broadening of the neural response to the first formant frequency, leading to a reduction in speech perception. Simple hearing aid amplification schemes that apply amplification independently across different frequency bands cannot satisfactorily compensate for sound-induced hearing loss. Miller et al. [5] describe a hearing aid amplification technique that can improve the neural response to vowel sounds in sensorineural damaged auditory systems. This scheme, called *Contrast Enhanced Frequency Shaping (CEFS)* amplification, tries to reverse the effects of the sensorineural hearing loss by compensating for the frequency dependent threshold shift and tries to restore the neural representation to that of a 'normal' ear [1] [4]. CEFS tries to boost the speech signal energy in the regions where the neural thresholds have shifted to higher values. The result of proper CEFS amplification is to restore the 'normal' neural response representation of the formant frequencies to vowel like sounds [5].

Figure 2-22 shows the original and CEFS modified power spectra of a synthesized vowel (/ɛ/). The CEFS spectrum is obtained by high-pass filtering the stimulus with a cut-off frequency set about 50 Hz below the second formant frequency (F2) of the vowel. The CEFS processed vowel is amplified for all frequency components above the cut-off frequency and all frequency components below the cut-off frequency are set to have the same magnitude as the original signal. Boosting the magnitude of the signal for all frequencies above F2 increases the speech signal energy in the regions where the neural thresholds have shifted higher due to the sensorineural hearing loss [1] [5].

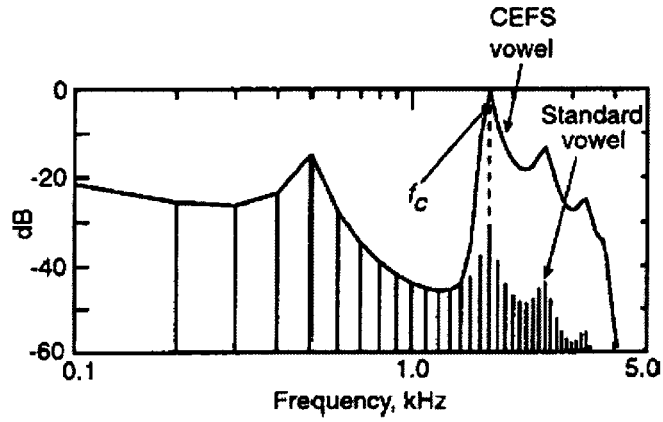

Figure 2 - 22 – Power spectra of the standard and the CEFS versions of the /ε/ vowel (Reprinted from Sachs et al. [1])

For a constant vowel like sound such as /ε/ the second formant frequency remains constant and therefore once F2 is accurately estimated the sound can be properly amplified as per the CEFS scheme. However, for complex stimuli such as continuous speech signals the different phonemes have different formant frequencies and so F2 varies with time (and indeed so do all the formant frequencies). In order to apply CEFS to continuous speech the second formant frequency has to be accurately measured in real-time. This represents a significant engineering challenge, and is the primary motivation behind the formant estimation technique developed in this thesis, although other applications for the algorithm do exist..

Accurate formant frequency estimates are used for a variety of applications other than CEFS amplification for hearing aids. Formant frequencies have been used to make natural sounding computer synthesized speech. Ding & Campbell proposed a formant frequency based distance measure in unit selection for concatenation synthesis of speech [18]. Lincoln et al. proposed using formant estimates for speaker normalisation for automatic speech recognition (ASR) systems [19]. Applications for formant frequencies have also been identified for speech coding [20] and for speaker recognition [21].

3. FORMANT TRACKING ALGORITHM

A block diagram and brief description of the Formant Tracker being proposed was presented in Figure 1-4 and Section 1.4. In this chapter the formant tracking algorithm is discussed in greater detail.

3.1. Pre-Emphasis

Voiced speech signals have a natural spectral tilt, with the lower frequencies (below 1 kHz) having greater energy than the higher frequencies. The lower frequencies have more energy because they contain the glottal waveform and the radiation load from the lips [8]. In some speech processing applications it is desirable that this spectral tilt be removed by *pre-emphasis* or spectral equalisation of the signal. A common method of pre-emphasis is to filter the speech signal using a High-Pass Filter (HPF) that attenuates the lower frequencies. The result of the pre-emphasis is the approximate removal of the contribution of the glottal waveform and the radiation load effect from the lower frequencies of the signal, i.e. the energy in the speech signal is re-distributed to be approximately equal in all frequency regions. Figure 3-1 shows the frequency response of the FIR pre-emphasis HPF filter that is used in this formant tracking algorithm. Figure 3-2 shows a spectrogram of a speech signal before and after it has been pre-emphasised using the filter from Figure 3-1. After the signal has been the pre-emphasised it is equalised to have a global RMS energy value of 0 dB. This equalisation ensures that the energy threshold levels are set properly and to appropriate energy levels (see Section 3.4).

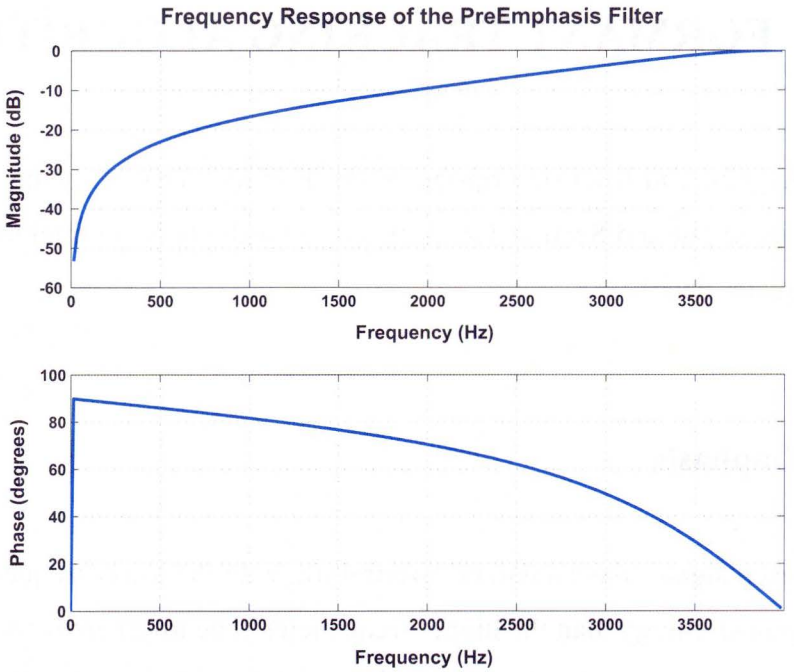

Figure 3 - 1– Frequency and phase responses of the FIR pre-emphasis high-pass filter

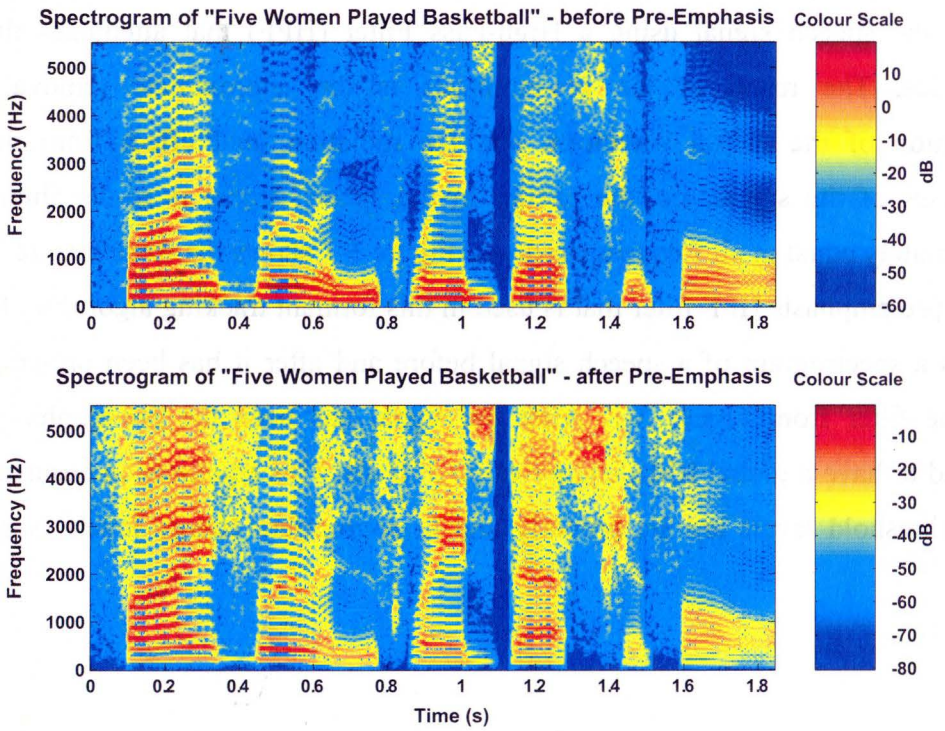

Figure 3 - 2 – Spectrogram of the speech signal before and after pre-emphasis

3.2. Hilbert Transformer

After the speech signal has been pre-emphasised, a complex version of the signal is calculated using an approximate Hilbert transformer. The primary reason behind converting the signal into its complex representation is to allow the use of complex filters in the formant filterbank (AZFs and DTFs). The formant filters are designed as complex filters because it is easier to design the unity gain and zero-lag filters in the complex domain (see Section 3.3). Converting the real-valued signal into its analytic version also decreases the amount of aliasing in the signal and thus increases the accuracy of the spectral estimation technique that is used for formant frequency estimation [11] [13].

The method of representing a real-time discrete signal as a discrete complex signal is shown in Figure 3-4. The real-time discrete signal, $S_R[n]$, can be represented by its complex form, $S_C[n]$, as $S_C[n] = S_R[n] + j\, S_H[n]$, where $S_H[n]$ is the Hilbert transform of $S_R[n]$. Although Hilbert transformers have been commonly used in various signal processing applications, ideal Hilbert transformers can not be implemented in real-time. This limitation has led to the development of various different methods that implement an approximate Hilbert transformer in real-time. The particular technique used to implement the Hilbert transformer in the formant tracking algorithm uses an optimum FIR filter [14].

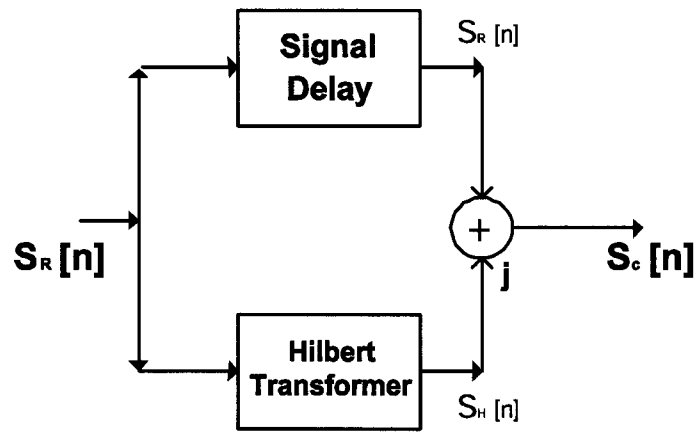

Figure 3 - 3 – Converting the real-valued signal into its analytic representation

The Hilbert transformer is implemented with a 20^{th}-order linear-phase FIR filter designed using the Parks-McClellan algorithm (Remez exchange algorithm) [15]. The frequency and phase responses of the filter are shown in Figure 3-4. The filter is designed using the Remez exchange algorithm and Chebyshev approximation to have an optimal fit between the desired and actual frequency responses. The filter is optimal in a minimax Chebyshev sense (the maximum error between the desired and actual frequency response is minimized). The real part of the signal is added back to the Hilbert transformed part after a signal delay to account for the delay in implementing the approximate Hilbert transform (10 samples in this case). The results obtained for the analytic signal using the FIR filter method were found to be approximately the same as those obtained using an ideal Hilbert transform.

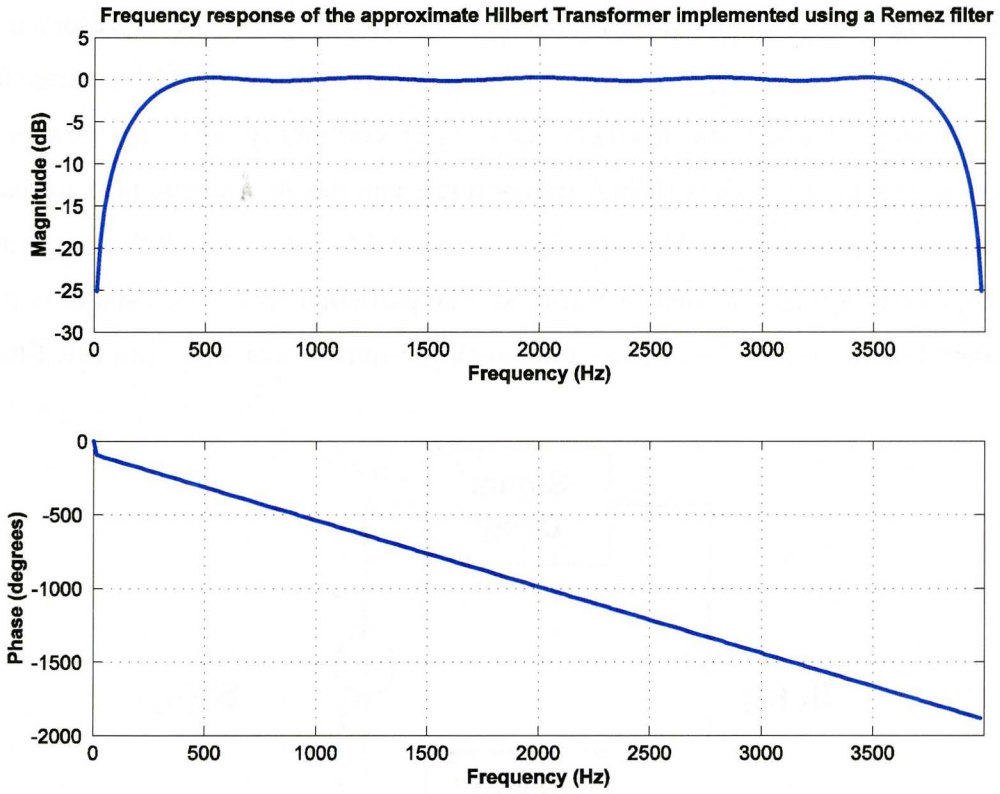

Figure 3 - 4 – Frequency response of the Hilbert transformer

3.3. The Adaptive Band-Pass Filterbank

The adaptive band-pass filterbank used in the formant tracking algorithm (shown in Figure 3-5) is similar to the one proposed by Rao and Kumaresan but it has a modified first formant filter that removes the effects of the pitch from the first formant filterband [11]. Each channel of the filter bank consists of an all-zero filter (AZF) cascaded with a single pole dynamic tracking filter (DTF). The combination of the AZF and the DTF is called a formant filter and is responsible for tracking one individual formant frequency. The filters are designed in the complex domain because it is easier to design the unity gain and zero phase lag filters in the complex domain [11]. Adaptively varying the zeros and pole of each formant filter, allows the suppression of interference from neighbouring formant frequencies and from other spectral noise sources, while tracking an individual formant frequency as it varies with time.

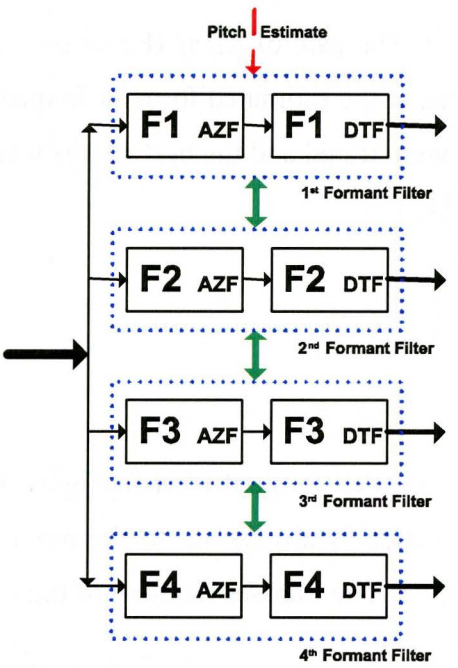

Figure 3 - 5 – Adaptive band-pass filterbank

3.3.1. AZFs

In Figure 3-5 the box labelled 'AZF' in each formant filter is the adaptive all-zero filter whose three zero locations are always set to the value of the previous formant frequency estimated from the other three formant filters. The transfer function of the k^{th} AZF at time sample index n is

$$H_{AZFk}(n,z) = K_k[n] \times \prod_{\substack{l=1 \\ l \neq k}}^{4} (1 - r_z e^{j2\pi f_l[n-1]} z^{-1}) \qquad \textbf{Equation (3.1)}$$

$$\text{where } K_k[n] = \frac{1}{\prod_{\substack{l=1 \\ l \neq k}}^{4} (1 - r_z e^{j2\pi (f_l[n-1] - f_k[n-1])})} \qquad \textbf{Equation (3.2)}$$

and r_z is the radius of the zeros on the Z-plane, $f_l[n-1]$ is the formant frequency of the l^{th} filter estimated at time index $n-1$ and, $f_k[n]$ is the formant frequency of this filter (k^{th} filter) estimated at index $n-1$. The gain of $K_k[n]$ (Equation 3.2) ensures that the AZF has unity gain and zero phase lag at the estimated formant frequency of the k^{th} component. A wide range of values for r_z were tested and the best results were obtained (for the range of values tested) for $r_z = 0.98$ [11].

3.3.2. DTFs

The box labelled 'DTF' in each formant filter in Figure 3-5 is a single-pole dynamic tracking filter. The pole location is always set to the previous estimate of the formant frequency of that formant filter. The transfer function of the k^{th} DTF at index n is

$$H_{DTFk}(n,z) = \frac{1 - r_p}{(1 - r_p e^{j2\pi f_k[n-1]} z^{-1})} \qquad \textbf{Equation (3.3)}$$

where r_p is the radius of the pole and $f_k[n-1]$ is the formant frequency of the k^{th} filter at time index $n-1$. A wide range of values for r_p were tested and the best results were obtained (for the range of values tested) using $r_p = 0.90$ [11].

3.3.3. The First Formant Filter

The transfer function of the 1st formant AZF is slightly different than that of the other AZFs. The AZF of the first formant filter has an additional zero at the location of the pitch estimate to suppress pitch effects on the first formant estimate. The transfer function of the 1st AZF at index n is

$$H_{AZF1}(n,z) = K_k[n] \times \prod_{\substack{l=0 \\ l \neq k}}^{4} (1 - r_z e^{j2\pi f_l[n-1]} z^{-1}) \qquad \textbf{Equation (3.4)}$$

$$\text{where } K[n] = \frac{1}{\prod_{\substack{l=0 \\ l \neq k}}^{4} (1 - r_z e^{j2\pi (f_l[n-1] - f_k[n-1])})} \qquad \textbf{Equation (3.5)}$$

and $f_0[n-1]$ is the pitch estimate at time index $n-1$, that is provided to the 1st formant filter by the gender detector.

After the placement of the pole and zeros for each of the formant filters, the transfer function and the complex filter coefficients of the four formant filters are calculated. These complex filter coefficients are then used to filter the analytic speech signal into four band-limited spectral regions from which the four formant frequencies are estimated.

3.3.4. The Frequency Response and the Results of the Formant Filters

The frequency responses of the four formant filters are shown in Figure 3-6. In this figure the pitch (F0) is set to 200 Hz, the first formant frequency (F1) is set to 700 Hz, the

second formant frequency (F2) is set to 1500 Hz, the third formant frequency (F3) is set to 2200 Hz and the fourth formant frequency (F4) is set to 3500 Hz. The position of the pole and the zeros of the filters is updated for each sample. The bandwidth of the formant filters is related to the values of r_z and r_p, and is kept constant since the values of r_z and r_p are not changed. All four of the filters have unity gain and zero phase lag at the location of the pole (peak of the band-pass filter that corresponds to the estimated formant frequency).

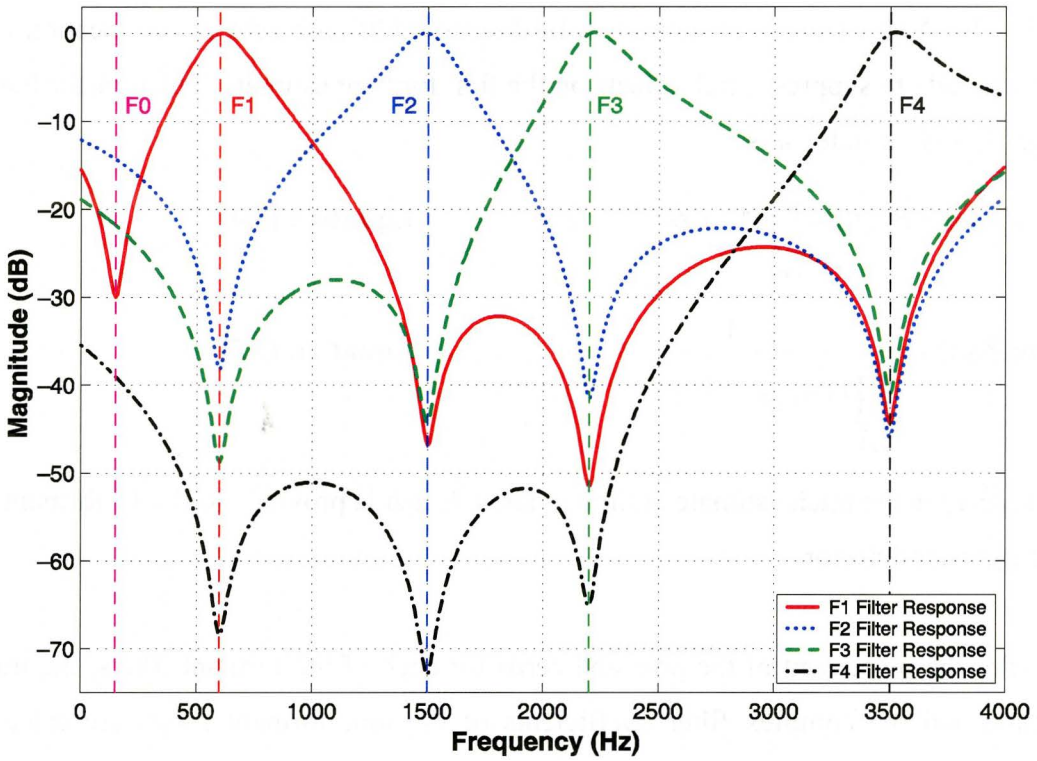

Figure 3 - 6 – Filter response of the four formant filters

Figure 3-7 shows the spectrograms of a speech signal and the spectrograms of the corresponding spectral regions that come out of the first formant filter (used for F1 estimation), second formant filter (used for F2 estimation), third formant filter (used for F3 estimation), and fourth formant filter (used for F4 estimation). As can be seen from the spectrograms, the pitch area is effectively filtered out and the higher formant

frequencies are greatly attenuated for the F1 region. The effect of the pitch, the first formant frequency and the upper formant frequencies are all minimized for the F2 region.

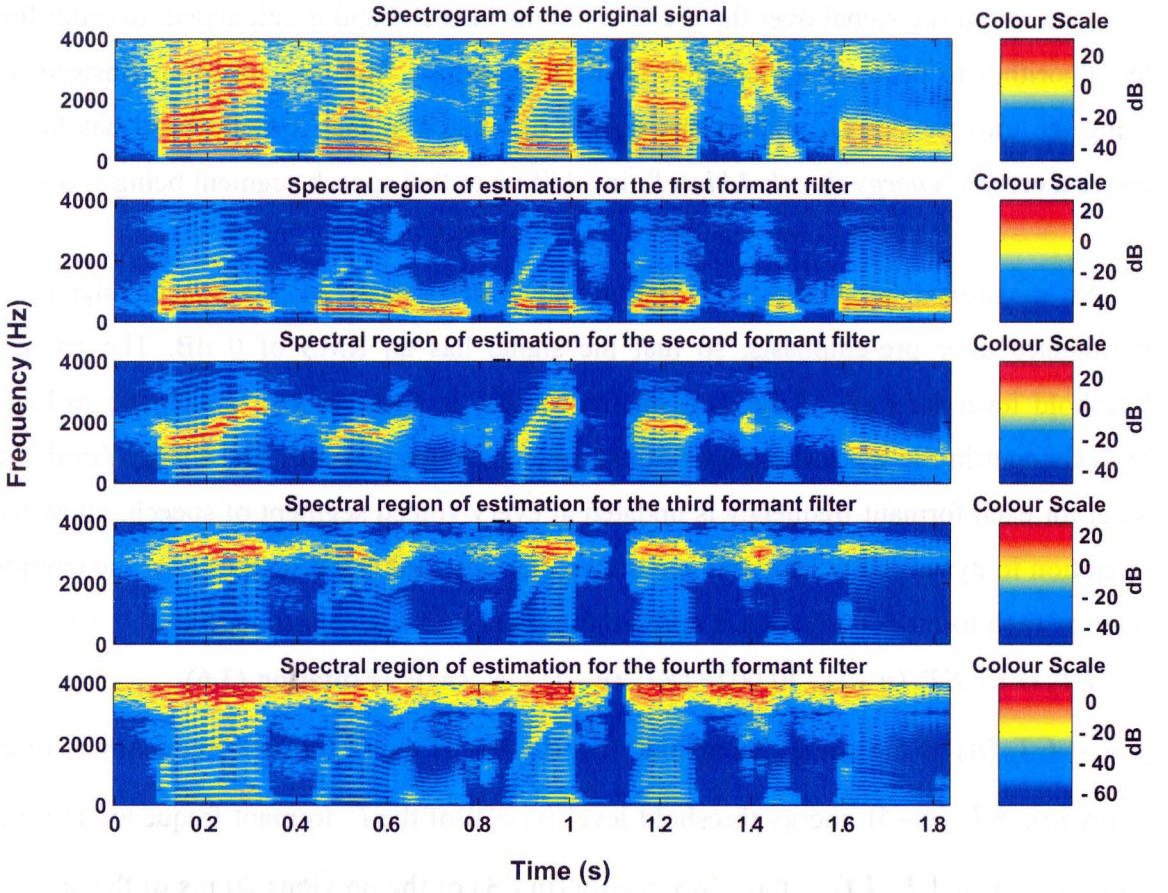

Figure 3 - 7 – Spectrograms of the original speech signal and the signals from the formant filterbank

3.4. Adaptive Energy Detector

After the speech signal has been filtered using the adaptive band-pass filterbank, the RMS energy of the signal over the previous 20 ms in each band is calculated. In order for the algorithm to estimate a particular formant frequency from the spectrum (instead of using the moving average value), the energy calculated in that formant band has to be above a certain '*energy threshold level*', in addition to that speech segment being voiced.

As mentioned in Section 3.1 the global RMS energy of the speech signal is normalised after pre-emphasis, so that the signal has an RMS of 0 dB. The energy threshold level for each of the formant frequencies is different and is adaptive to long term changes in the spectral energy of the formant frequency bands. The energy threshold level for each formant frequency is updated at every voiced segment of speech, allowing operation in dynamically changing environments. Equation 3.6 describes how the energy level of each formant frequency is updated during voiced segments of speech:

$$ET_{F_i}(n) = ET_{F_i}(n-1) - \left(0.002 * \left(ET_{F_i}(n-1) - E_{F_i}(n)\right)\right) \quad \textbf{Equation (3.6)}$$

where $ET_{F_i}(n)$ is the energy threshold level (in dB) of the i^{th} formant frequency at time index *(n)*, $ET_{F_i}(n-1)$ energy threshold level (in dB) of the i^{th} formant frequency at time index *(n-1)*, and $E_{F_i}(n)$ is the RMS energy (in dB) of the previous 20 ms of the speech signal.

The energy in each band is calculated independently of the energy of the other bands. Therefore, it is possible for the energy in some of the bands to be below their threshold level and the energy in other bands to be above the threshold levels concurrently. This scenario results in one or more of the formant frequencies being spectrally estimated, while others revert to their moving average value. Keeping the threshold levels and the energy calculations in each of the frequency bands independent allows accurate formant estimation in at least a few of the formant bands when there is low energy in only some

of the frequency bands. If there are long term changes in the energy of a formant band, the threshold level adapts to these energy changes gradually. Not changing the threshold levels abruptly prevents long term errors to the energy detector and allows the algorithm to recover quickly from brief loud sounds.

The threshold levels are measured in decibels and the initial energy threshold levels are set at the start of the algorithm and updated at voiced segments of speech. Various initial threshold levels were tested and the best results were obtained using the following initial threshold levels:

Initial F1 Energy Threshold Level = –35 dB;
Initial F2 Energy Threshold Level = –40 dB;
Initial F3 Energy Threshold Level = –45 dB;
Initial F4 Energy Threshold Level = –50 dB.

It is important to note that these initial values are calibrated for speech signals whose energy levels have been normalised to have a mean of 0 dB. If the signal energy is not normalised, the algorithm would require some time to adapt to the actual levels of energy present in each formant frequency band, before normal operation of the algorithm can resume. The variation of the energy threshold levels for the four formant filters throughout an energy normalised speech signal is shown in Figure 3-8. The signal used is a synthesized speech signal for a female speaker saying "Five women played basketball".

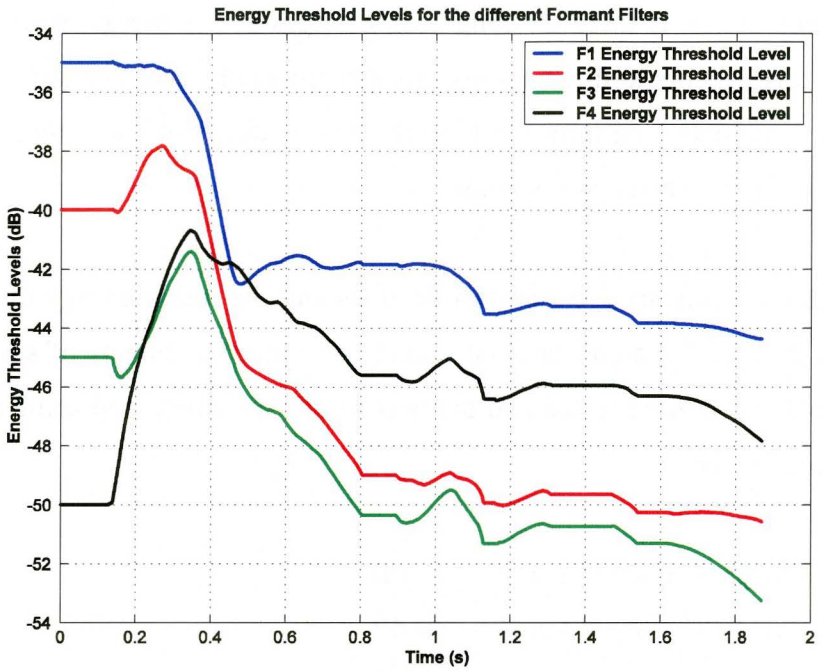

Figure 3 - 8 – Variation of the energy threshold levels through time for a female speaker speech signal: 'Five women playing basketball'

3.5. Calculating the Linear Predictor Coefficients

The idea behind linear prediction is to approximate each sample of the speech signal as a linear combination of past samples. A linear predictor of order p is defined as:

$$\tilde{S}[n] = \sum_{k=1}^{p} \alpha_k S[n-k] \qquad \textbf{Equation (3.7)}$$

where $\tilde{S}[n]$ is the prediction of $S[n]$ by the sum of p past weighted samples of $S[n]$. The system function of the p^{th} order predictor is a FIR filter of length p given by:

$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} \qquad \textbf{Equation (3.8)}$$

and the associated prediction error filter is:

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} = 1 - P(z) \qquad \textbf{Equation (3.9)}$$

The roots of the inverse of the prediction error filter corresponds to the poles placed to model the original signal as closely as possible while minimising the mean squared error between the estimated and original signals. First order linear prediction (p = 1) obtains one linear predictive coefficient and the corresponding single pole is placed to model the original signal as well as possible. Second order LPC tries to model the original signal using two poles, and so on [2]. The first four formant frequencies of the speech signal are estimated from the four filterbands of the adaptive bandpass filterbank using first-order LPC. The analytic signal from each of the bands is first windowed using a 20-ms periodic Hamming window and then the linear predictive coefficient (one per band) of the previous 20 ms of the windowed signal from each band is calculated. LPC tries to fit a single pole model to each signal and the location of the pole corresponds roughly to the vocal tract pole (formant frequency) in that band, for voiced segments of speech. The LPCs are only calculated from the bands if the entire previous 20-ms window of the speech signal is voiced (as determined by the voicing detector).

3.6. Voicing Detector

Figure 3-9 shows a block diagram of the voicing detector that has been designed for use with the formant tracking algorithm. The purpose of the voicing detector is to provide the formant tracking algorithm with a reliable sample by sample decision on whether a signal is voiced or unvoiced. Functionality has been built into the voicing detector to prevent it from switching its decisions spuriously. Parameters of the voicing detector need to be changed to be able to work for both male and female speakers. The gender detector provides regular updates to the voicing detector about the gender of the speaker so that the voicing detector can use the correct set of parameters.

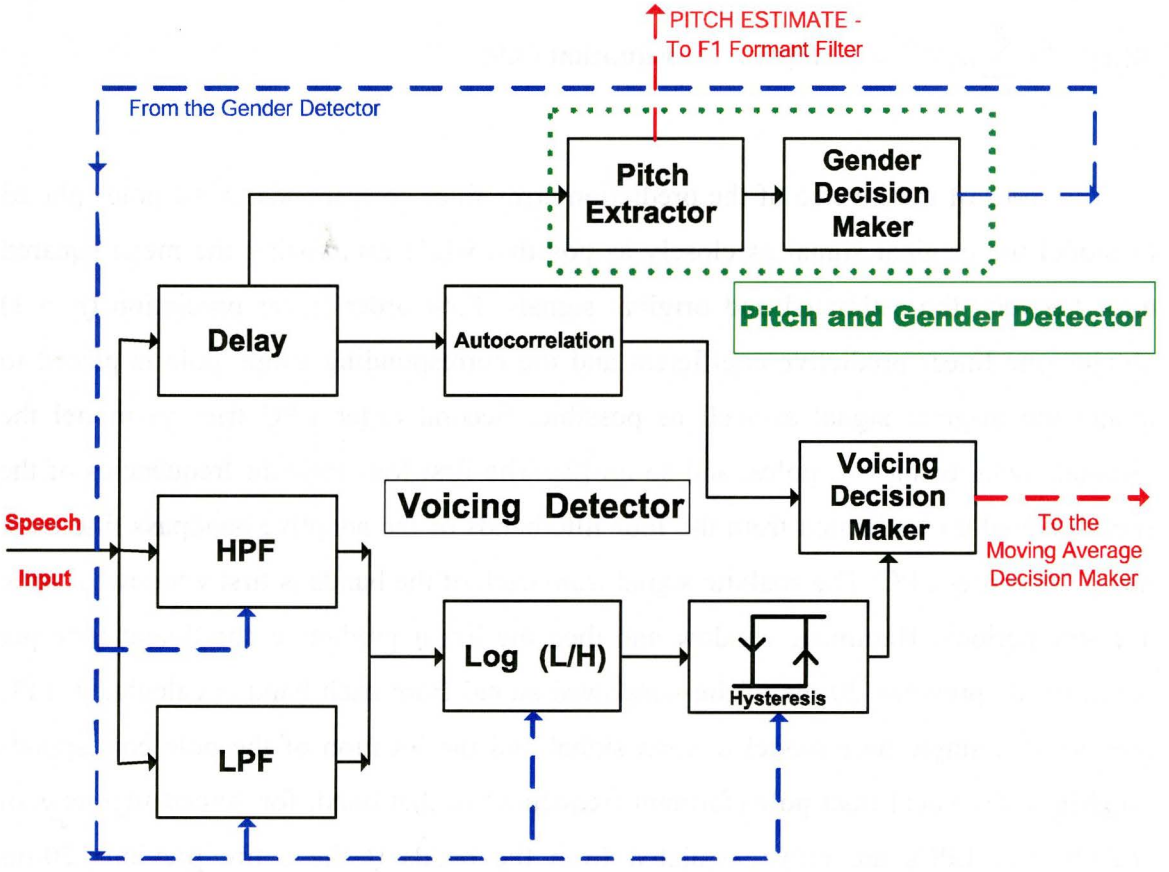

Figure 3 - 9 – Block Diagram of the Voicing Detector

3.6.1. The High Pass Filter and Low Pass Filter of the Voicing Detector

The original speech signal (the real valued signal) is filtered into two different frequency bands by passing it through a High-Pass Filter (HPF) and a Low-Pass Filter (LPF). Figure 3-10 shows the frequency and phase responses of the 20^{th}-order Butterworth HPF and LPF where the cut-off frequency of the two filters is set to 700 Hz (dotted black line).

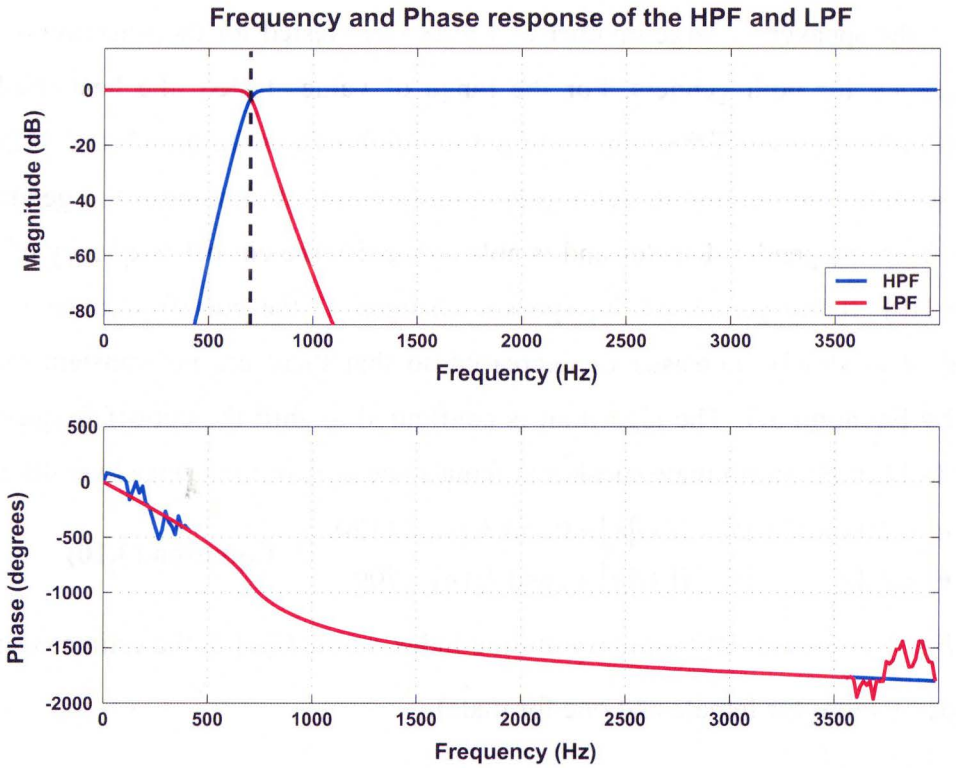

Figure 3 - 10 – The Frequency and Phase responses of the HPF and LPF

Once the signal is filtered into the two frequency bands, the log ratio of the RMS energy for the previous 20 ms of the signal, between the lower and the higher frequency bands is calculated. Voiced speech is made up of more lower frequency components than unvoiced speech, so the energy in the lower frequency band is expected to be greater than the energy in the higher frequency band during voiced speech. During voiced speech segments, the log ratio of the low frequency band to high frequency band is positive,

indicating that the energy in the lower frequency band is greater than that in the higher frequency band. The log energy ratio is calculated with a sliding window moving sample by sample and a windowed signal is classified as voiced if its log ratio exceeds a set threshold level. This energy ratio measure serves as the primary means of classification for determining if a speech segment is voiced or unvoiced.

The best value of the cut-off frequency of the HPF and the LPF depends on the gender of the speaker. A large number of values were tested for the selection of the cut-off frequency for both genders. For the range of values tested, the best results were obtained when the cut-off frequency was set to 700 Hz for male speech and 1120 Hz for female speech. The voicing detector gets updates every 20 ms about the gender of the speaker from the gender detector and is able to modify the cut-off frequency of the LPF and the HPF if the gender of the speaker changes. If the cut-off frequency is to be changed, it is slowly increased or decreased so that there are no transient effects, as shown by Equation 3.9. The algorithm is configured to shift the cut-off frequency from 700 Hz to 1120 Hz (from male speaker to female speaker) or vice versa over 40 ms.

$$\begin{aligned} F_c[n] &= F_c[n-1]+10, \quad \text{if } G[n]=0 \text{ and } F_c[n]<1120, \\ F_c[n] &= F_c[n-1]-10, \quad \text{if } G[n]=1 \text{ and } F_c[n]>700, \end{aligned} \qquad \textbf{Equation (3.10)}$$

where $F_c[n]$ is the cut-off frequency at time index n and $G[n]$ is the estimated gender at time index n (zero for female and one for male).

3.6.2. Threshold with Hysteresis

The log energy ratio used to determine if the input is voiced or unvoiced is reliable and accurate only for phonemes whose frequency components do not vary too much over time. The presence of transient frequencies in certain phonemes makes the log energy ratio unreliable on its own for determining voicing in continuous speech. This is because transient frequency components can make the voicing detector results oscillate too quickly between the voiced and unvoiced states. In order to avoid these fast oscillations

between the two states, Bruce et al. [12] proposed a threshold with hysteresis. This allows changes in the voicing state (from voiced to unvoiced or vice versa) only if the state of the current sample changes from the previous sample and the current sample has a log ratio greater than a set threshold level. These threshold levels depend on the gender of the speaker and have to be changed as the gender of the speaker changes.

If the previous sample is unvoiced and the current sample has a log ratio greater than a set threshold level (Log_Ratio_Threshold_Voiced), then the current sample is assigned as being voiced, i.e. the switch from unvoiced to voiced state occurs only if the log energy ratio is greater than the proper threshold level. If the previous sample is voiced and the current sample has a log ratio less than a set threshold level (Log_Ratio_Threshold_Unvoiced), then the current sample is assigned as being unvoiced, i.e. the switch from voiced to unvoiced state occurs only if the log energy ratio is below the proper threshold level. From the range of values tested, the best results were obtained when the level was set to 0.2 for males and 0.3 for females for Log_Ratio_Threshold_Voiced and 0.1 for males and 0.2 for females for Log_Ratio_Threshold_Unvoiced. The gender of the speaker is checked every 20 ms to confirm that the proper set of parameters are being used. If the gender of the speaker changes, the threshold levels are updated slowly over 40 ms to avoid any transient effects.

3.6.3. Autocorrelation Test

The contribution of energy due to AWGN over short time durations may not be 'white', but instead be 'coloured' (be randomly concentrated in the lower or upper frequency band). Voicing decisions based solely on the log ratio measure would rely only on the energy distribution of the signal over the previous 20 ms of data. If the short-term energy from AWGN is concentrated in the lower frequency band, the log energy ratio will erroneously detect the signal as being voiced. In order to avoid the problem of

erroneous voicing detection in the presence of AWGN, the voicing detector algorithm performs an autocorrelation based test to check if the energy in the lower frequency band of the signal is due to AWGN or due to some other non-random signal. The autocorrelation of the previous 20 ms of the signal is calculated. The signal is classified as voiced, if the autocorrelation at any lag ($\tau \neq 0$) is greater than the autocorrelation_threshold_multiplier times the autocorrelation at zero ($\tau = 0$) and there is at least one point in the window whose autocorrelation is greater than 0. If the low frequency energy in the signal is determined to be due to AWGN, the autocorrelation of the signal will be very low since random signals have very low or zero autocorrelation values (when $\tau \neq 0$), and the above test will fail. The value of the autocorrelation_threshold_multiplier is different for male and female speakers. Through trial and error the best results were obtained when the autocorrelation_threshold_multiplier was set to 0.25 for female speakers and 0.6 for male speakers. The gender of the speaker is checked every 20 ms and the value of the autocorrelation_threshold_multiplier can be changed if the gender of the speaker changes.

3.6.4. Voicing Detector Testing and Results

The testing of the voicing detector algorithm was conducted using both synthesized sentences and recorded speech sentences from the TIMIT database. Testing using the synthesized sentences allows quantitative measurements of the performance of the voicing detector for both male and female speakers since the exact time of the onset of voicing is known. In tests using the TIMIT database sentences, it is only possible to visually gauge the performance and accuracy of the voicing detector (using the spectrogram) since the exact time of the onset of voicing is unknown. Figures 3-11 and 3-12 show the performance of the voicing detector for the male and female synthesized sentence "Five women played basketball". The dotted black line indicates the actual voicing information from the synthesized speech parameters and the solid black line shows the estimate of voicing obtained through the voicing detector. When the lines are at zero ('low'), it indicates that the speech is unvoiced and when the lines are non-zero ('high'), it indicates that speech is voiced.

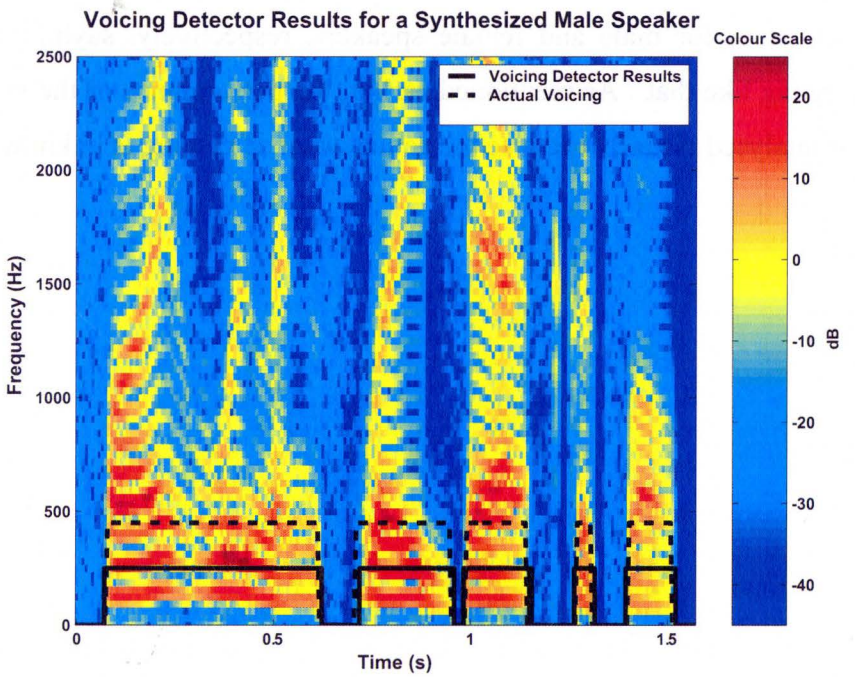

Figure 3 - 11 – Voicing Detector results for a synthesized male speaker

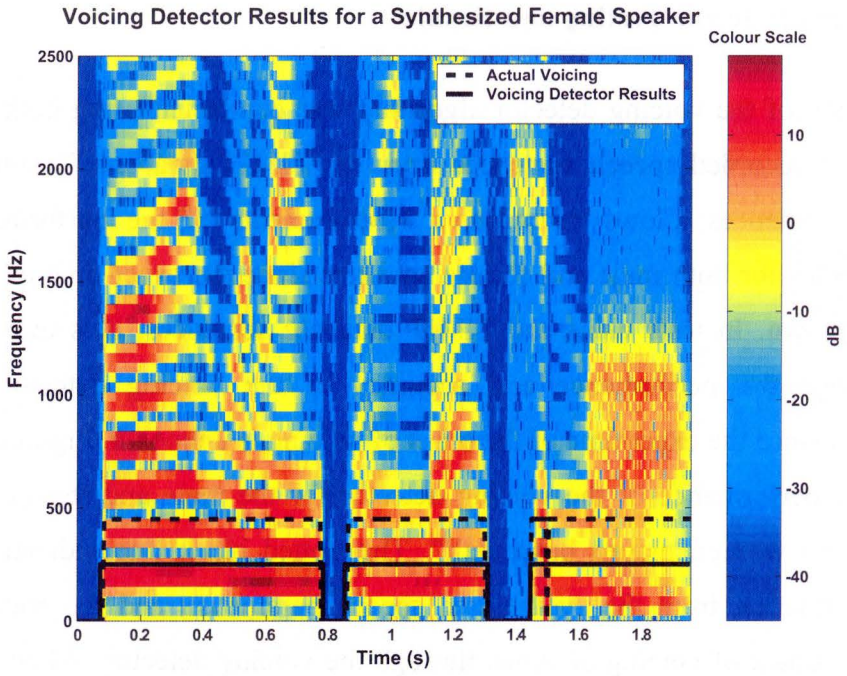

Figure 3 - 12 – Voicing Detector results for a synthesized female speaker

Figures 3-13 and 3-14 show the results of the voicing detector applied to TIMIT database sentences for male and female speakers, respectively, saying 'Don't ask me carry an oily rag like that'. As mentioned earlier, the performance of the voicing detector can only be analysed visually because the actual onset of voicing is unknown.

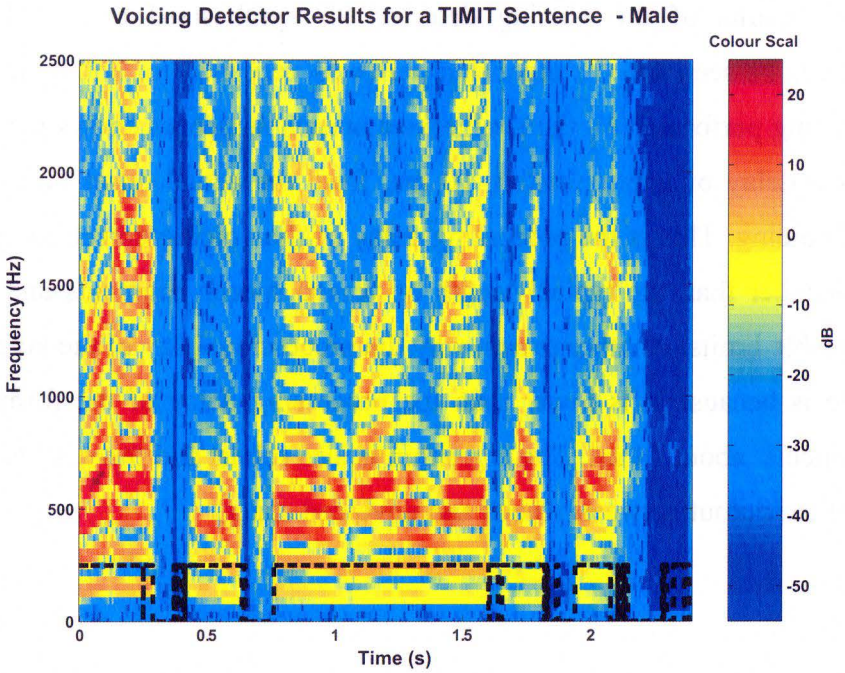

Figure 3 - 13 – Voicing Detector results for a male speaker from TIMIT database

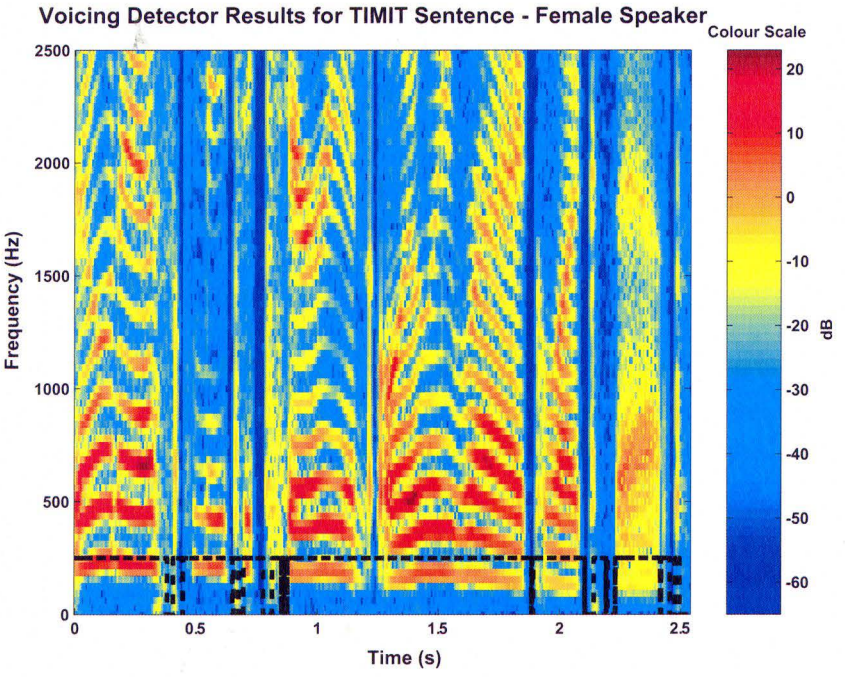

Figure 3 - 14 – Voicing Detector results for a female speaker from TIMIT database

From the results of the voicing detector for synthesized and TIMIT database sentences, it can be seen that the algorithm performs very well for both male and female speakers. Testing performed on the synthesized speech sentences shows that the voicing detector has a delay of approximately 10 ms, from the actual onset of voicing to the detection of voicing. This is the processing delay of the voicing detector algorithm. The duration is smaller than the typical length of voiced speech segments and is therefore within acceptable limits. Another reason why the processing delay of the voicing detector is acceptable is because it is lower than the processing delay of the formant tracking algorithm which is about 14 ms. The algorithm is also robust and there is very little or no oscillation of the output between voiced and unvoiced states.

3.7. Gender Detector

The difference in pitch between male and female speakers is sufficient to serve as a discriminating parameter between the two types of speakers. The gender detector calculates the pitch and determines the gender of the speaker. It provides this information to the voicing detector so that the voicing detector can update its parameters to work properly for both male and female speakers. Accurate pitch estimation from continuous speech is a difficult task to accomplish. Several complicated algorithms have been proposed to achieve this task [2] [3] [15]. However, for the purposes of constructing a low-computation and low-delay gender detector, it was deemed sufficient to have an approximate estimate of the pitch as long as there is still clear discriminability between male and female speakers. Therefore, a well known fast and simple approach to pitch estimation is chosen that uses an autocorrelation based approach [2] [14].

In the gender detector algorithm, pitch is estimated from the real valued speech signal using the short-time autocorrelation of the previous 60 ms of the signal. The 60 ms signal is divided into non-overlapping frames whose length must be greater than at least one pitch period in order to measure the pitch in the frame accurately. The gender detector algorithm segments the signal into non-overlapping 20 ms-frames. Each frame is low-pass filtered using a fourth-order Butterworth filter (LPF) to reduce the range of spectral estimation. The pitch information is contained within the lower frequencies of speech (< 1000 Hz) so the higher frequencies contained in the signal can be discarded. The frequency response of the LPF is shown in Figure 3-15.

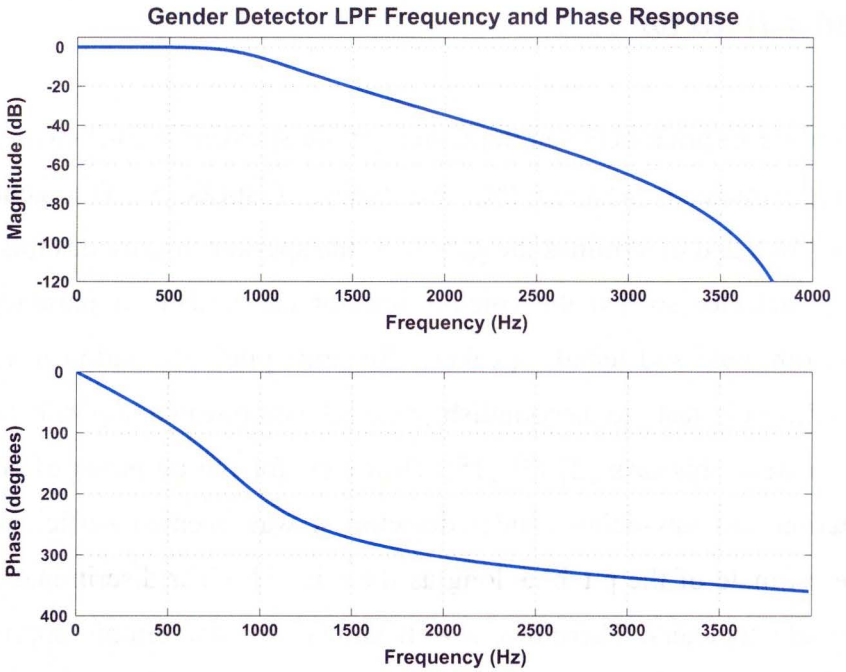

Figure 3 - 15 – LPF for the Gender Detector

3.7.1. Centre Clipping

There can be interaction between the pitch frequency and the first formant frequency when the first formant frequency bandwidth is narrow relative to the harmonic spacing. In such cases the autocorrelation function of the signal has higher peaks due to the vocal tract response (first formant frequencies) than due to the vocal excitation (pitch frequency). This makes it difficult to estimate the pitch frequency using short time autocorrelation [2]. To avoid this problem, a nonlinear time-domain technique called *centre-clipping* is used that makes the periodicity of the speech signal more prominent while suppressing the other features of the speech that contribute to the extra peaks of the autocorrelation function [16] [17]. The gender detector uses the three level centre clipping function shown in Figure 3-16 that was first proposed by Rabiner and Schafer

[16]. In this figure the clipping level is set to 68% of the maximum amplitude of the signal in that frame. A different clipping level is calculated and used for every frame.

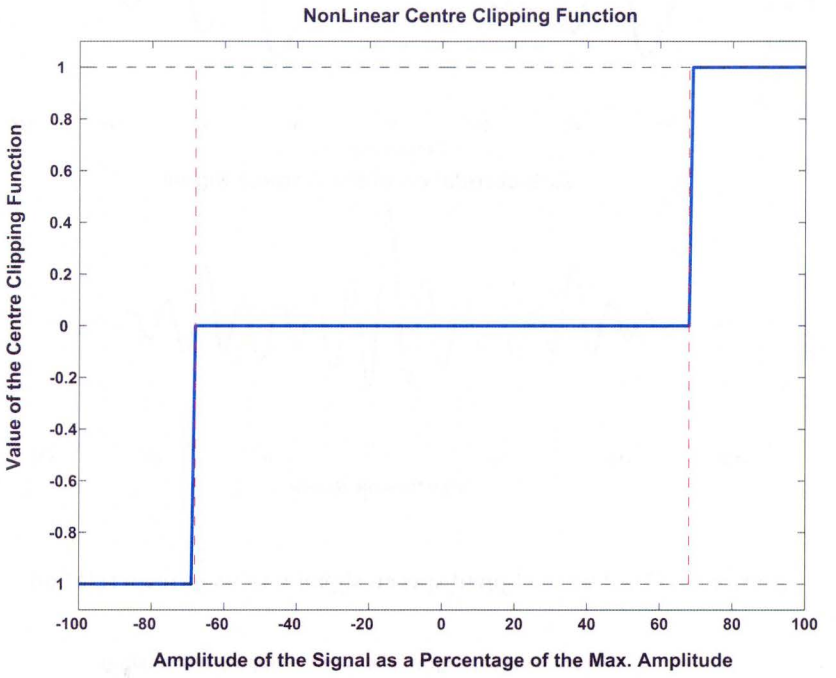

Figure 3 - 16 – Three level centre clipping function

Figure 3-17 shows an unclipped speech signal and its autocorrelation function while the centre clipped version of the same signal and its corresponding autocorrelation function is shown in Figure 3-18. The extra peaks in the autocorrelation function that do not represent the vocal excitation are removed by centre clipping the signal.

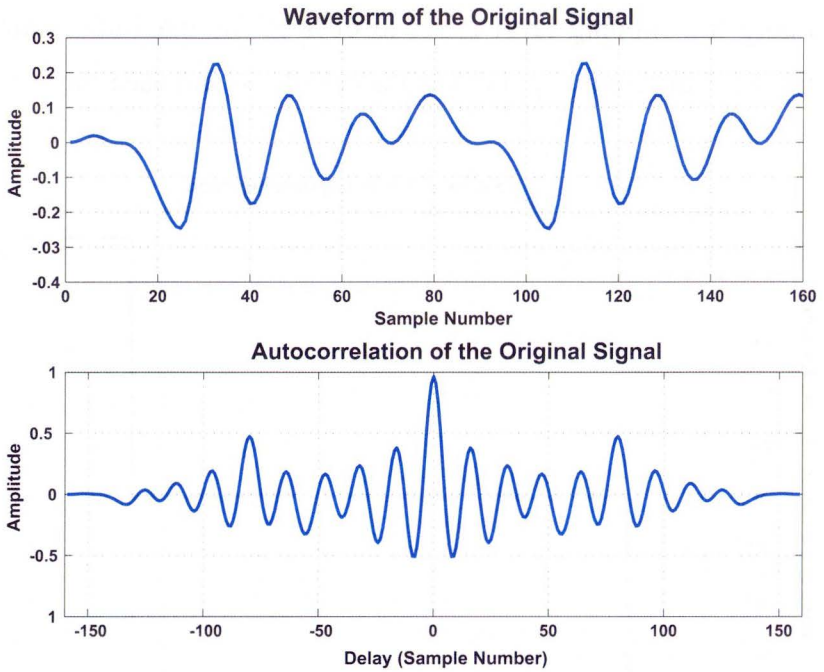

Figure 3 - 17 – The unclipped speech signal and its autocorrelation

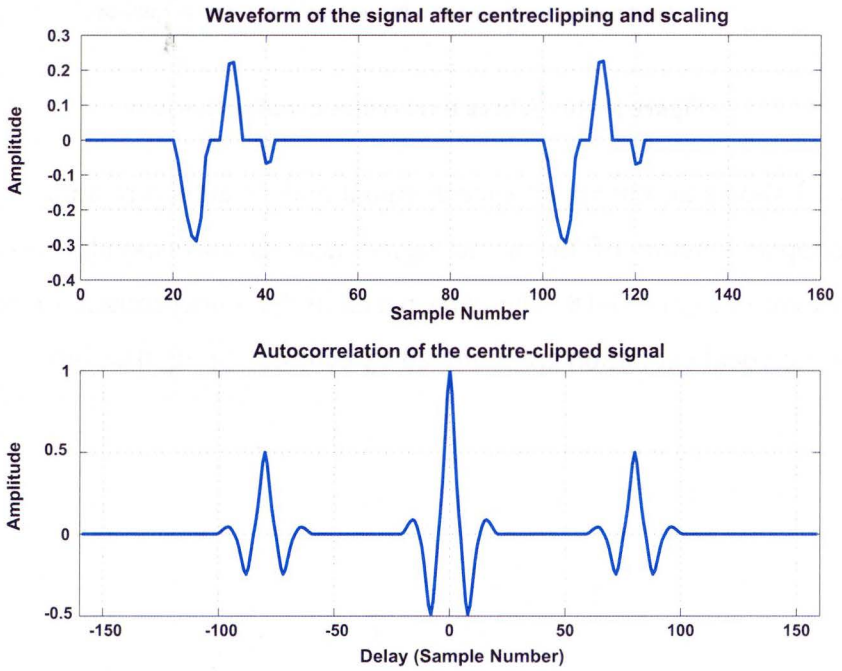

Figure 3 - 18 – Centre-clipped speech signal and its autocorrelation

The problem with three-level centre clipping is the determination of the proper clipping level, C_L, for each frame (indicated by the dotted red lines and set to 68% of the peak amplitude in the signal in Figure 3-16). It is possible for the amplitude of the signal to vary significantly within a frame. If the clipping level is set to too high a percentage of the maximum amplitude in that frame, most of the signal may be lost due to clipping. To avoid this problem, the clipping level is set to 68% of the average peak amplitude in the first third and last third segments of each frame.

3.7.2. Determination of the average pitch period and the gender of the speaker

After the signal has been centre clipped, its autocorrelation, R_n, is calculated and the location of the highest peak, p, of the autocorrelation function is located. If $R_n(p)$ is less than 0.4 x $R_n(0)$, then the segment is classified as being unvoiced and its pitch is set to 0 Hz. Otherwise, the pitch period is calculated as being the location of the highest peak of the autocorrelation function. The range of acceptable values for the pitch frequency is between 60 and 320 Hz, and if the calculated value of the pitch is outside this range then it is set to the moving average value of the pitch in that segment. The value of the pitch frequency in each frame is used to calculate the average pitch frequency of each segment of the signal (of 60 ms duration) passed to the gender detector algorithm. The average pitch frequency of each segment is sent to the first formant filter to be used for the placement of the additional zero at the pitch frequency location. The gender $G[n]$ of the speaker is considered to be male ('0') if the average pitch frequency is below 180 Hz and is set to female ('1') if it is above that value. Figure 3-19 shows the results of the gender detector algorithm for a female speech signal.

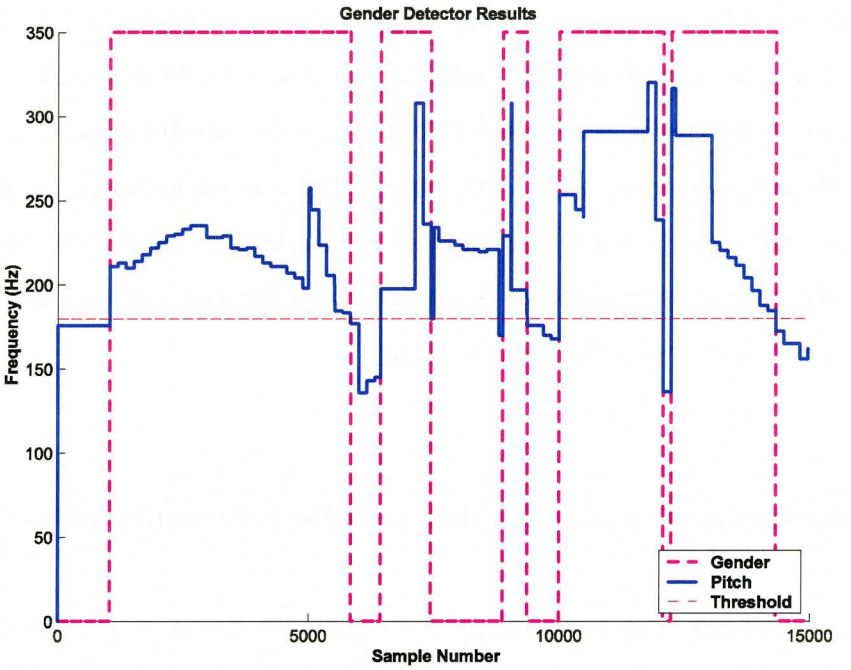

Figure 3 - 19 – Gender Detector results for a female speaker speech signal

In Figure 3-19 the gender of the speaker is represented by the dotted pink line where a 'high' represents a female speaker and a 'low' represents a male speaker. The pitch frequency is shown in blue and the threshold level for the gender classification is shown in red. The gender detected for the speaker changes through the test signal even though the original signal is actually only for a female speaker. This erroneous gender detection occurs when the pitch frequency dips below the threshold level (180 Hz) and the algorithm classifies the speaker as being male. Most of the erroneous gender detection occurs when the speech is actually unvoiced and therefore does not affect the performance of the overall formant detection algorithm (see sections 3.6 and 3.8). Due to this reason, during unvoiced speech segments, the gender of the speaker is assumed to remain unchanged.

3.8. Moving Average Decision Maker

The moving average decision maker has two main purposes:

- to calculate and update the moving average value of each formant frequency and
- to determine whether to assign the LPC estimated value or the moving average value to each formant frequency.

The moving average decision maker assigns the estimated value to the formant frequencies (from the LPCs) only when the segment is voiced and the energy of the formant frequency is above its respective threshold level (see section 3.4). If the segment is unvoiced or if the energy of a particular formant is below its respective threshold level, then the current value of the formant frequency decays toward the moving average value for that formant frequency according to:

$$F_i[n] = F_i[n-1] - \left(0.002 * \left(F_i[n-1] - F_i^{\mathrm{MA}}[n-1]\right)\right) \qquad \textbf{Equation (3.11)}$$

where $F_i[n]$ is formant estimate the i^{th} formant frequency at time index *(n)* and $F_i^{\mathrm{MA}}[n-1]$ is the previous value of the moving average for the i^{th} formant frequency.

Equation 3.12 describes the update rule for the moving average value of each formant frequency:

$$F_i^{MA}[n] = \frac{1}{n}\sum_{k=1}^{n} F_i[k] \qquad \textbf{Equation (3.12)}$$

where $F_i^{MA}[n]$ is the moving average value for the i^{th} formant frequency at index n and $F_i[n]$ is the estimate of the i^{th} formant frequency at index n.

3.9. Other Considerations

3.9.1. Limitations on the proximity of formant frequencies

The filter response of the formant filterbank becomes poor when the location of the poles and zeros are very close. Therefore, the formant tracking algorithm limits how close the formant frequencies can come to each other. The algorithm does not allow F1 to be less than 150 Hz from the pitch frequency and any estimate of F1 that is less than 150 Hz from the pitch is set to be pitch+200 Hz. F2 is also limited from being less than 300 Hz from F1. Any F2 values that are less than 300 Hz from F1 are set to be F1+400 Hz. Similarly, F3 is not allowed to be less than 400 Hz from F2. All values of F3 that are less than 400 Hz apart from F2 are set as F2+400 Hz. Finally, all F4 values that are less than 400 Hz from the F3 values are set as F3+400 Hz. This limitation on the proximity of the formant frequency values ensures that the poles and zeros of the formant filterbank are never too close to cause problems to the frequency response of the filterbank. Figure 3-20 shows the algorithm used for updating the formant frequencies values when they are too close to each other.

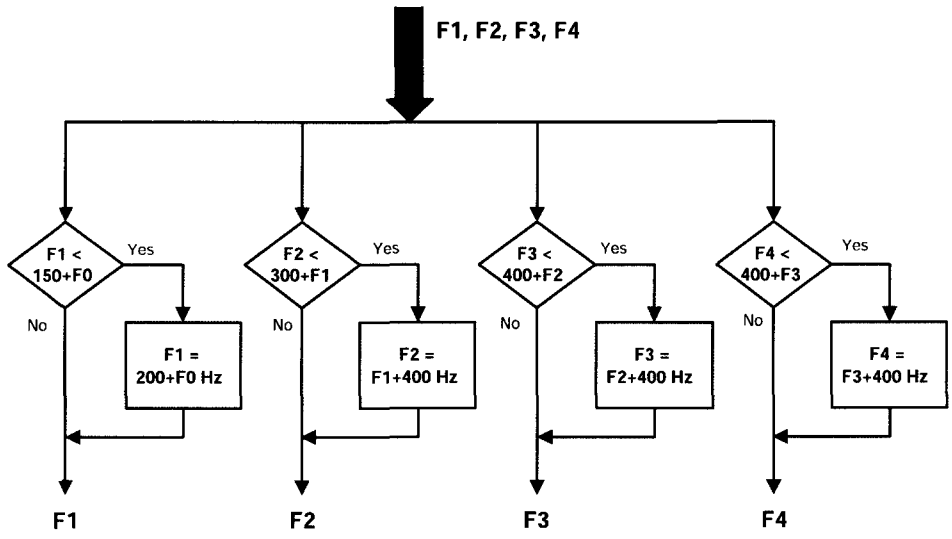

Figure 3 - 20 – Update rules for the formant frequency proximity

4. TESTING REGIME AND RESULTS

The primary goal of this project is to develop a reliable formant tracking algorithm that is robust in real-life noise scenarios. To this end, rigorous and systematic testing of the formant tracking algorithm has been conducted in order to find best values for the operating parameters as well as to ensure that the algorithm performs well under various levels of different background noise. In this chapter, the different test cases are described and the performance of the algorithm under these conditions is discussed.

The algorithm has been tested using synthesized speech signals as well as speech signals from the TIMIT recorded speech database. Testing using synthesized sentences allows quantitive analysis of the performance of the formant tracker because the formant frequency values of the synthesized speech signals are known. The testing and analysis of the formant tracking algorithm for synthesized speech signals has been fully automated (using `MATLAB` scripts) to increase the efficiency of the testing regime and help speed up analysis of the results. These scripts test the formant tracking algorithm for over 80 different scenarios and then determine the RMS error between each actual and estimated formant frequency. The TIMIT database speech signals are recorded from actual speakers and therefore sound more natural than the synthesized speech signals. However, the actual formant frequency values of the TIMIT database speech signals are unknown, therefore, only qualitative analysis of the results can be performed through visual inspection.

Formant Frequency Initializations

There are a large number of values that have to be properly initialized before the formant tracking algorithm can start to estimate the formant frequencies from the speech signal. The formant frequency values are initialized to random values with a set mean and standard deviation, until the algorithm is able to estimate the formants from the speech signal. The initial values assigned to the formant frequencies are important because they will be used to determine the exact shape of the formant filterbank (at the start) that will be used to estimate the formant frequencies from the signal. If the initial formant frequencies assigned are not close to the actual formant frequency values, then the filtered spectrum from each of the filters will not contain the proper formant regions, and the resulting estimated formant frequencies will not be accurate. The initial formant values are set using the following general equation:

$F_i^{IV} = \alpha \tilde{F}_i + \bar{F}_i$ **Equation (4.1)**

where F_i^{IV} is the initial value of the ith formant frequency, α is a normally distributed random number with a standard deviation of 1, $\tilde{F}_i$ is the standard deviation of the ith formant frequency and $\bar{F}_i$ is the mean value of the ith formant frequency.

The mean ($\bar{F}_i$) and the standard deviation ($\tilde{F}_i$) for the initial values of each formant frequency are obtained from the actual formant frequency values for a large selection of synthesized speech sentences. The initializations for the four formant frequencies and the pitch frequency are:

Initial Pitch Frequency = (α x50) + 175 Hz
Initial First Formant Frequency = (α x 115.9433) + 397.3253 Hz
Initial Second Formant Frequency = (α x 461.5834) + 1490 Hz
Initial Third Formant Frequency = (α x 381.7358) + 2490 Hz
Initial Fourth Formant Frequency = (α x 258.653) + 3550 Hz

Calculating the RMSE

Quantitive error is measured in terms of the root mean squared error between the actual and estimated formant frequencies, for voiced segments of speech when there is sufficient energy in the signal for the algorithm to estimate formant frequencies through spectral estimation. A function was created that finds the time indices, *i*, for which the algorithm uses spectral estimation to obtain the formant frequencies of a signal. When background noise is present in the signal the indices are calculated for the speech signal in the presence of the noise. Therefore, if there is sufficient background noise energy present in the signal, the indices will also include samples where the background energy causes the algorithm to use spectral estimation even though the actual speech signal may be unvoiced or may have insufficient energy. The inclusion of these points in the RMSE calculation means that if the formant tracker starts tracking the formant frequencies of a background speaker when the primary speaker is silent, the overall error of the algorithm will rise. Such a scenario only arises at very low SNRs and adds to the high RMSE observed for the algorithm at these SNRs.

For some applications, it is not important what the formant frequency algorithm tracks when the primary speaker is silent and therefore the error indices should not include these points. For such applications, a better way to gauge the performance of the algorithm would be to measure RMSEs only for those time indices where the primary speaker's speech is voiced and has sufficient energy for spectral estimation, regardless of the background noise. This can be accomplished by using the existing RMSE calculation function and finding out the error indices, *j*, for the 'clean' signal (without any background noise). Then the RMSE should be calculated using these time indices no matter what the amount of background noise present. This new suggested method for calculating the RMSEs should show better performance of the formant tracking algorithm in lower SNRs than shown in the discussion in this thesis.

4.1. Testing with White Noise

Additive White Gaussian Noise (AWGN) may be present in real-life environments from a variety of sources such as fans, air-conditioners, running water, etc. Since the formant tracker is to be implemented and used in a real-life environment, it must be able to operate in AWGN. The operation of the algorithm is tested and analysed in the presence of background AWGN at various Signal-to-Noise Ratios (SNRs), from 40 dB to -10 dB, for various synthesized and TIMIT database speech signals (for both male and female speakers). AWGN adds wideband spectral noise to each of the four formant bands and the long-time average energy added to each of the bands is roughly equal. Due to the equal energy contribution of AWGN on the formant frequency bands and the nature of the formant tracker, the performance of the formant tracking filters should not be affected greatly in AWGN for voiced segments of speech. However, the performance of the voicing detector and the pitch will both be adversely affected due to AWGN at low SNRs because of the added energy in the lower frequency bands. The voicing detector in particular may erroneously detect voicing during unvoiced segments of speech. The addition of the autocorrelation based testing as well as the adaptive energy thresholds in the voicing detector (described in section 3.6) prevents this from occurring.

Figure 4-1 shows the spectrogram of a female synthesized speaker saying "Five women played basketball" in AWGN at a SNR of 40 dB. The figure also shows the original formant frequencies (plotted in black), the estimated formant frequencies (plotted in white) as well as the voicing decisions (plotted in purple). It can be seen that at this high SNR level the formant frequencies are estimated accurately and the voicing detector estimates detects voicing accurately. As the formant frequencies change, the formant tracker is able to follow them and capture the formant frequency transitions. Figure 4-2 shows the spectrogram of a synthesized male speaker saying the same sentence in AWGN at a SNR of 40 dB.

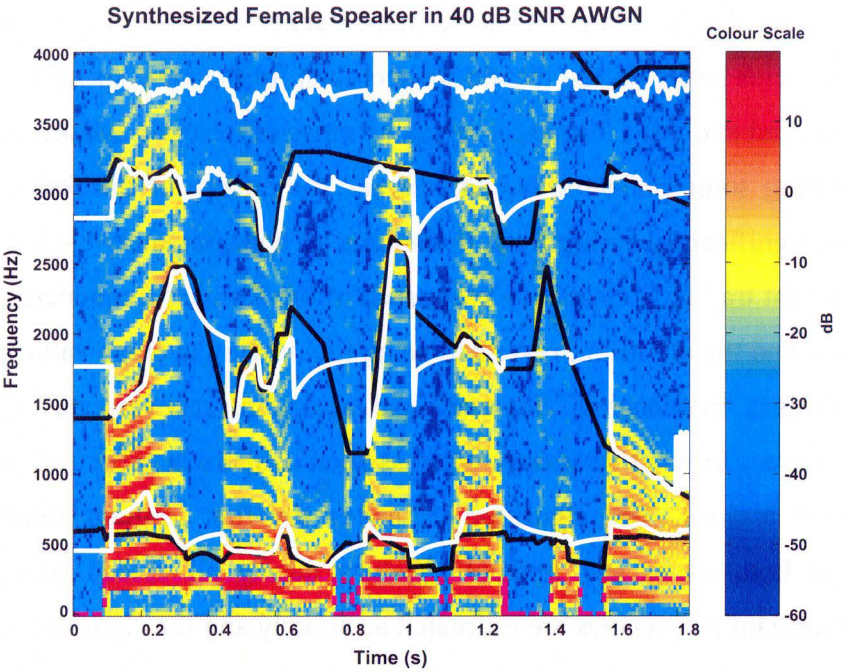

Figure 4 - 1 – Spectrogram for a synthesized female speaker in AWGN at 40 dB SNR

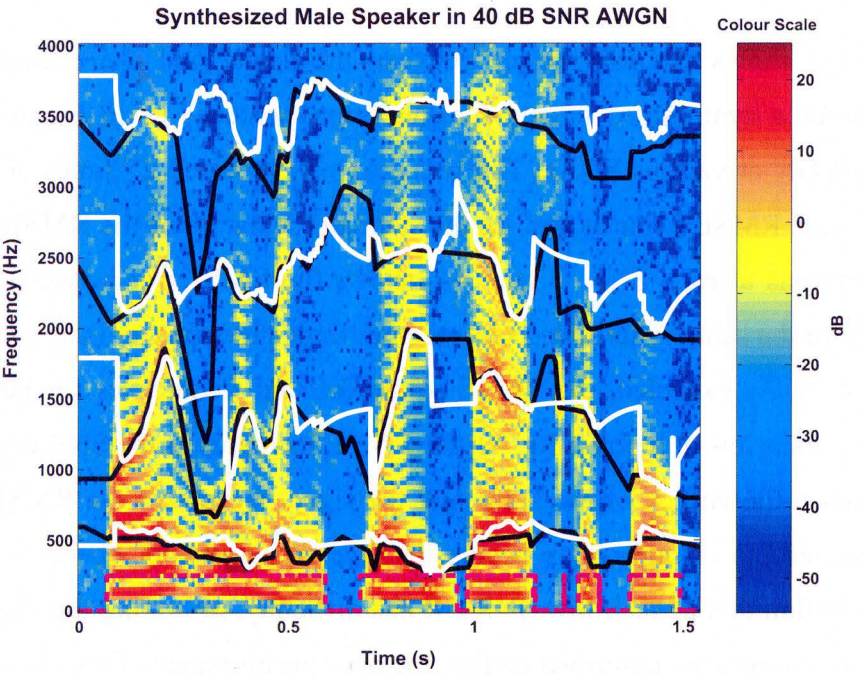

Figure 4 - 2 – Spectrogram for a synthesized male speaker in AWGN at 40 dB SNR

It can be seen that at this SNR the formant tracker estimates the second and third formant frequencies better for the female speaker than for the male speaker. This happens because the second and third formant frequency values of the male speaker have very fast transitions during some phoneme boundaries and the energy of the formant frequency regions drops significantly during these transitions (at approximately t = 0.25 sec and 1.4 sec). The formant tracker is unable to keep track of the formant frequencies during these fast transitions and the algorithm reverts to using the moving average value of the second and third formant frequencies. However, the algorithm recovers quickly and starts tracking the correct formant frequency when as soon as there is sufficient energy present in the formant regions. The algorithm estimates the first formant frequencies quite accurately for both males and females speakers during all voiced speech segments. Overall, the formant frequencies are estimated accurately and the algorithm is robust. The voicing detector performs well in predicting the voiced segments of speech for both male and female speakers.

Figure 4-3 shows the RMS error between the actual and the estimated formant frequencies for a synthesized female speaker (same sentence as in Figure 4-1) in the presence AWGN at various SNRs. The test case was repeated 25 times and the figure shows the mean RMSE value as well as the standard deviation of the RMSE results over the 25 repetitions at each SNR. Figure 4-4 shows the RMS error between the formant frequencies for the same synthesized sentence but for a male speaker (same sentence used for Figure 4-2). As expected, the RMSE of the formant frequencies for both male and female speakers is low at high SNRs, but the RMSE increases as the SNR decreases. At 0 dB SNR (when the energy of the noise and the signal are equal) the RMSE for all the formant frequencies is high. This occurs because the algorithm operates without any noise cancellation and at such low SNRs, it has difficulty separating the signal from the noise. Due to the reasons described earlier, the poor performance of the algorithm for the synthesized male speaker can also be seen and overall the performance of the algorithm for male speakers is generally worse than for female speakers.

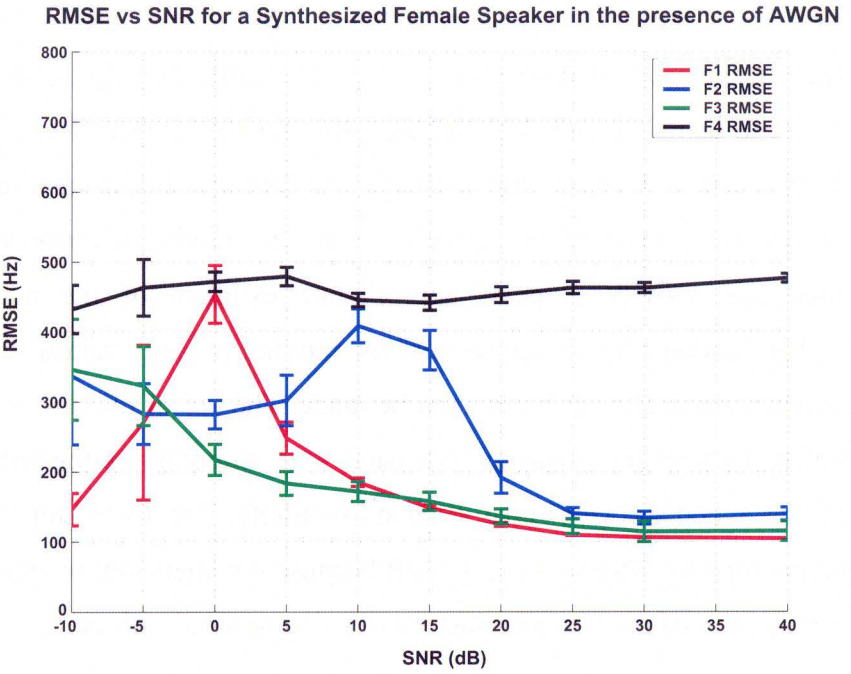

Figure 4 - 3 – RMSE vs. SNR for a synthesized female speaker in AWGN

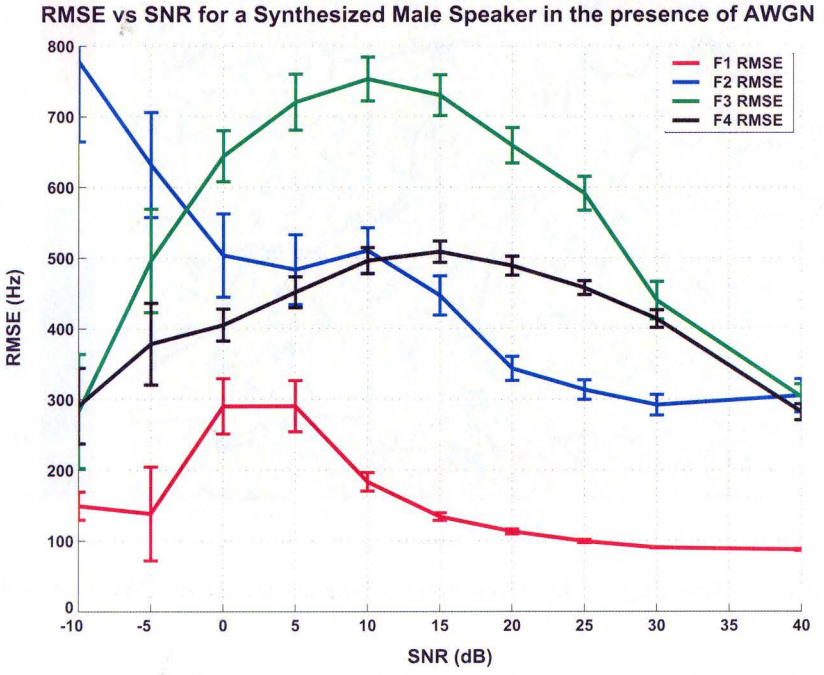

Figure 4 - 4 – RMSE vs. SNR for a synthesized male speaker in AWGN

Figure 4-5 shows the spectrogram of a female synthesized speaker saying "Five Women Played Basketball" in AWGN at a SNR of -5 dB. The figure shows that the algorithm reverts to using the moving average value for the formant frequencies due to the presence of noise. It is clear that although the formant frequencies are not being spectrally estimated for most of the speech signal, the moving average values of the formant frequencies provide a good approximate estimate of the actual formant frequencies. This validates the assignment of the moving average values to the formant frequencies when a spectral estimate can not be made. The use of moving average values also ensures that the formant frequency estimates vary smoothly as the speech changes from voiced to unvoiced and vice versa, even at low SNRs. The RMSE for both male and female speakers drops for SNRs less than 0 dB because the algorithm reverts to using the moving average values of the formants instead of using spectral estimation.

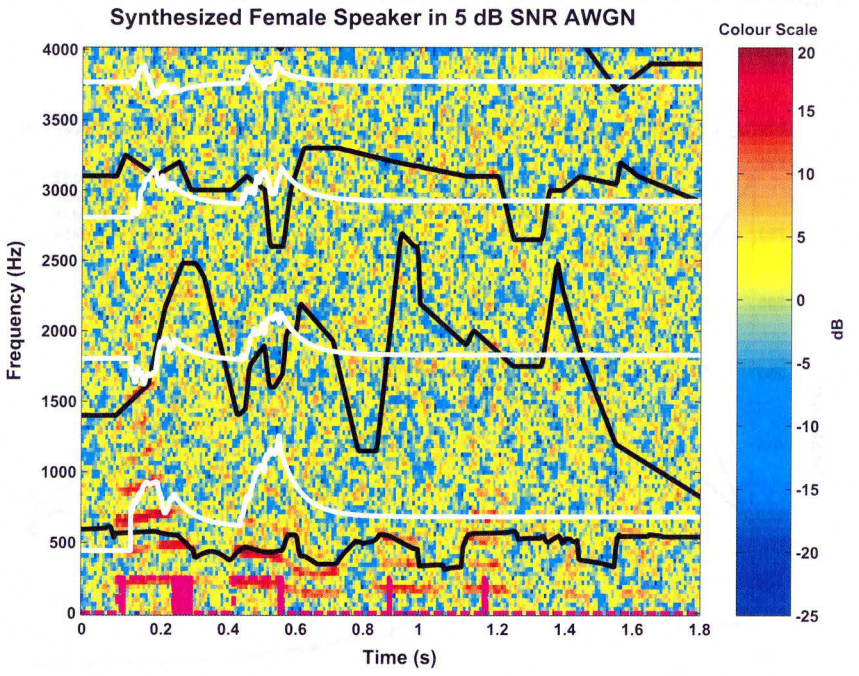

Figure 4 - 5 – Spectrogram for a synthesized female speaker in AWGN at -5 dB SNR

The algorithm was also tested using recorded natural speech for both male and female speakers from the TIMIT database. Figure 4-6 shows the spectrogram and the estimated formant frequencies for a natural female speaker from the TIMIT database saying "Don't ask me to carry an oily rag like that" in the presence of background AWGN at a SNR of 30 dB. From the spectrogram it can be visually observed the formant tracker is able to detect and track the formant frequencies relatively well and also makes good voicing decisions. The formant frequency transitions are also captured well by the algorithm as can be seen from the second formant frequency estimates for approximately t = 1.5 sec.

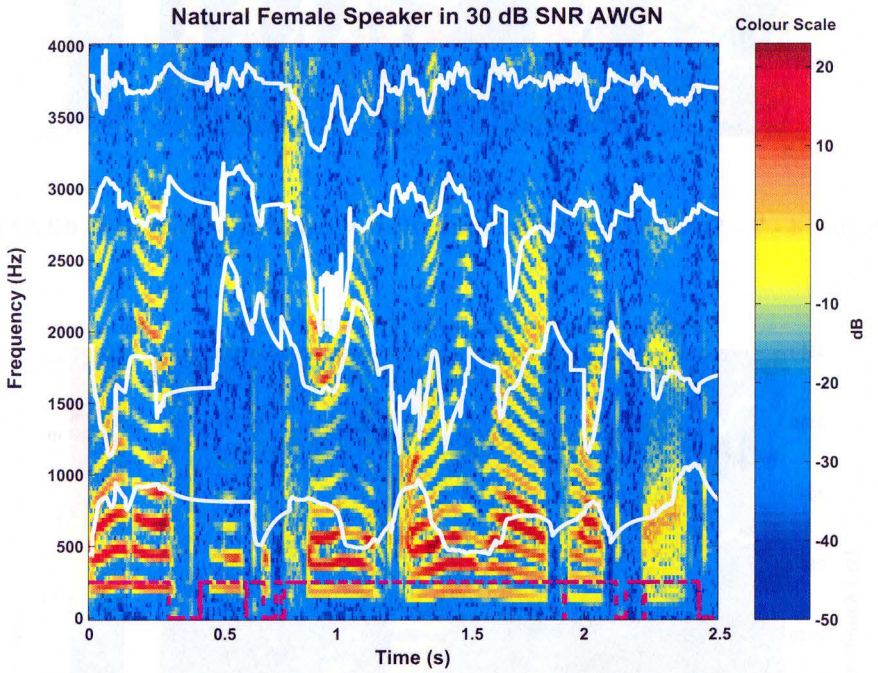

Figure 4 - 6 – Spectrogram for a natural female speaker in AWGN at 30 dB SNR

Figure 4-7 shows the spectrogram and the estimated formant frequencies for a natural male speaker from the TIMIT database saying "Don't ask me to carry an oily rag like that" in the presence of background AWGN at a SNR of 30 dB. Figure 4-8 shows a magnified version of the same spectrogram, illustrating the ability of the algorithm to track formants closely during phoneme transitions and to produce smooth formant frequency estimates as the speech switches between voiced and unvoiced segments.

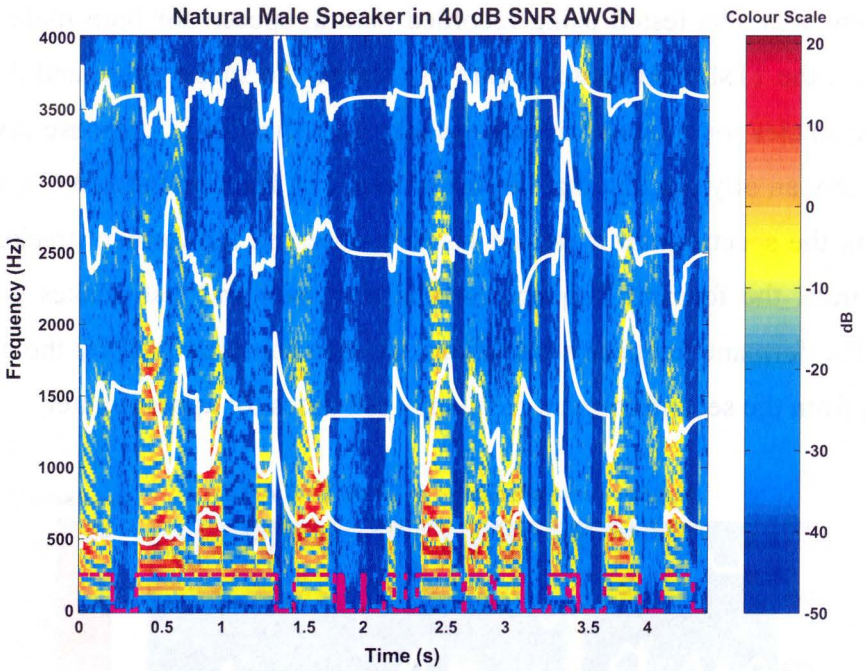

Figure 4 - 7 – Spectrogram for a natural male speaker in AWGN at 40 dB SNR

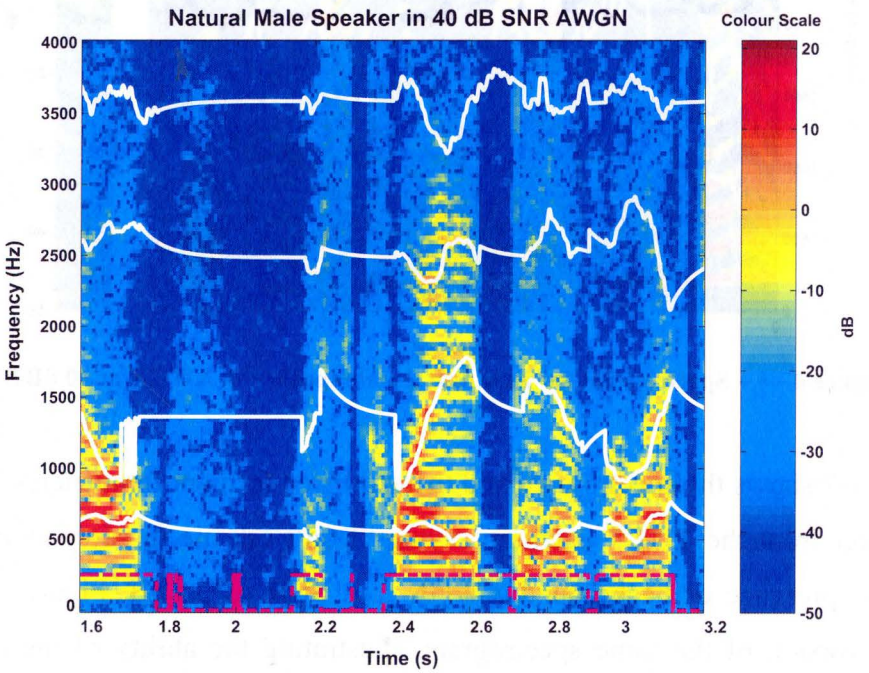

Figure 4 - 8 – Spectrogram for a natural male speaker in AWGN at 40 dB SNR (magnified)

4.2. Testing in the presence of a female single background speaker

In real-life there is often more than just one speaker present in an environment and the formant tracking algorithm has to be able to accurately estimate formant frequencies for the dominant speaker in the presence the background speakers. In this test case, the algorithm is evaluated in the presence of a female single background speaker where the background speaker serves as the 'noise source'. The loudness of the background speaker often varies in real-life and therefore the algorithm is tested at varying SNRs (from 40 dB to -5 dB). This scenario is challenging for the algorithm because over a particular short time period, the background speaker may contribute significant energy to the formant frequency regions of the primary speaker, especially at lower SNRs. This will cause the algorithm to start tracking the formant frequencies of the background speaker instead of those of the primary (more dominant) speaker.

There are short moments of silence during the speech of any speaker while the speaker inhales, or exhales, and during phoneme transitions etc. Another source of concern when there are background speakers present is that if the background speaker says something during the brief moments of silence of the primary speaker, the formant tracking algorithm may start to track the formant frequencies of the background speaker. In this case the formant frequencies estimated will switch back and forth between those of the primary and the background speakers. Another point to keep in mind is that the 'noise source' is a female speaker and this can lead to one of two scenarios when the primary speaker is male. The formant frequencies of the background female speaker are higher than those of the primary male speaker. This may lead the overall performance of the algorithm to be better for male speakers, because there will be less energy contribution to the male speakers' formant frequency regions. On the other hand, at low SNRs, the formant tracking algorithm may start tracking the formant frequencies of the background female speaker and push the RMSEs of the algorithm higher.

Figure 4-9 shows the spectrogram of a synthesized female speaker saying "Five women played basketball" in the presence of a female single background speaker saying "He sees the ball" at a SNR of 25 dB. It can be seen that at this high SNR the formant frequencies are estimated fairly accurately for most of the speech signal, except at time t = 1 sec. when both the second and third formant frequencies suddenly jump. This sudden jump may be due to the energy contribution of the background speaker while there was a momentary silence from the primary speaker, as discussed above.

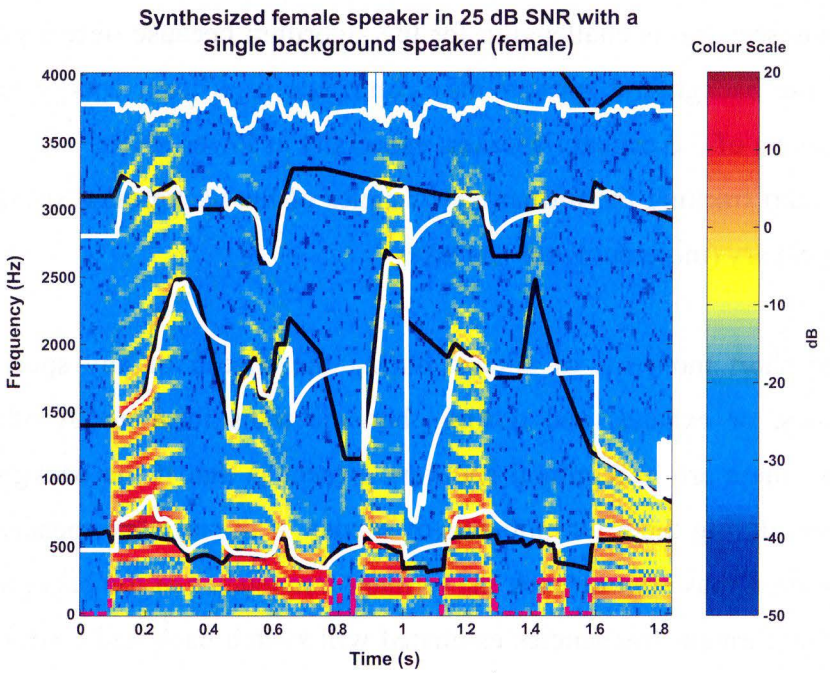

Figure 4 - 9 – Spectrogram of a synthesized female speaker in the presence of female single background speaker at 25 dB SNR

Figure 4-10 shows the spectrogram of a synthesized male speaker saying "Five women played basketball" in the presence of a female single background speaker at a SNR of 30 dB. From the spectrogram it is clear that the algorithm is able to accurately detect the formants frequencies for the male speaker as well. Comparing Figure 4-9 to Figure 4-1 it can be seen that formant frequency estimates for both cases are very similar to each other. Therefore, it can be concluded that at high SNRs the background noise

does not affect the performance of the algorithm. This observation is once again noted when Figures 4-10 and 4-2 are compared.

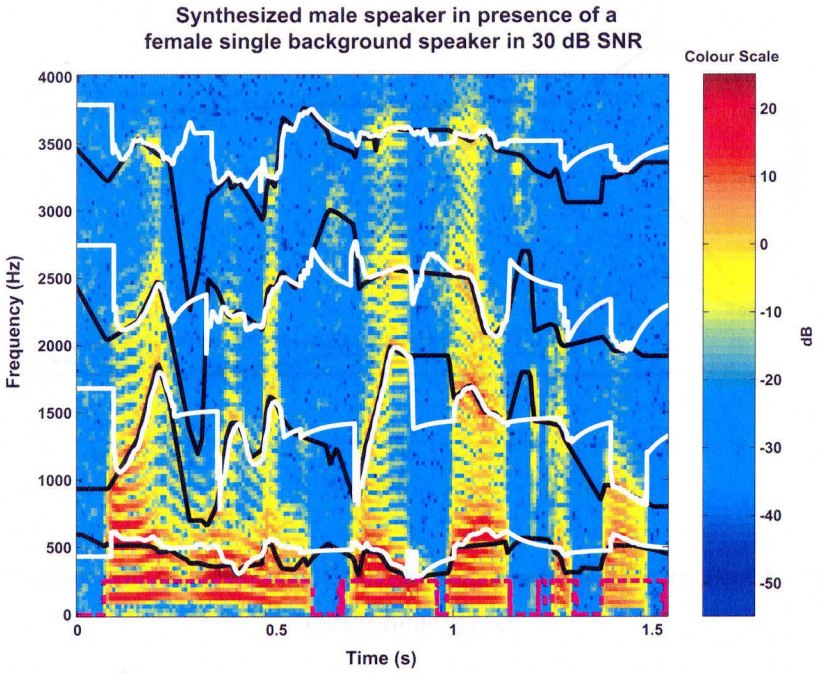

Figure 4 - 10 – Spectrogram of a synthesized male speaker in the presence of female single background speaker at 30 dB SNR

Figure 4-11 shows the variation of the RMSE with the SNR for a female synthesized speaker (saying 'Five women played basketball') in the presence of a female single background speaker (saying 'Don't ask me to carry an oily rag like that'). From this figure it can be seen that the algorithm performs very well even at low SNRs and is able to estimate the first three formant frequencies with a reasonable amount of error for female speakers.

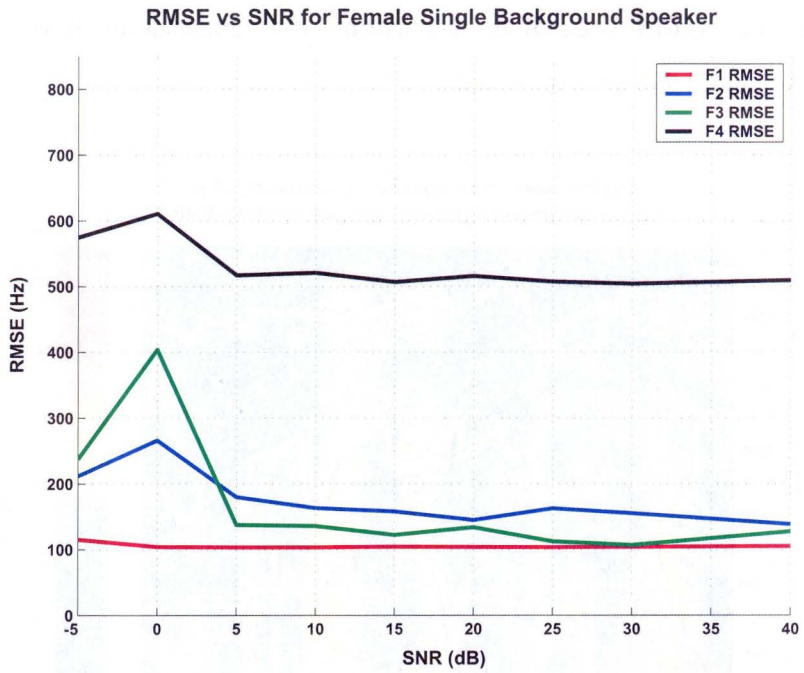

Figure 4 - 11 – RMSE vs. SNR for a synthesized female speaker in the presence of female single background speaker

Figure 4-12 shows the RMSE vs. SNR plot for a synthesized male speaker (saying 'Once upon a midnight dreary') in the presence of a single female background speaker (saying 'Five women played basketball'). The algorithm performs well for the synthesized male speaker as well as, if not better than, it does for the synthesized female speaker.

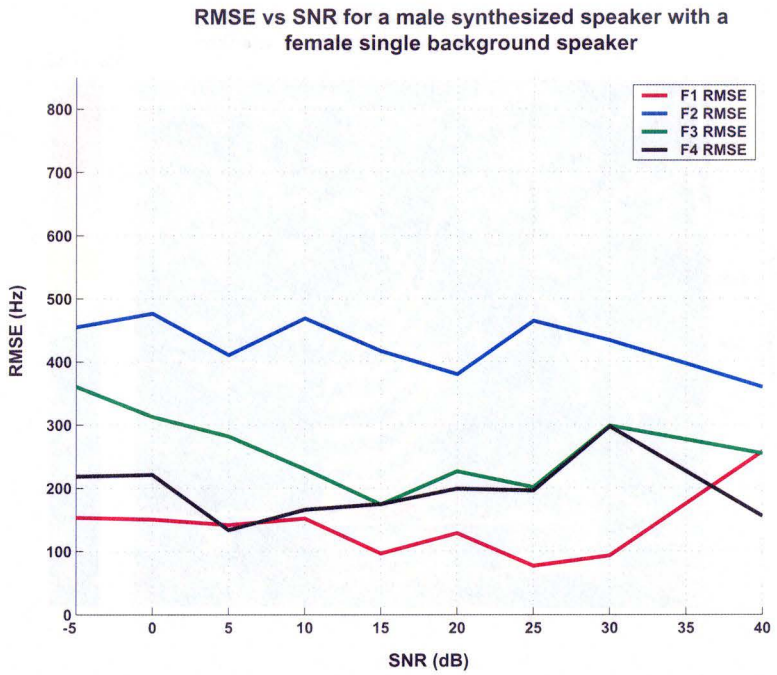

Figure 4 - 12 – RMSE vs. SNR for a synthesized male speaker in the presence of female single background speaker

The algorithm is also tested using more natural sounding speech (for both male and female speakers) from the TIMIT database at various SNRs. Figures 4-13 and 4-14 show portions of the spectrogram for natural female and male speakers in the presence of a female single background speaker at 20 dB and 15 dB SNR respectively. From visual inspection of these spectrograms, it can be seen that the algorithm performs well for both genders despite the relatively low SNRs. The algorithm is also able to track formant frequencies as the speech switches between voiced and unvoiced segments and provides smooth formant frequency estimates. The algorithm was tested for TIMIT database sentences for a wide range of SNRs from 40 dB to -5 dB to ensure that the performance was acceptable for the entire range.

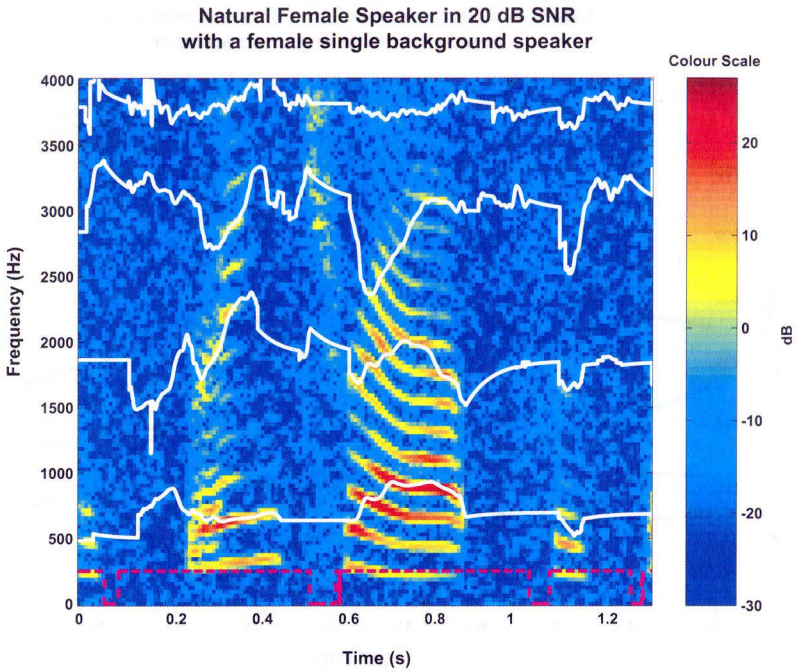

Figure 4 - 13 – Spectrogram of a natural female speaker in the presence of female single background speaker at 20 dB SNR

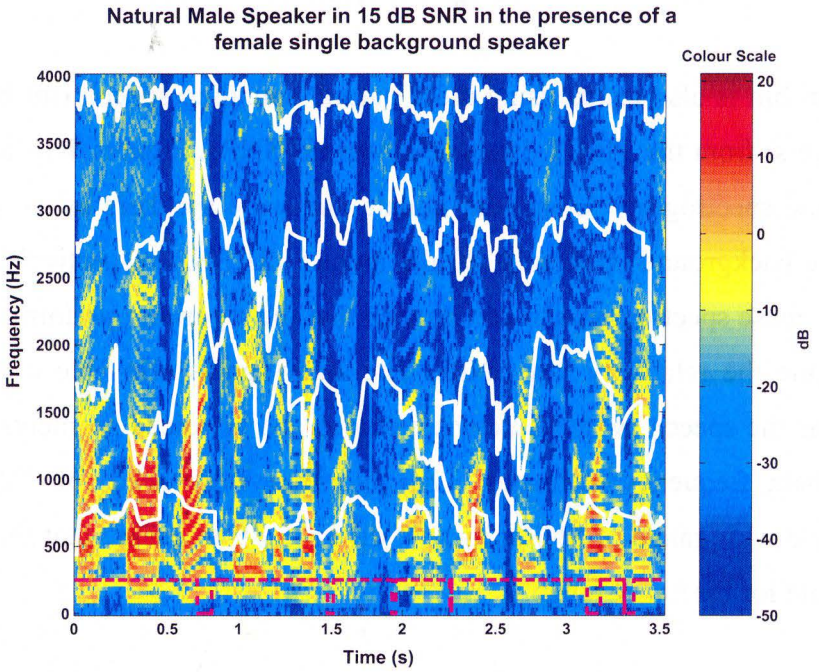

Figure 4 - 14 – Spectrogram of a natural male speaker in the presence of female single background speaker at 15 dB SNR

4.3. Testing in the presence of a male single background speaker

Due to reasons similar to those described for the previous test case, the algorithm was also tested in the presence of a male single background speaker at varying SNRs (from 40 dB to -5 dB). Concerns still remain regarding the algorithm starting to track the formant frequencies of the background speaker instead of the primary speaker at low SNRs. Similar to the female single background speaker case, the estimated formant frequencies can still switch back and forth between those of the primary speaker and the background speaker due to the noise contributions from the background speaker during momentary periods of silence of the primary speaker.

Figures 4-15 and 4-16 show the RMSE vs. SNR plot for a synthesized female and male speaker in the presence of a male single background speaker respectively. The RMSE is low for all the formant frequencies at higher SNRs but increases (as expected) when the SNR drops.

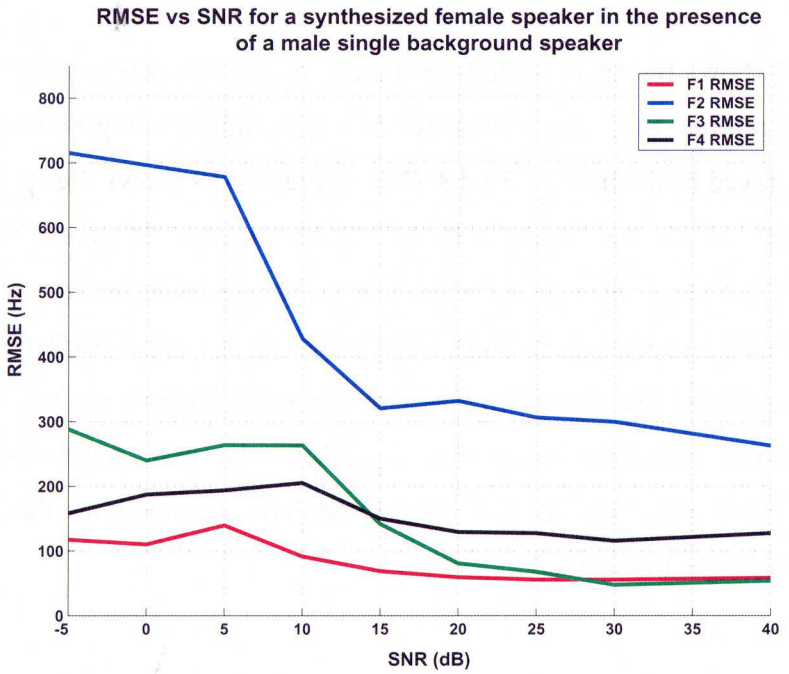

Figure 4 - 15 – RMSE vs. SNR for a synthesized female speaker (saying 'he sees the ball') in the presence of male single background speaker (saying 'Five women played basketball')

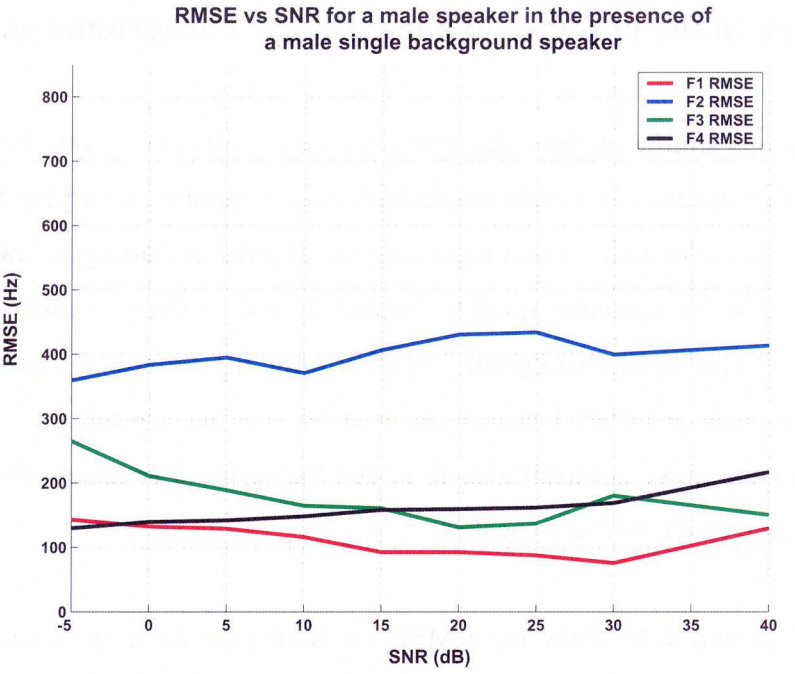

Figure 4 - 16 – RMSE vs. SNR for a synthesized male speaker (saying 'Five women played basketball') in the presence of male single background speaker (saying 'Once upon a midnight')

Figure 4-17 shows the spectrogram for natural female speaker saying "Don't ask me to carry an oily rag like that" in the presence of a male single background speaker saying "Five women played basketball" at 30 dB SNR. Figure 4-18 shows the spectrogram for a natural male speaker saying "It was a fairly modern motel with quite a bit of electrical display in front" in the presence of the same male single background speaker at 25 dB SNR. Analysis of the spectrograms shows that the formant tracker performs well for both genders at these SNRs. The algorithm was also tested for a whole range of SNRs (40 dB to -5 dB) for this test case using various TIMIT database sentences. The overall performance of the algorithm for the TIMIT database sentences in the presence of a single background speaker is good.

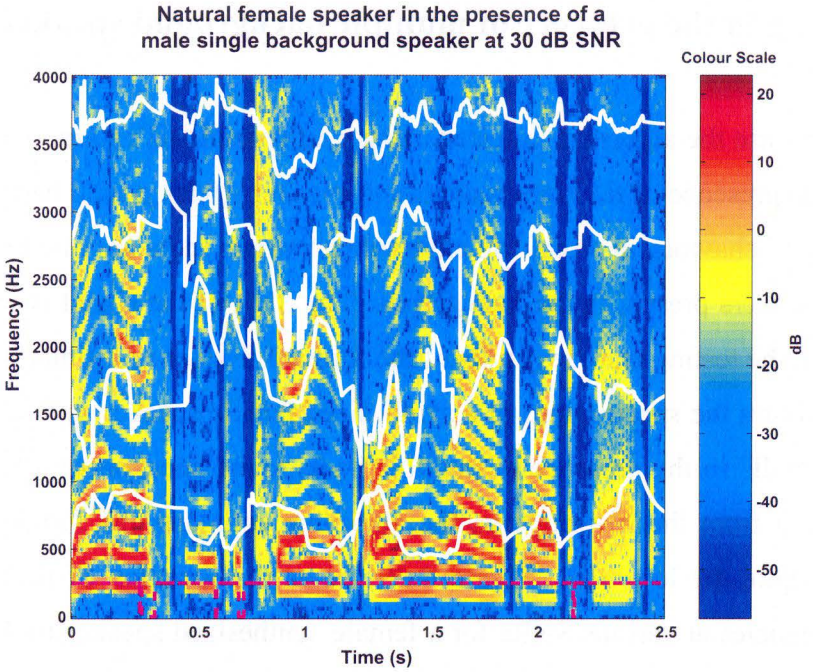

Figure 4 - 17 – Spectrogram of a natural female speaker in the presence of a male single background speaker at 30 dB SNR

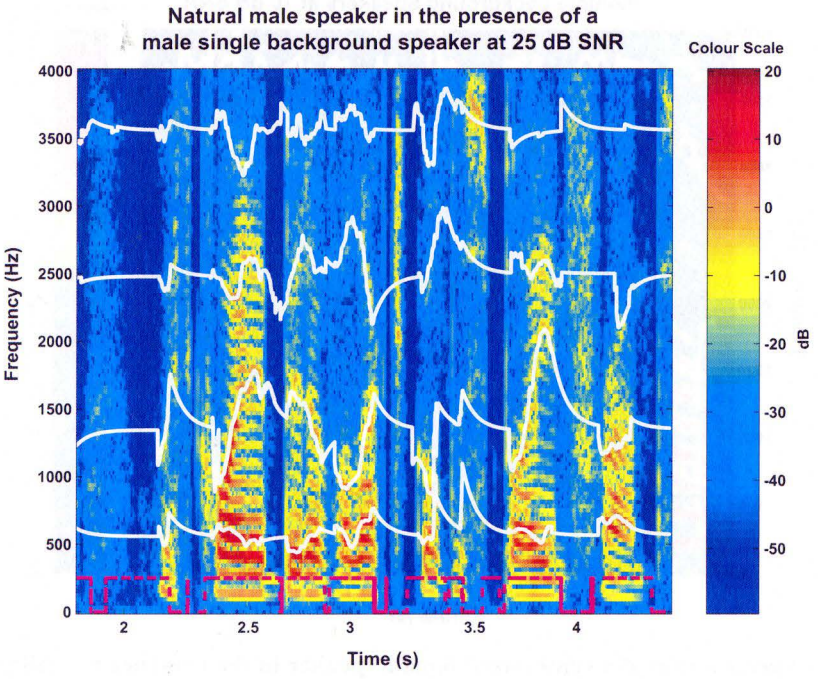

Figure 4 - 18 – Spectrogram of a natural male speaker in the presence of a male single background speaker at 25 dB SNR

4.4. Testing in the presence of multiple background speakers

In this test case the algorithm is tested using synthesized and natural male and female speakers in the presence of multiple background speakers (background babble) to analyse the algorithm's behaviour in a real-life environment where there are often more than just one or two speakers present in the background. The SNR of the signal is varied from 40 dB to -5 dB in the testing and the results of the formant frequency estimates are analysed. Figure 4-19 shows the spectrogram of a synthesized female speaker saying "Five women played basketball" in the presence of multiple background speakers at a SNR of 10 dB. As can be seen from the spectrogram, the algorithm estimates the formant frequencies quite well despite the low SNR. Figure 4-20 shows the variation of the RMSE of the formant frequencies at various SNRs for a female synthesized speaker in the presence of multiple background speakers.

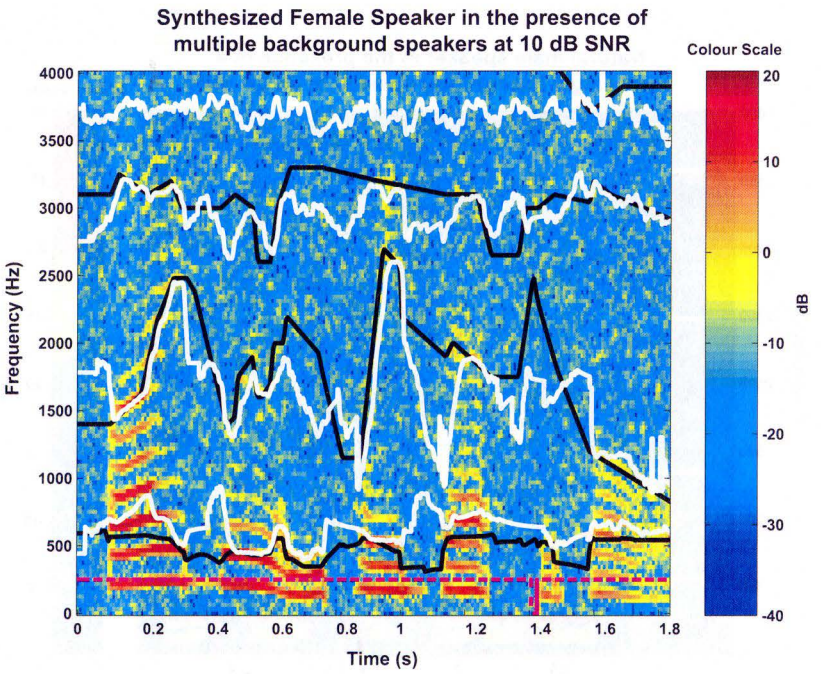

Figure 4 - 19 – Spectrogram of a synthesized female speaker in the presence of multiple background speakers at 10 dB SNR

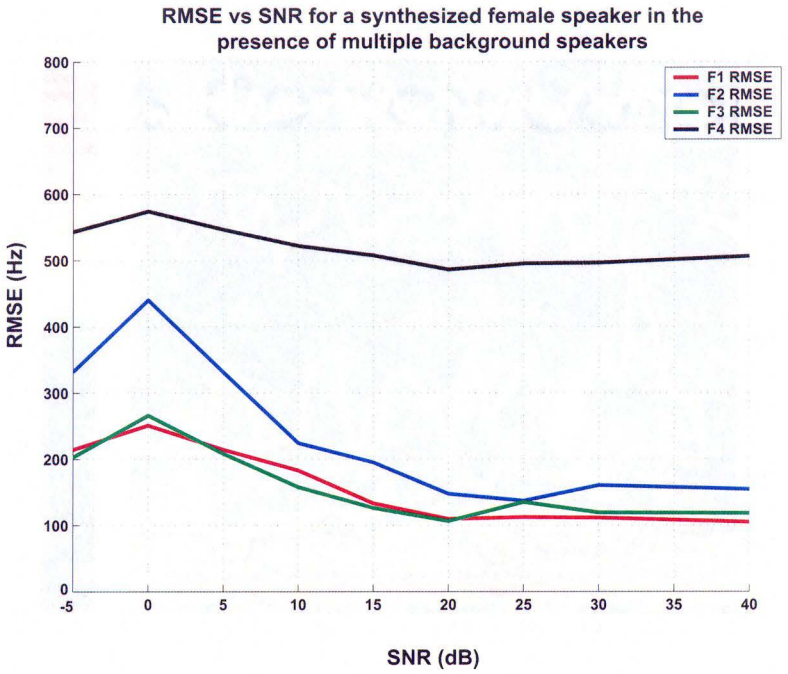

Figure 4 - 20 – RMSE vs. SNR for a synthesized female speaker in the presence of multiple background speakers

From the plot it is clear that the algorithm performs well in the presence of multiple background speakers until about 10 dB SNR after which the error rises sharply. This occurs, because below 10 dB SNR the energy from the background speakers causes the algorithm to start tracking the background speakers during the moments of silences of the primary speaker. This leads to the estimated formant frequencies (especially the second formant) to switch between those of the primary and the background speakers and leads to high RMSE. These findings are illustrated in Figure 4-21 which shows the spectrogram of the synthesized female speaker saying "Five women played basketball" in the presence of multiple background speakers at a SNR of 0 dB. Figure 4-22 shows the variation of the RMSE of the formant frequencies at various SNRs for a male synthesized speaker in the presence of multiple background speakers. It shows similar behaviour to that of the female synthesized speaker.

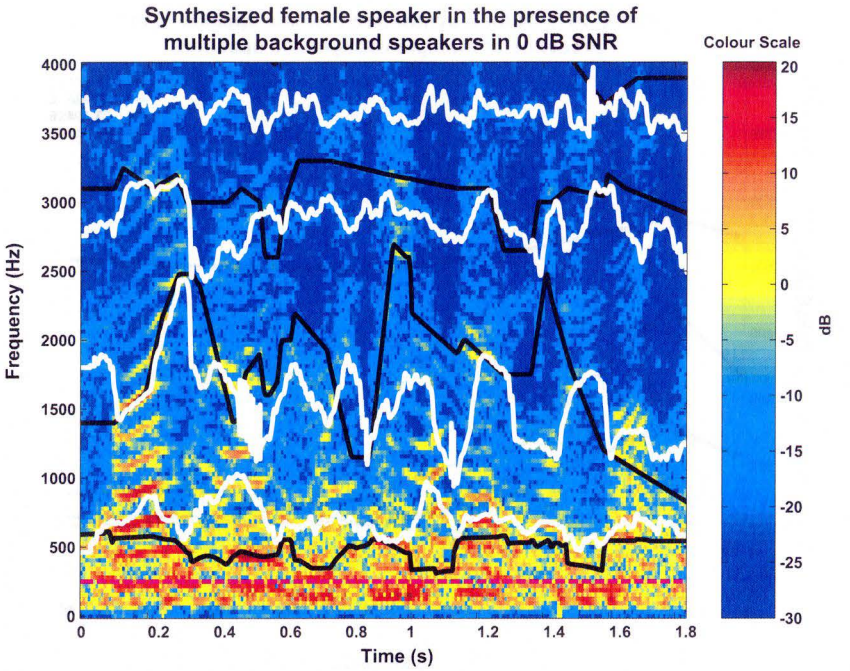

Figure 4 - 21 – Spectrogram of a synthesized female speaker in the presence of multiple background speakers at 0 dB SNR

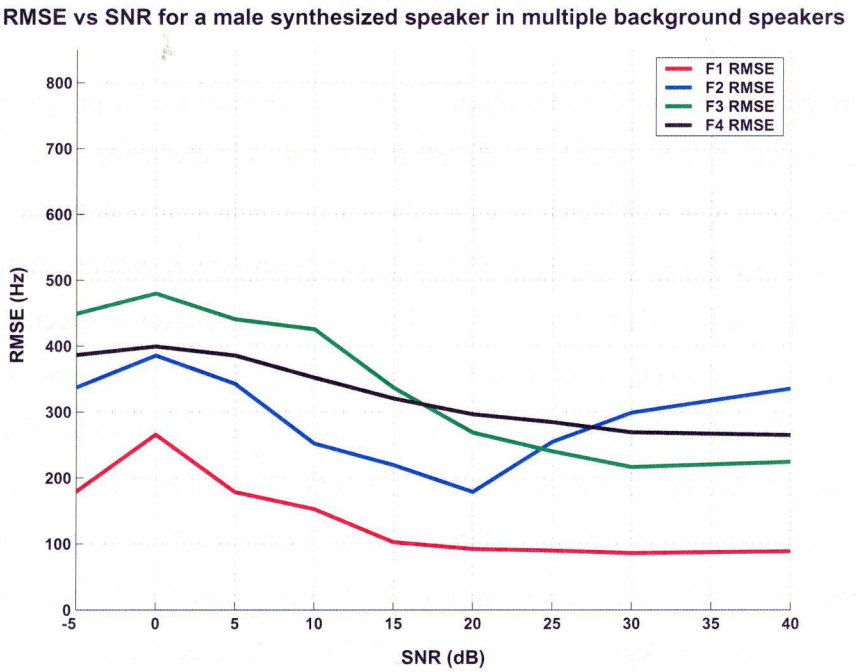

Figure 4 - 22 – RMSE vs. SNR for a synthesized male speaker in the presence of multiple background speakers

The algorithm was also tested for this test case using natural speech from the TIMIT database for both male and female speakers in a SNR range of 40 dB to -5 dB. The performance was similar to that in synthesized speech and the algorithm was able to track formant well until about 5 dB SNR, after which the estimated and visually observed formant frequencies seemed to diverge. Figure 4-23 and 4-24 show spectrograms of female and male speakers from the TIMIT database at a SNR of 15 and 10 dB respectively in the presence of multiple background speakers. The spectrograms show the formant tracker is able to estimate the first and second formant frequencies reasonably accurately for both the female and male speakers in these SNR levels. The third formant frequency shows some deviation from the actual formant frequency of the speakers at certain times, but is able to recover quickly and track the actual formant for most of the duration. Figure 4-25 shows the spectrogram for the male speaker at 5 dB SNR and it is clear that the algorithm is not able to accurately track the formant frequencies of the primary speaker the voicing detector also seems to have some problems correctly detecting voicing at this low SNR.

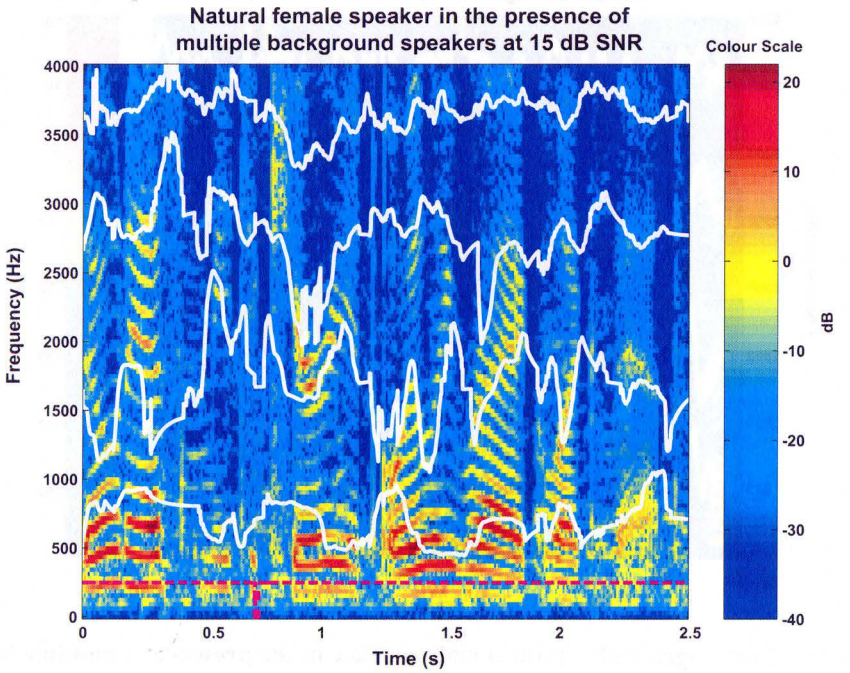

Figure 4 - 23 – Spectrogram of a natural female speaker in the presence of multiple background speakers at 15 dB SNR

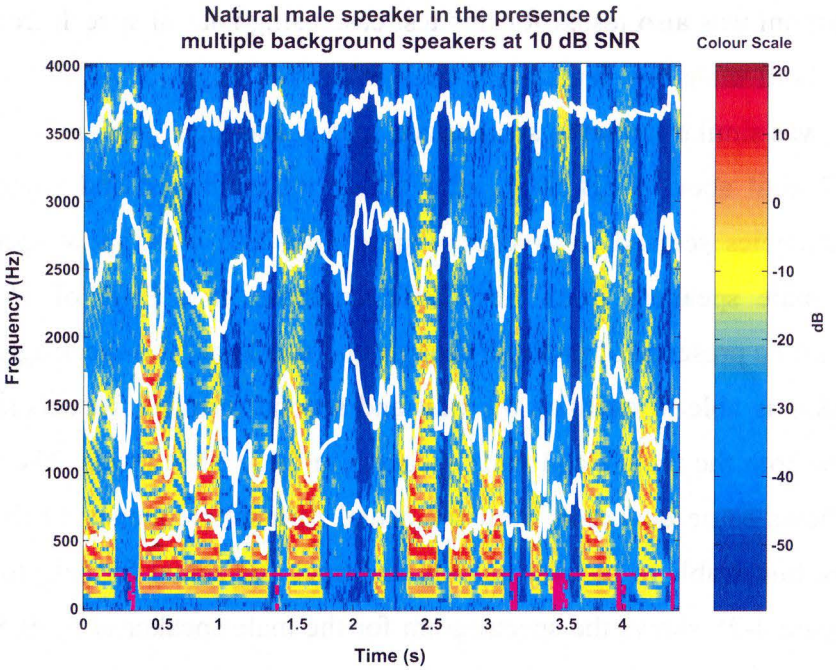

Figure 4 - 24 – Spectrogram of a natural male speaker in the presence of multiple background speakers at 10 dB SNR

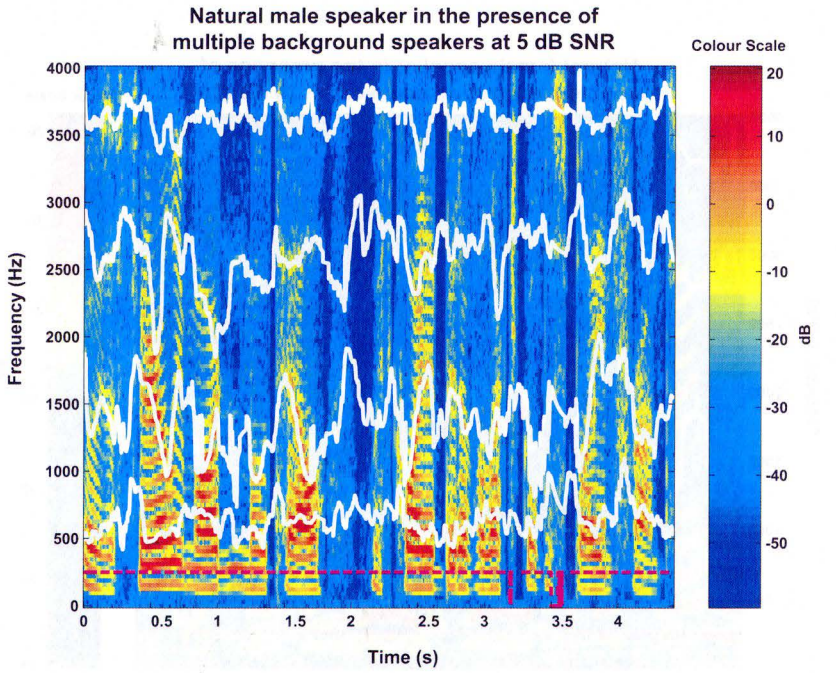

Figure 4 - 25 – Spectrogram of a natural male speaker in the presence of multiple background speakers at 5 dB SNR

4.5. Testing in the presence of background music

In real-life a speaker can be talking in the presence of background music from a variety of sources and the formant tracking algorithm has to be able to accurately estimate formant frequencies in the presence of this noise source. In this test case the algorithm is tested in the presence of background music for a range of SNRs from 40 dB to -5 dB. Musical instruments have particular spectral envelopes that give them their distinct sound and serve the same purpose as the vocal tract in the auditory system. The spectral envelope of an instrument may have an adverse effect on the performance of the formant tracking algorithm. Figure 4-26 shows a spectrogram of a female synthesized speaker saying "He sees the ball" in the presence of background music at a SNR of 40 dB. The formant frequencies are estimated accurately for most of the signal except for the second formant frequency that oscillates briefly between t = 0.7 sec and t = 0.8 sec. The oscillations occur due of the lack of energy in the second formant frequency region from the primary speaker leading to the algorithm picking energy peaks from the background music as the second formant frequency.

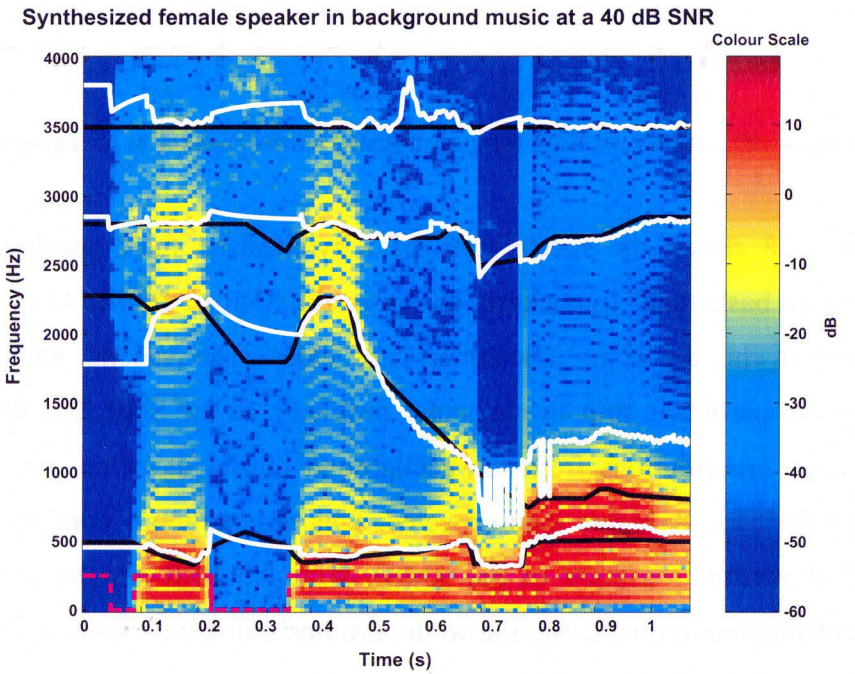

Figure 4 - 26 – Spectrogram of a synthesized female speaker in background music at 40 dB SNR

Figure 4-27 shows the variation of the RMSE with the SNR for a synthesized female speaker in the presence of background music. Trends similar to those observed in earlier test cases can be seen once again in this figure. The RMSE rises as the SNR decreases until about 0 dB when the algorithm adapts and starts using the moving average value of the formant frequencies instead of spectral estimates, leading to a drop in the RMSE.

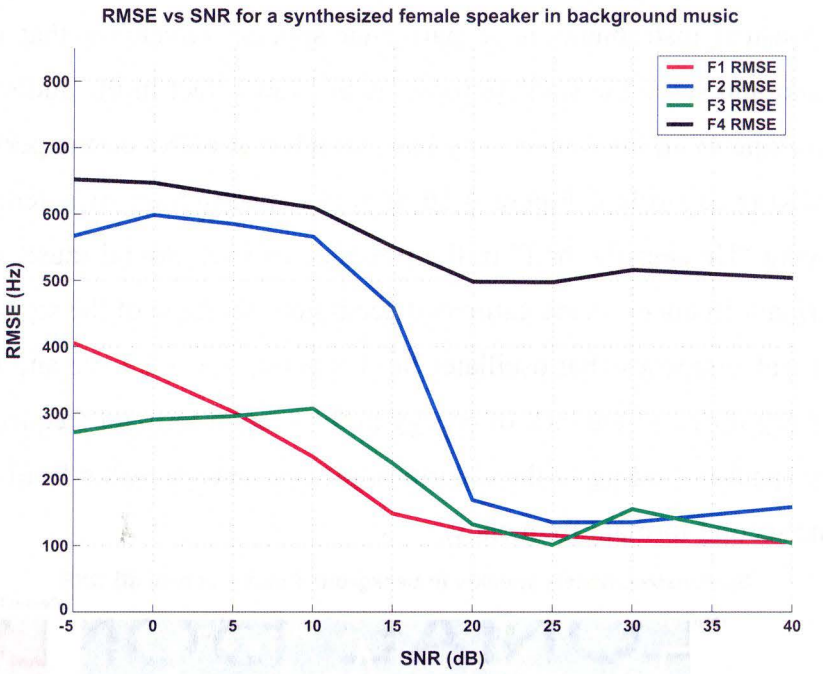

Figure 4 - 27 – RMSE vs. SNR for a synthesized female speaker in background music

Figure 4-28 shows the spectrogram of a synthesized male speaker saying "Five women played basketball" in background music at a SNR of 10 dB. At this low SNR the problems that the algorithm encounters are clear. The second and third formant frequencies in particular start to encounter problems as the SNR degrades. Due to the excess energy contributions from the background music into these formant frequency bands their RMSEs are higher than expected. This observation is confirmed in Figure 4-29 which shows the variation of the RMSE with the SNR for the synthesized male speaker in background music. Similar to the synthesized female speaker case, the overall RMSE rises with the decreases in SNR.

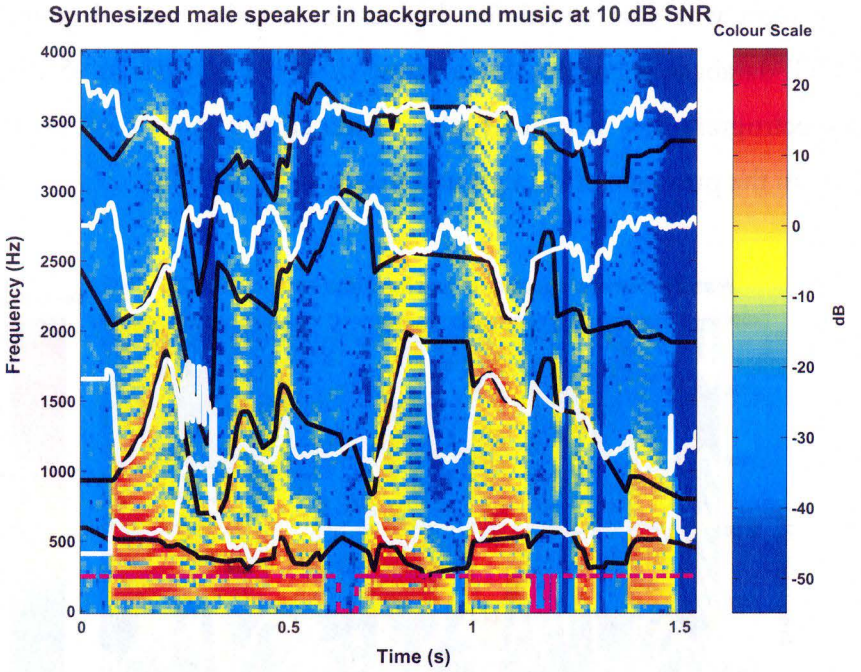

Figure 4 - 28 – Spectrogram of a synthesized male speaker in background music at 10 dB SNR

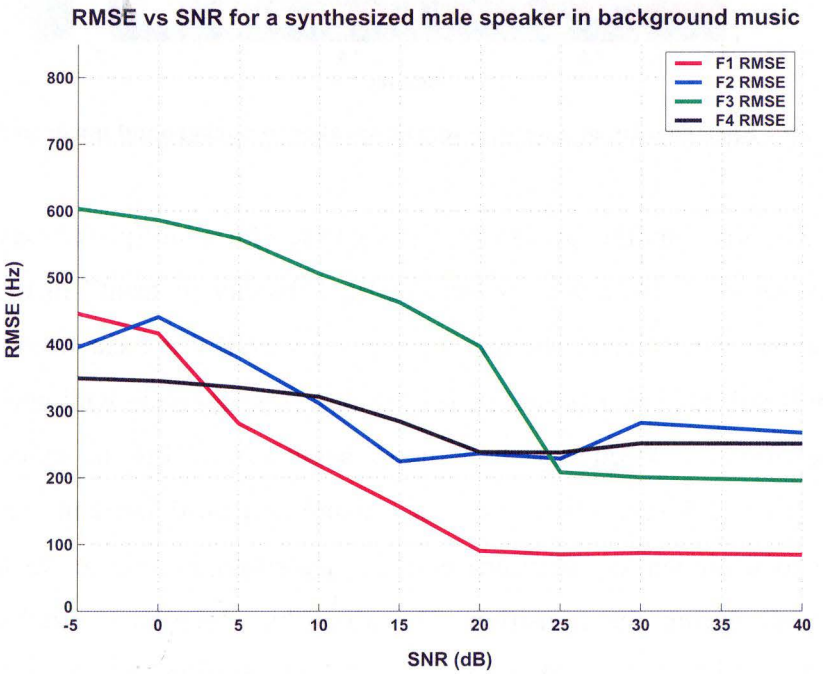

Figure 4 - 29 – RMSE vs. SNR for a synthesized male speaker in background music

The algorithm was also tested for this test case using natural sounding male and female TIMIT database sentences for a range of SNRs (40 dB to -5 dB). Figure 4-30 shows the spectrogram of a natural female speaker saying "Don't ask me to carry an oily rag like that" in the presence of background music at 30 dB SNR.

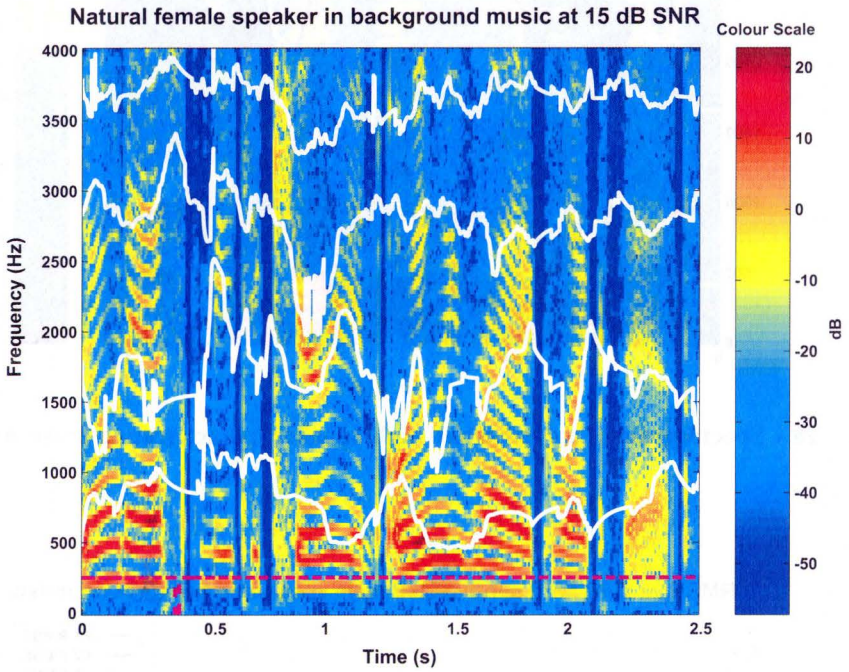

Figure 4 - 30 – Spectrogram of a natural female speaker in background music at 15 dB SNR

Figure 4-31 shows another spectrogram, this time of a male speaker saying "It was a fairly modern motel with quite a bit of electrical display in front" in the presence of background music at 0 dB SNR. Despite the very low SNR it is clear from the figure that the algorithm is still able to pick up formant frequencies relatively well and it is also able to track the formant frequency transitions accurately. This is evident from the spectrogram at t = 2.5 sec. where the first, second and third formant frequencies shift sharply (due to a phoneme transition) and the algorithm is able to track the formant frequencies well during the transition. The algorithm does have problems when the primary speaker is silent (t = 3.5 sec.). At t= 3.5 sec. the formant frequency estimates

oscillate because the algorithms starts to track the formant frequencies of the background music instead of those of the speaker.

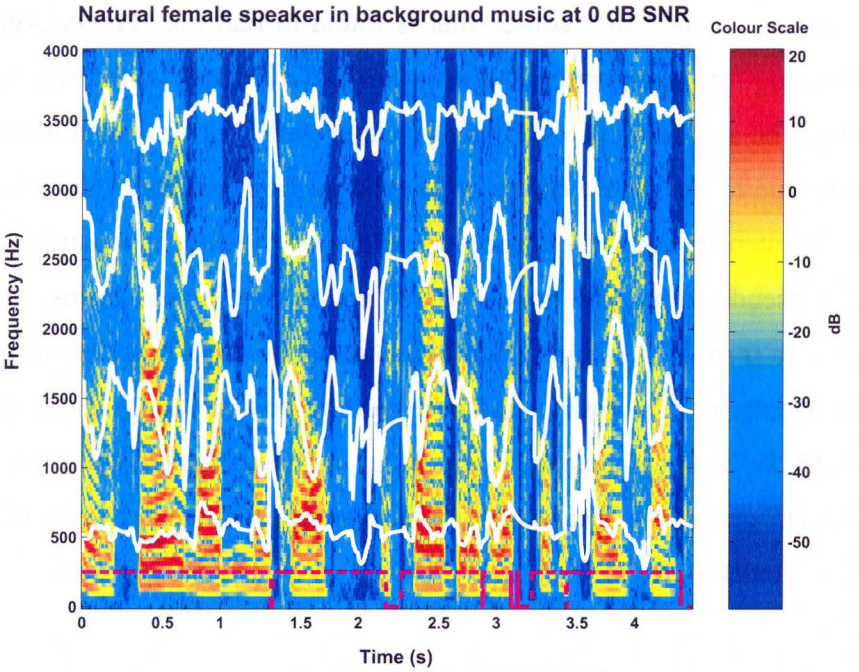

Figure 4 - 31 – Spectrogram of a natural male speaker in background music at 0 dB SNR

4.6. Testing in the presence of background traffic noise

This test case is similar to the previous one where the algorithm is tested in the presence of a background noise source that is found in real-life. The background traffic case is challenging for the algorithm because the type of noise that the algorithm is tested in is changing. The algorithm is tested using 'heavy traffic' noise which is usually similar to white-noise due to the nature of the noise emitted from passing vehicles. However, it can change in intensity as the distances of the cars passing by changes and is often coupled with pseudo-impulsive noises (burst noise) due to car horns, etc. All these sources couple to form a challenging and dynamic background noise environment for the algorithm to operate in. The testing has been performed for a wide range of SNRs from 40 dB to -5 dB for both synthesized and natural male and female speakers. Figure 4-32 shows the spectrogram of a synthesized female speaker in the presence of background traffic noise in 20 dB SNR.

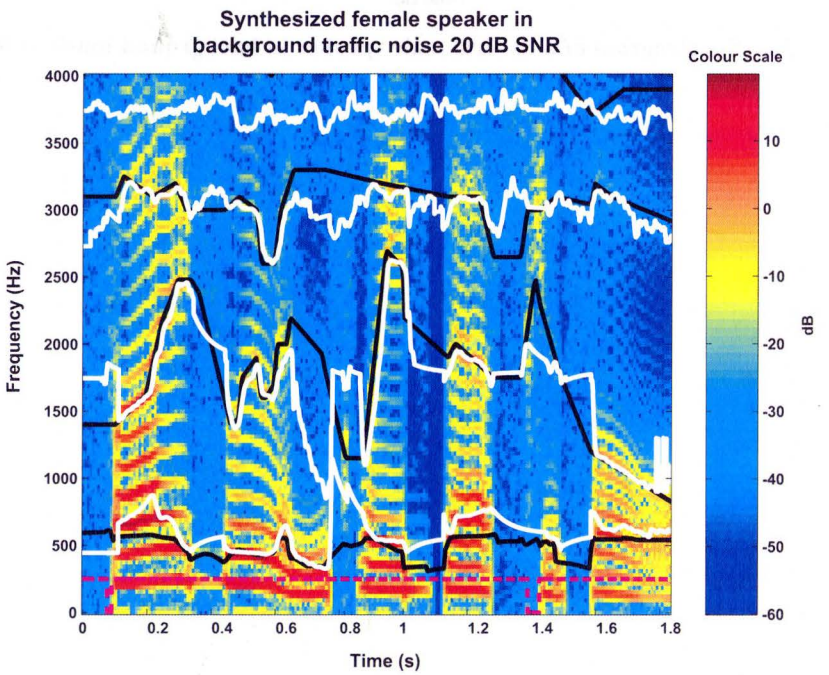

Figure 4 - 32 – Spectrogram of a synthesized male speaker in background music at 20 dB SNR

The main problem associated with background traffic noise is illustrated in this figure where at t = 0.75 sec the second formant frequency shifts suddenly due a burst of noise from the background (e.g., car horn). However, the algorithm recovers quickly and is able to go back to tracking the proper formant frequencies soon afterwards. Figure 4-33 shows the variation of the RMSE with the SNR for the synthesized female speaker saying "Five women played basketball" in background traffic noise. This plot shows that the formant frequencies all have relatively low RMSEs (except for the fourth formant frequencies) even at very low and negative SNRs. This result conforms to expectations that the traffic noise is similar to white noise and therefore the algorithm should have low RMSEs even at low SNRs. The effect of burst noise present in traffic has localized effects on the ability of the algorithm to track formant frequencies and therefore is not a major source of error.

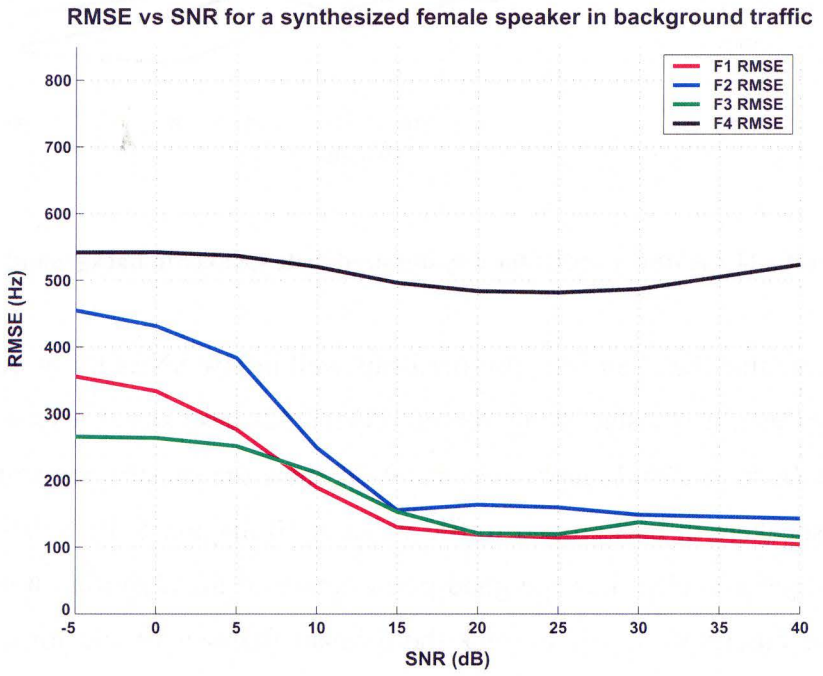

Figure 4 - 33 – RMSE vs. SNR for a synthesized female speaker in background traffic

Figure 4-34 shows the variation of the RMSE with the SNR for the synthesized male speaker saying "Once upon a midnight dreary" in background traffic noise. The RMSE of the male speakers is also relatively lower as in the synthesized female speaker case.

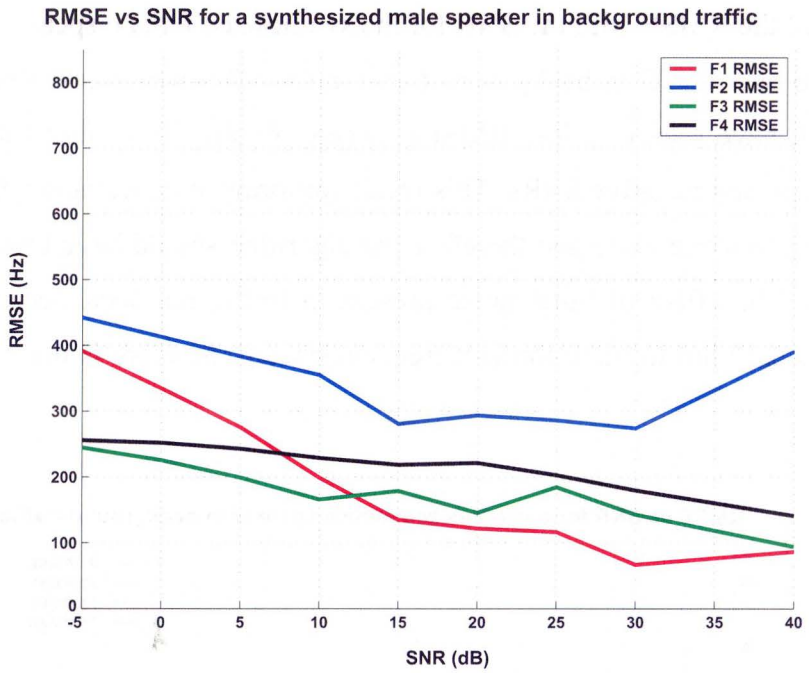

Figure 4 - 34 – RMSE vs. SNR for a synthesized male speaker in background traffic

To illustrate that the algorithm also performs well in low SNRs for more natural male and female speech, it is tested using various TIMIT database sentences for a wide range of SNRs (40 dB to -5 dB). Figures 4-35 and 4-36 show the spectrograms of natural male and female speakers in background traffic noise at 10 dB and 5 dB SNRs respectively. From the figures it is clear that the good performance of the algorithm holds for natural speakers. The algorithm is able to track the formant frequencies and formant frequency transitions well.

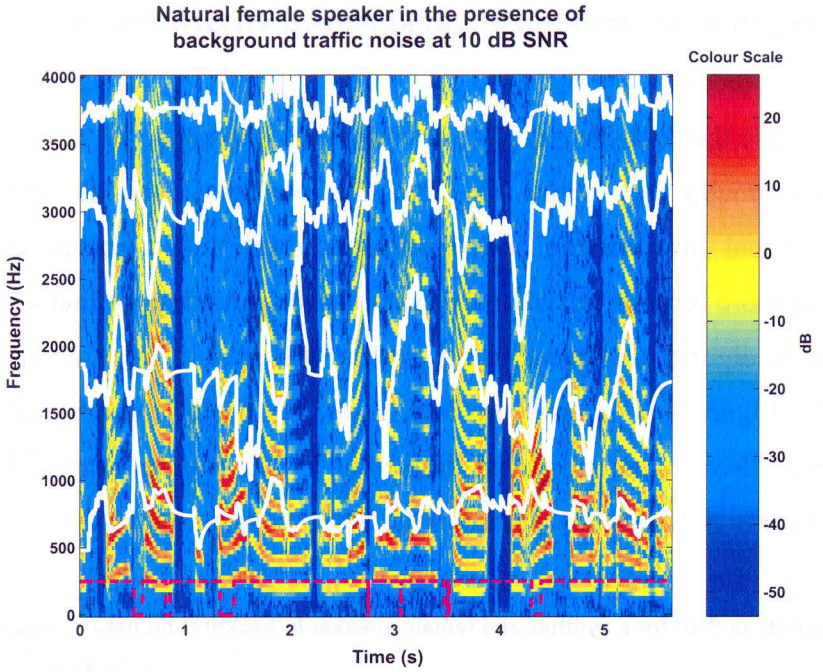

Figure 4 - 35 – Spectrogram of a natural female speaker in background music at 10 dB SNR

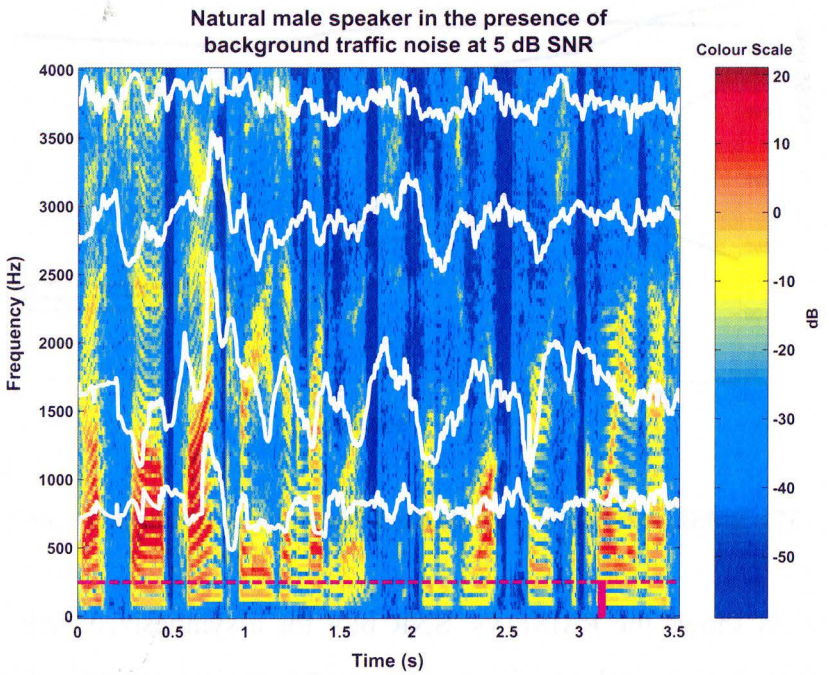

Figure 4 - 36 – Spectrogram of a natural male speaker in background music at 5 dB SNR

4.7. Testing in a natural noise background environment

In this test case the background noise environment is made up of 'natural' sounds that a speaker is normally present in. This includes biological sounds from insects and animals, occasional low volume conversations, distant traffic noises, etc. Since this environment is more representative of the type of background noise that a speaker would be present in, the algorithm is tested using it as the background noise at various SNRs (40 dB to -5 dB). Figure 4-37 shows the RMSE vs. SNR plot for a synthesized female speaker saying "Five women played basketball". The figure shows that RMSE is relatively low for high SNRs but rises after the SNR drops below 15 dB.

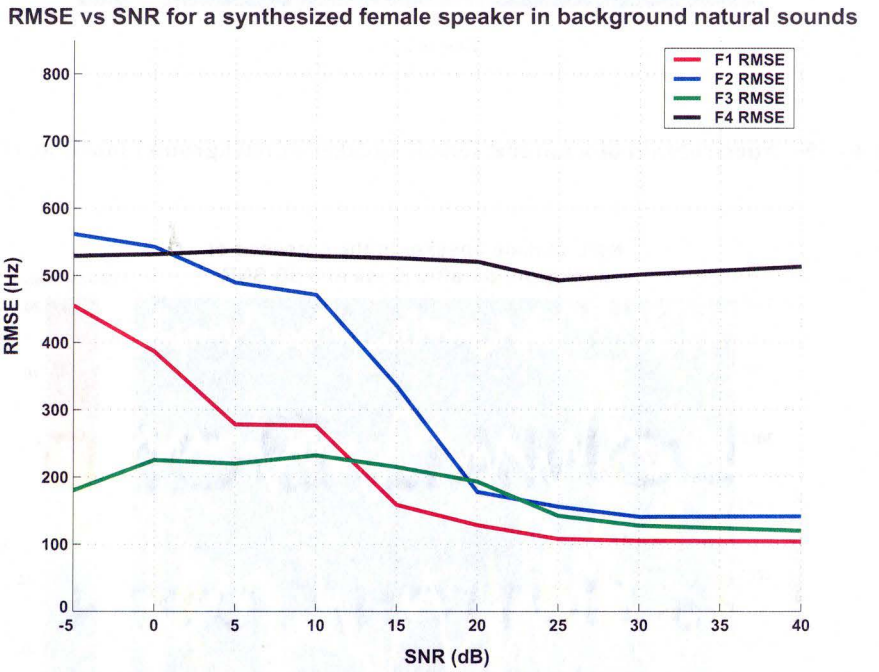

Figure 4 - 37 – RMSE vs. SNR for a synthesized female speaker in background natural sounds

Figure 4-38 shows the RMSE vs. SNR plot for a synthesized male speaker saying "Five women played basketball" in background natural sounds. The same trends observed for synthesized female speaker are seen for the synthesized male speaker.

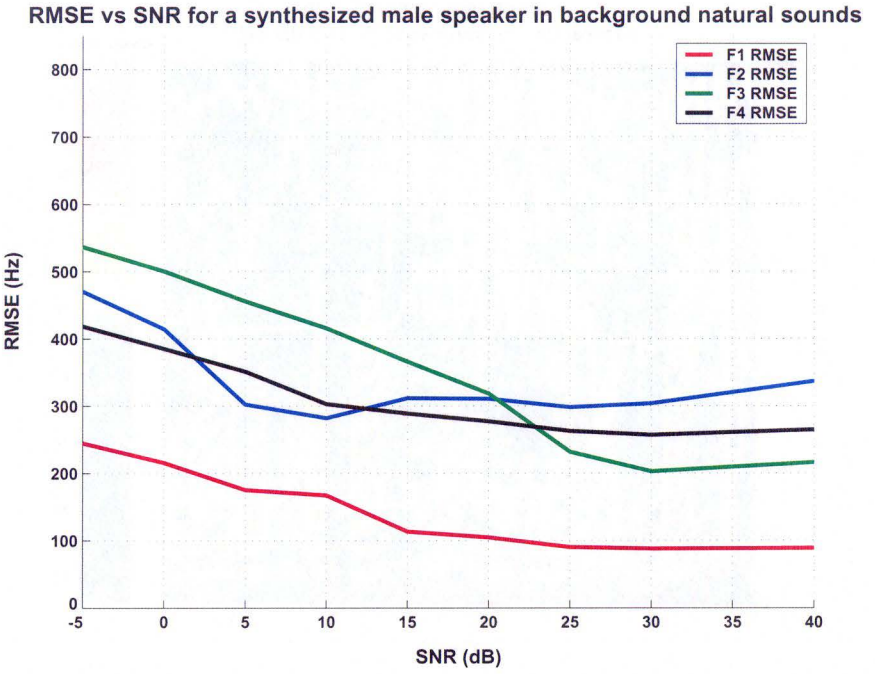

Figure 4 - 38 – RMSE vs. SNR for a synthesized male speaker in background natural sounds

The algorithm is also tested using natural speech signals from the TIMIT database for both male and female speakers. Figure 4-39 shows the spectrogram of a female speaker saying "A spring trap for solid mounting and a regular hand trap are also available" in the presence of background natural sounds at a 5 dB SNR. Figure 4-40 shows the spectrogram of a male speaker saying "Gus saw pine trees and redwoods on his walk through Sequoia national forest" in the presence of background natural sounds at a 0 dB SNR. From both these figures it is clear that, despite the low SNRs the algorithm performs acceptably in tracking the formant frequencies and providing smooth estimates during voiced and unvoiced speech for both male and female speakers.

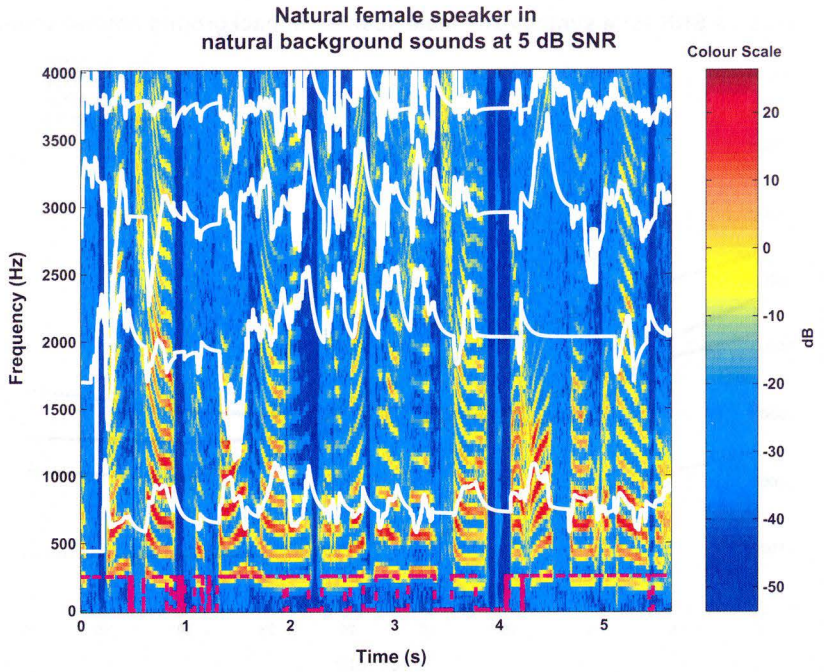

Figure 4 - 39 – Spectrogram of a natural female speaker in background natural sounds at 5 dB SNR

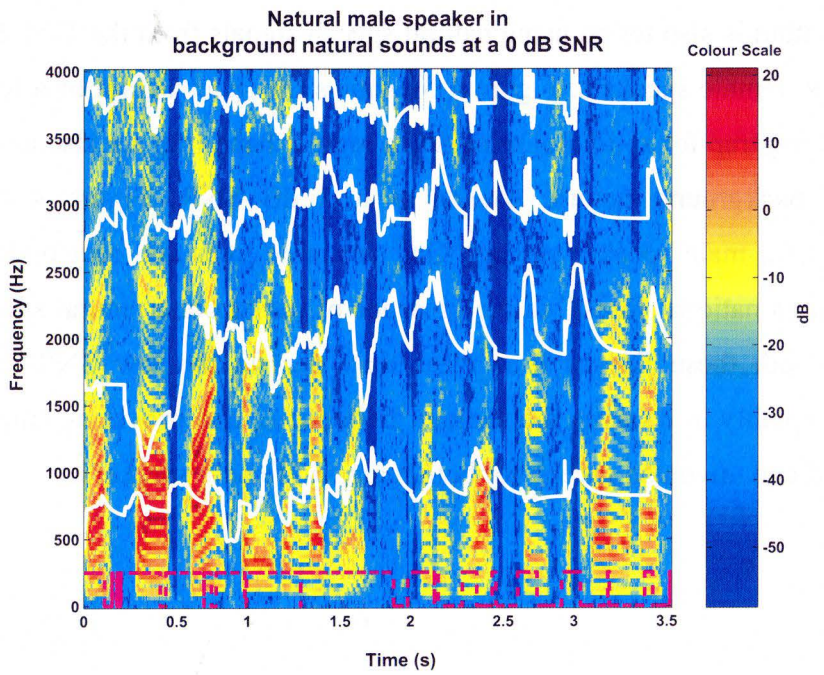

Figure 4 - 40 – Spectrogram of a natural male speaker in background natural sounds at 0 dB SNR

4.8. Testing the algorithm for fading speech

In this test case the algorithm is tested to observe the effect of a speaker whose speech is fading 'in and out' on the algorithm's ability to track formant frequencies. The fading effect of speech is simulated by amplitude modulating the signal using a low-frequency sinusoid. Figure 4-41 shows the variation of the RMSE of the formant frequencies with the frequency of modulation of the speech for a synthesized female speaker saying "Five women played basketball". It is clear from the spectrogram that the amplitude modulation simulating fading of the volume of the speaker has little or no effect on the performance of the algorithm.

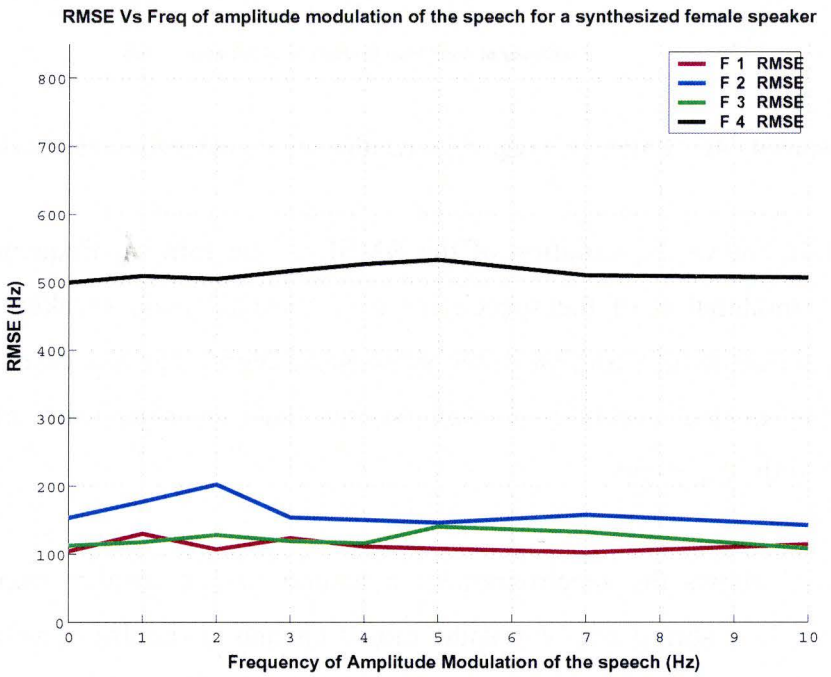

Figure 4 - 41 – RMSE vs. Freq. of modulation for a synthesized female speaker

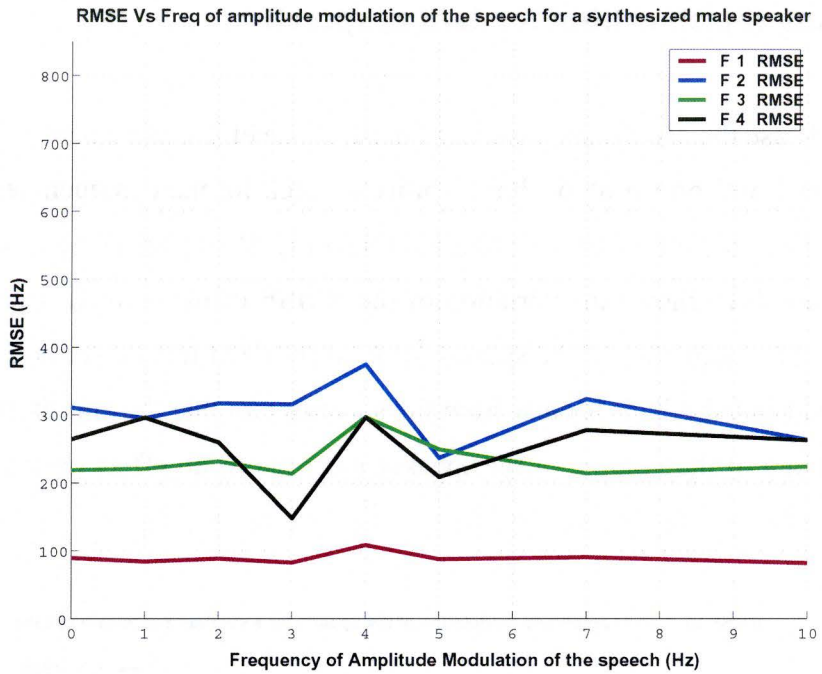

Figure 4 - 42 – RMSE vs. Freq. of modulation for a synthesized male speaker

Figure 4-42 shows the variation of the RMSE of the formant frequencies with the frequency of modulation of the speech for a synthesized male speaker saying "Five women played basketball". Similar to the synthesized female speaker case the frequency of modulation of the synthesized male speech has no significant effect on the performance of the algorithm.

Figure 4-43 shows the spectrogram of a natural female speaker from the TIMIT database saying "a spring trap for solid mounting and a regular hand trap are also available" while the amplitude of the signal is modulated at 5 Hz. From the figure it can be seen that the amplitude modulation has no effect on the formant tracking as in the synthesized speaker cases. Figure 4-44 shows the spectrogram of a natural male speaker saying "it was a fairly modern motel with quite a bit of electrical display in front" while the amplitude of the signal is modulated at 10 Hz. The natural male speaker shows similar trends as the other speakers despite the higher frequency of modulation.

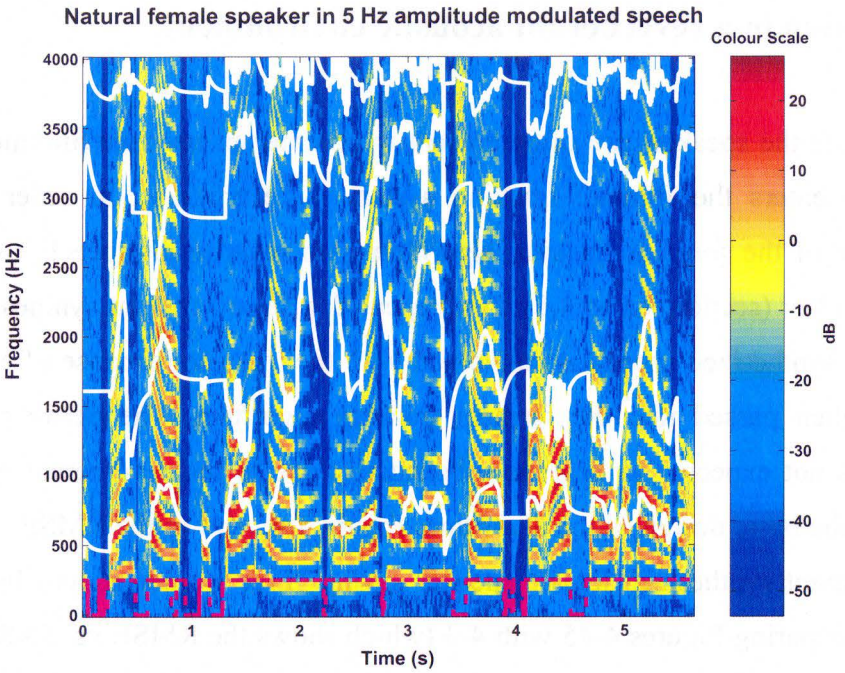

Figure 4 - 43 – Spectrogram of a natural female speaker in 5 Hz amplitude modulation of speech

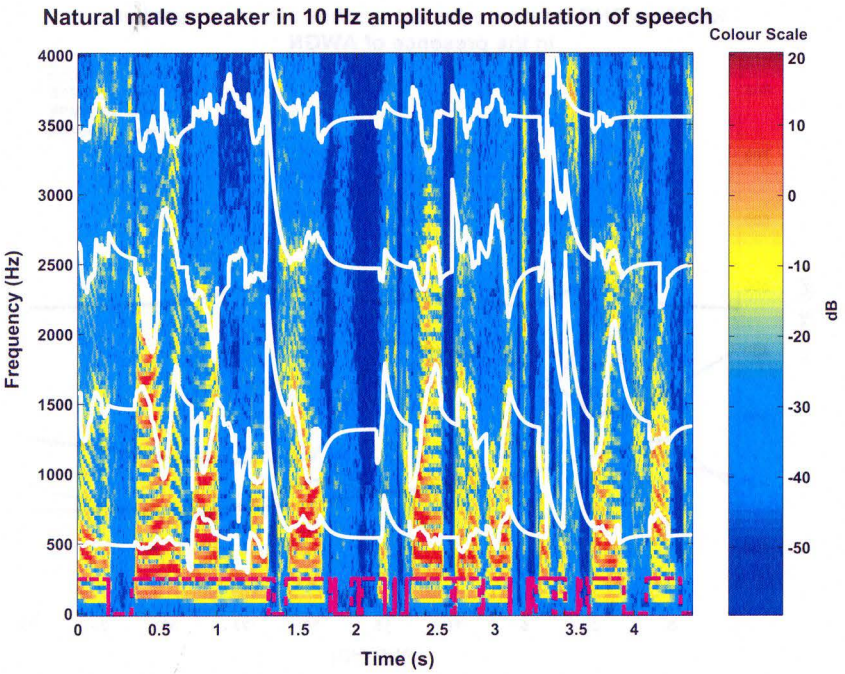

Figure 4 - 44 – Spectrogram of a natural male speaker in 10 Hz amplitude modulation of speech

4.9. Testing in a reverberant acoustic environment

In real-life the speakers are often present in reverberant noise-environments and the speech that enters the formant tracking system is reverberant. In order to test the performance of the formant tracking algorithm with reverberant speech, the test cases described earlier (section 4.1 to 4.8) were repeated with reverberant synthesized speech. The 'clean' synthesized speech is convolved with the impulse response of a reverberant room and then passed into the formant tracking algorithm. The performance of the algorithm is not expected to degrade substantially due to the reverberant environment, because of the basic design of the algorithm. Figure 4-45 shows the RMSE vs. SNR plot of a reverberant synthesized female speaker saying "Five women played basketball" in AWGN. Comparing Figures 4-45 with 4-3 (which shows the RMSE vs. SNR plot for the same speech signal but without reverberance), it is clear that reverberance does not have a significant effect on the performance of the algorithm in AWGN even at low SNRs.

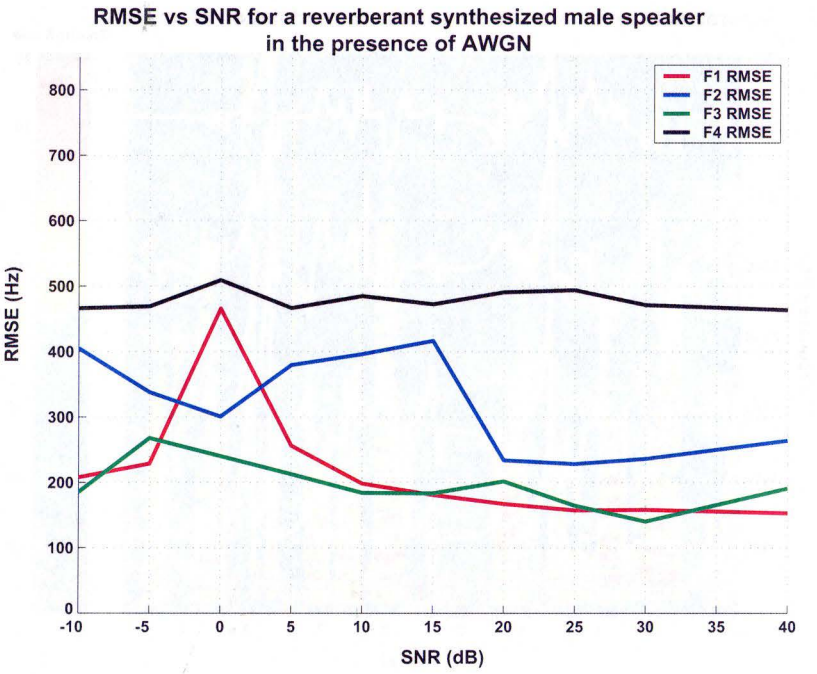

Figure 4 - 45 – RMSE vs. SNR of a reverberant female synthesized speaker in AWGN

The performance of the algorithm was also evaluated for reverberant synthesized male and female speakers in the presence of single and multiple background speakers. Figure 4-46 shows the RMSE vs. SNR plot of a reverberant synthesized male speaker saying "Five women played basketball" in the presence of multiple background speakers. This figure is compared to Figure 4-22, which shows the RMSE vs. SNR plot for the same speech signal but without any reverberance. From the comparison it is clear that reverberance has little effect on the performance of the algorithm in multiple background speakers.

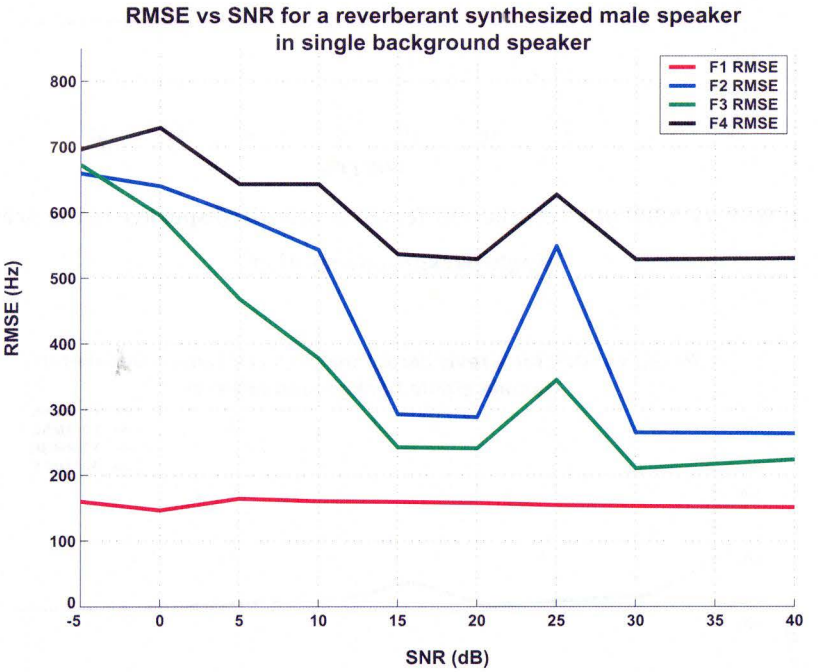

Figure 4 - 46 – RMSE vs. SNR of a reverberant male synthesized speaker in the presence of multiple background speakers

Figures 4-47 and 4-48 show the RMSE vs. SNR plot of a reverberant synthesized female speaker in the presence of a male single background speaker and a female single background speaker, respectively. In both cases the performance of the algorithm does not degrade significantly due to the reverberant environment.

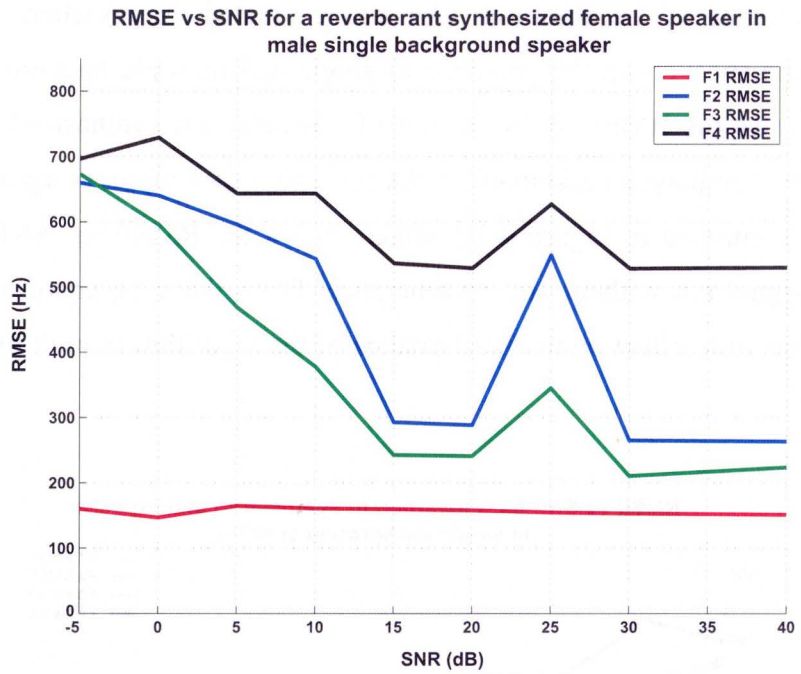

Figure 4 - 47 – RMSE vs. SNR of a reverberant female synthesized speaker in the presence of a male single background speaker

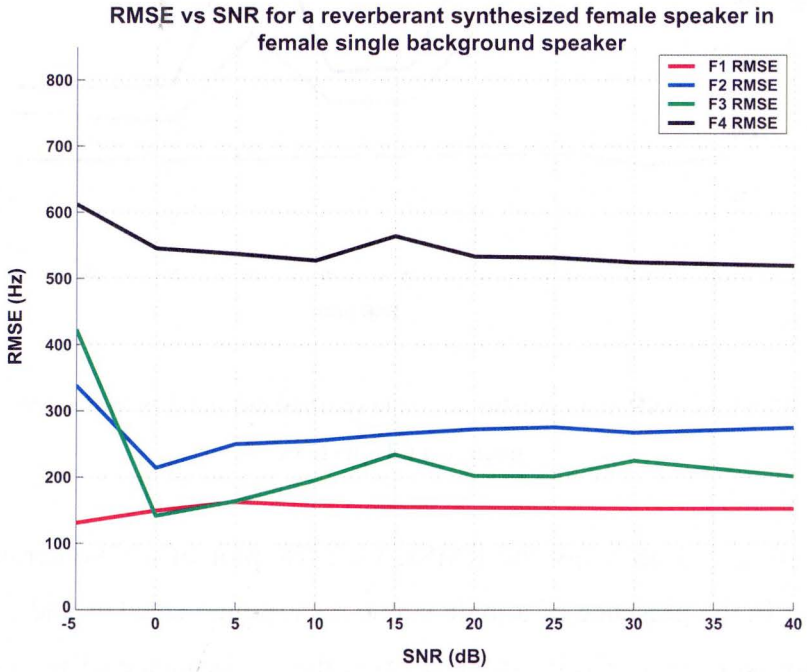

Figure 4 - 48 – RMSE vs. SNR of a reverberant female synthesized speaker in the presence of a female single background speaker

5. COMPARISON OF TRADITIONAL FORMANT ESTIMATION TECHNIQUES

Numerous signal processing techniques for formant frequency estimation have been proposed. These traditional formant frequency estimations techniques can be classified as being frequency domain or parametric. Frequency domain techniques involve estimating the formant frequencies from the frequency spectrum of the speech signal and include methods such as spectral peak picking from the short-time frequency spectrum. Parametric techniques are also called "analysis by synthesis" and involve generating a best match signal to the incoming signal based on a model of speech production. Traditional approaches to formant estimation do not accurately estimate formant frequencies in transient background noise such as from AWGN or background speakers. These algorithms are also not robust and are susceptible to being thrown off-track during unvoiced speech segments and are unable to recover quickly after periods of silence. Due to these limitations traditional techniques for formant frequency estimation cannot be used for obtaining the second formant frequency (F2) from continuous speech, in real-time, for use in CEFS amplification. The poor performance of these techniques also limits their use for other applications such as ASR, speech coding, etc.

Three algorithms that represent the best known traditional formant frequency estimation techniques have been selected for implementation in MATLAB in order to test and compare their performance with that of the formant estimation algorithm proposed in this thesis. The implementation of these traditional algorithms is discussed in some detail in this thesis. Further details on each implementation technique can be obtained from the references listed in each section.

5.1. Formant Frequency Estimation through Peak Picking of the Cepstrally Smoothed Spectrum

This is a frequency domain method and the formant frequency estimation technique involves computation of the smoothed spectrum, from the cepstrum, and then estimating the formant frequencies from the smoothed spectrum. The cepstrum is defined as the inverse transform of the log magnitude of the Fourier transform of the signal. The algorithm that was implemented is based on a paper by Schafer and Rabiner [8]. In this algorithm, formant frequencies are estimated from the smoothed log magnitude spectrum by adding constraints on the formant frequency ranges and relative levels of the spectral peaks in those frequencies ranges. The algorithm is designed to estimate the first three formant frequencies for voiced speech segments of male speakers. The block diagram for the smoothed spectrum peak picking based formant frequency estimation method is shown in Figure 5-1.

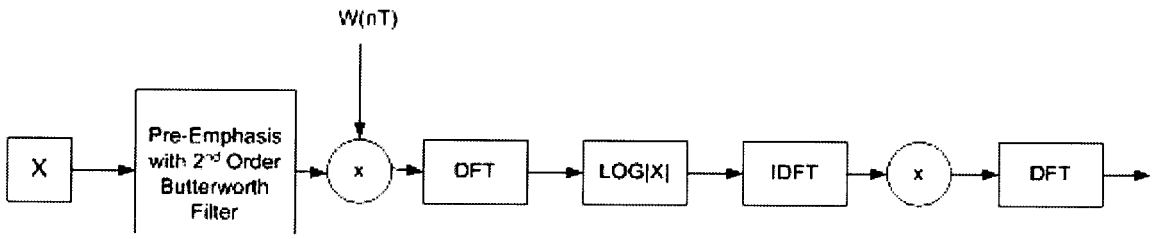

Figure 5 - 1 – Block diagram of Cepstral peak picking based formant frequency estimation technique

5.1.1. Estimation of the Spectral Envelope and Peak Picking

The speech signal is first segmented into non-overlapping 32 ms segments and then windowed using a Hamming Window of the same length. Each segment is pre-emphasised using the high-pass filter, shown in Figure 3-1, to remove the spectral tilt of the speech. Next, the DFT of each segment is evaluated and the log of the magnitude is taken, and then the inverse DFT of the log magnitude signal is calculated to obtain the cepstrum of the segment [8]. The 'low-time' part of the cepstrum corresponds to the vocal tract, glottal pulse and radiation information, while the 'high-time' part is due primarily to the excitation. The 'low-time' part of the cepstrum can be windowed or 'liftered' to remove the glottal pulsing information and extract the spectral envelope of the speech signal. The spectral envelope of the speech signal depends on the vocal tract and displays resonant structure of the vocal tract and its peaks correspond to the formant frequencies. Therefore, the formant frequencies are estimated from picking peaks of the spectral envelope, obtained from the 'cepstrally smoothed' log spectra [16].

However, limitations are placed on the frequency ranges in which the formant frequencies can lie to improve the accuracy of the estimation and to minimise the effect of transient noise. Therefore, the frequency locations and magnitudes of all the peaks of the cepstrally smoothed signal are found using a peak picking algorithm that picks all peaks where the slope of the signal changes from positive to negative. The three highest peaks corresponding to the formant frequencies are checked to see if they are adequately separated and lie in the correct frequency ranges to be assigned as the formant frequency estimates. However, the three highest peaks of the smoothed log spectrum of any segment are often either not adequately separated or lie in the wrong frequency ranges to be used as the first three formant estimates. Therefore, additional logic operations, discussed in the next section, are applied to improve estimates for the formant frequencies.

5.1.2. Estimation of Formant Frequencies from the Smoothed Spectrum

Before proceeding to details of the formant frequency estimation from the smoothed spectra, it is useful to know that the formant frequency ranges, to which the estimated formants frequencies are limited to, are derived from experimental data from male speakers. Table 1 shows these frequency ranges. For the three highest peaks of the smoothed spectrum to be the first three formant frequencies they must each be present in one of these formant frequency regions. The task of formant frequency estimation from direct peak picking is further complicated due to the high degree of overlap between the formant frequency regions.

Formant Frequency	**Frequency range in Hz** (Fmin – Fmax)
F1 region	200-900
F2 region	550-2700
F3 region	1100-3000

Table 1: Frequency ranges for the first three formant frequencies

If the peaks are located too close to each other they may smear together, making it impossible to identify the individual formant frequencies through peak picking of the smoothed spectrum. The solution is to enhance the frequency resolution of the smeared formant frequency region by using the chirp-z transform (CZT). The CZT increases the frequency resolution at the expense of a decreased temporal resolution of the smoothed spectrum [8] [16].

Estimation of F1

Formant Frequencies are picked in sequence, beginning with F1. First the amplitude of highest peak between 0-900 Hz is determined (F0AMP). The first formant frequency is

initially assigned to be the frequency of the highest peak in the F1 region and the amplitude of this peak is recorded (F1AMP). If F1AMP is greater than F0AMP-8.96 dB, then the current F1 is assigned to be the first formant frequency estimate and F1 estimation is complete. However, occasionally the magnitude of the F1 peak does not satisfy this condition because the F1 peak is merged and indistinguishable from the peak due to glottal pulsing. In this case the CZT is used to expand and enhance the region between 0-900 Hz [8].The highest peak of this enhanced section of the cepstrum is assigned to be the first formant frequency (F1) if the peak is within the F1 region and its amplitude is recorded (F1AMP). If the highest peak of the enhanced cepstrum lies outside the F1 region, the first formant frequency is assigned an arbitrary value of 200 Hz (F1min). Figure 5-2 shows the flowchart for estimating the first formant frequency from the cepstrally smoothed log spectrum of a speech segment.

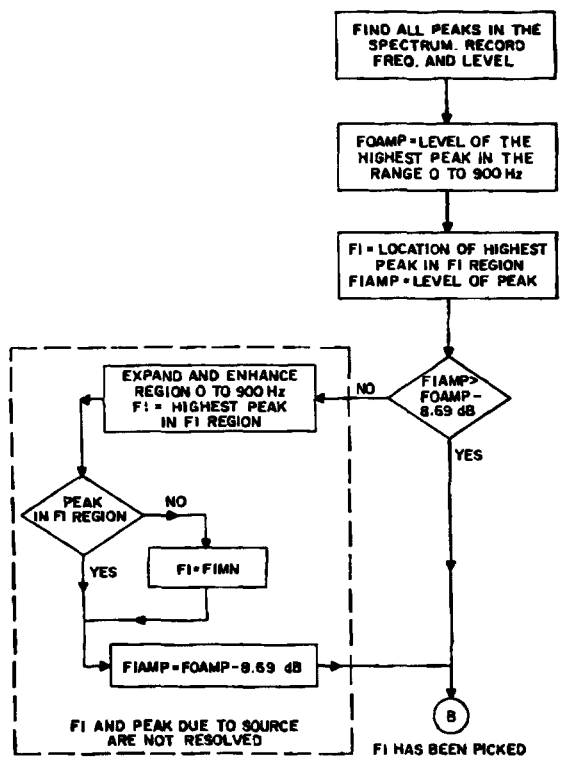

Figure 5 - 2 – Flowchart depicting the process of estimating F1 from the smoothed spectrum (Reprinted from Schafer et al. [8])

Estimation of F2

Figure 5-3 shows the flowchart for the determination of the second formant frequency. The frequency range to be searched for the second formant frequency depends on the value assigned to F1. If F1 is less than F2min, then only the F2 region defined in Table 1 is searched for second formant frequency estimation. However, due to the overlapping formant regions it is possible that F1 is greater than F2min in which case the value previously assigned to F1 can in fact be the second formant frequency. In the latter case, the lower limit of the region to be searched is set to F1min and the frequency region between F1min and F2max are searched. F2 is set to be the location of the highest peak in the search region and its amplitude is recorded (F2AMP). If F1AMP-F2AMP is greater than the frequency dependent threshold shown in Figure 5-4, then F2 is found and the threshold for F3 estimation is set as -17.38 dB. However, if no peak is found that exceeds the threshold then the F1 and F2 peaks are merged and further analysis using CZT is required to resolve these merged peaks. The enhanced region (F1–450 Hz to F1+450 Hz) is searched for peaks and the highest peak is assigned to F1 while the second highest peak is assigned to F2. If only one peak is found, then F2 is arbitrarily set to F1+200 Hz. If F2 has been located by CZT analysis, the threshold for F3 estimation is set to -1000 dB [8]. Finally, the values of F1 and F2 are checked to ensure that F1 is less than F2. If F1 is greater than F2, the frequencies are swapped.

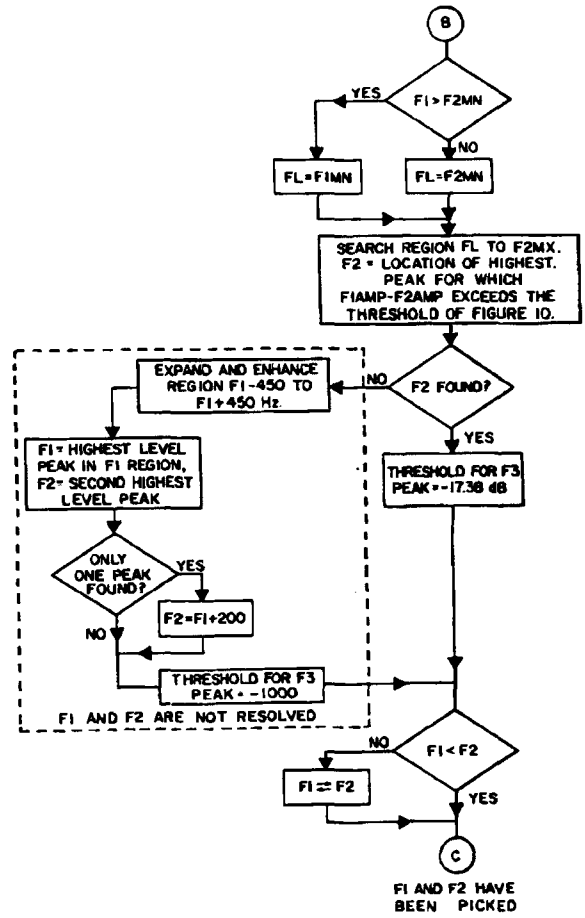

Figure 5 - 3 – Flowchart depicting the process of estimating F2 from the smoothed Spectrum (Reprinted from Schafer et al. [8])

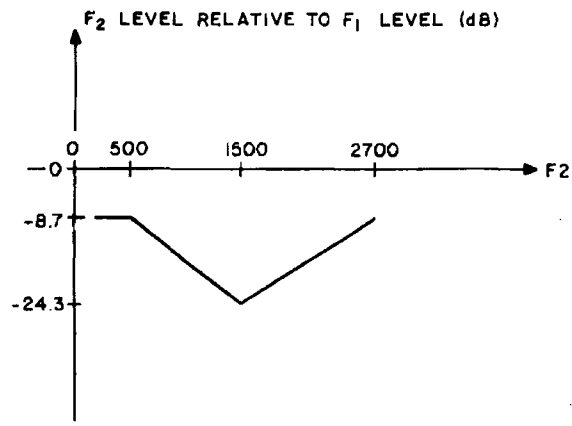

Figure 5 - 4 – Frequency dependent threshold for F2 estimation (Reprinted from Schafer et al. [8])

Estimation of F3

Figure 5-5 shows the flowchart for the determination of the third formant frequency. The frequency range to be searched for the third formant frequency depends on the value assigned to F2. If F2 is less than F3min, then only the F3 region defined in Table 1 is searched for the third formant frequency estimation. However, due to the overlapping formant regions, it is possible that F2 is greater than F3min. In this case, the value previously assigned to F2 can in fact be the third formant frequency. As such, the lower limit of the region to be searched is set to F2min and the frequency region between F2min and F3max is searched. F3 is set to be the location of the highest peak in the search region where the amplitude (F3AMP) is greater than the threshold set during F2 estimation. However, if no peak is found that exceeds the threshold, then the F2 and F3 peaks are merged and further analysis using CZT is required to resolve these merged peaks [8]. The enhanced region (F2–450 Hz to F2+450 Hz) is searched for peaks and the highest peak is assigned to F2 while the second highest peak is assigned to F3. If only one peak is found, then F3 is arbitrarily set to F2+200 Hz. Finally, the values of F2 and F3 are checked to ensure that F2 is less than F3. If F2 is not less than F3, then the frequencies are swapped.

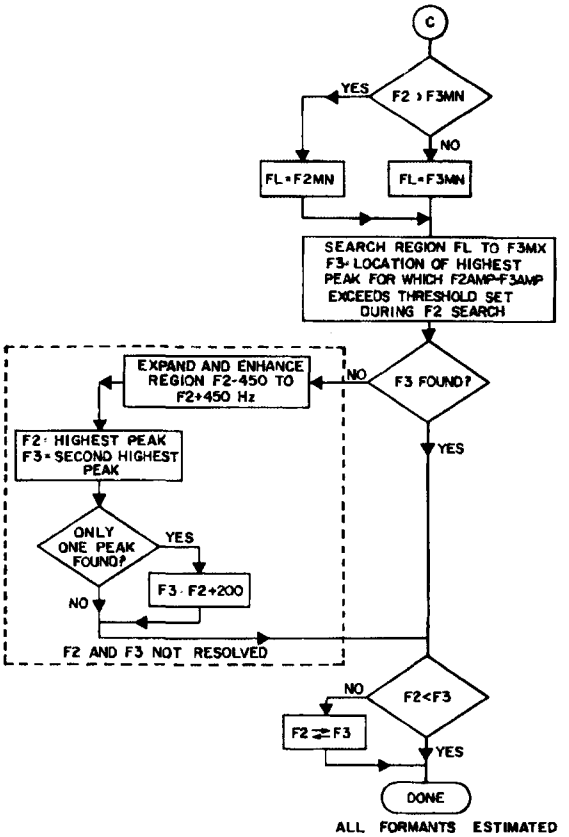

Figure 5 - 5 – Flowchart depicting the process of estimating F3 from the smoothed Spectrum (Reprinted from Schafer et al. [8])

These same calculations are repeated for each segment of the speech signal. The formants (F1, F2 and F3) are stored for each consecutive speech segment.

5.1.3. Results and Performance in noisy backgrounds

The algorithm was designed specifically for use during voiced speech segments, so it was slightly modified to work for sentences that consist of voiced and unvoiced speech segments. The voicing detector developed for use with the new formant tracking algorithm was adapted to work for the Schafer and Rabiner algorithm [8]. During unvoiced speech segments, the formant frequencies are assigned the value that was estimated during the previous voiced speech segment. Figure 5-6 shows the spectrogram of a synthesized male speaker saying "Five women played basketball", with the first three estimated formant frequencies obtained using this algorithm shown in white and the actual formant frequencies plotted in black. From the figure it can be seen that algorithm does not perform well even for the voiced segments of speech at a high SNR. It is also not accurate in estimating formant frequency transitions and is slow to respond to formant frequency movements.

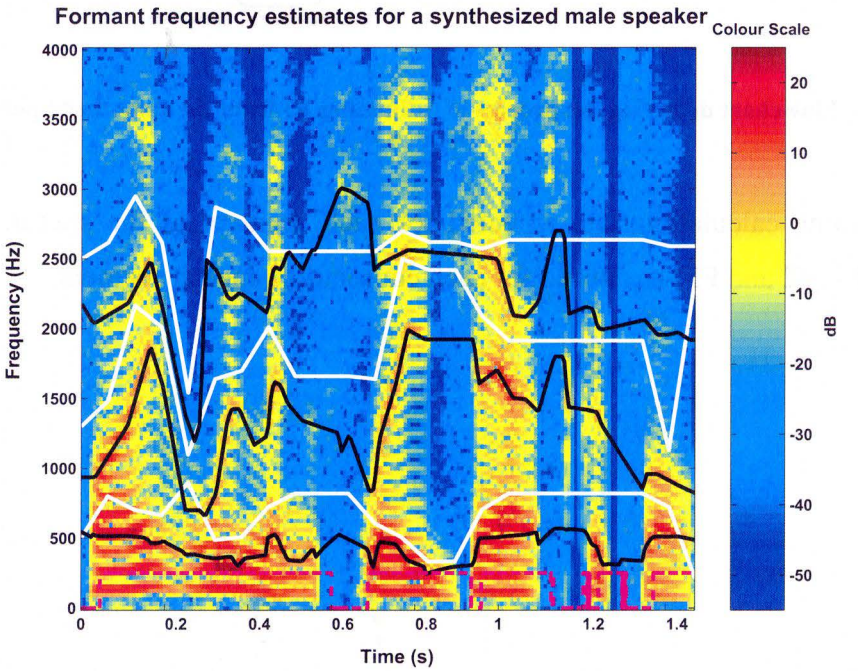

Figure 5 - 6 – Spectrogram and formant frequency estimates for a synthesized male speaker

Figure 5-7 shows the RMSE vs. SNR plot for the first three formant frequencies of a synthesized male speaker saying "Five women played basketball" in the presence of AWGN. The figure shows that the RMSE is very high even at high SNRs, which means that the algorithm is not able to track the formants very well.

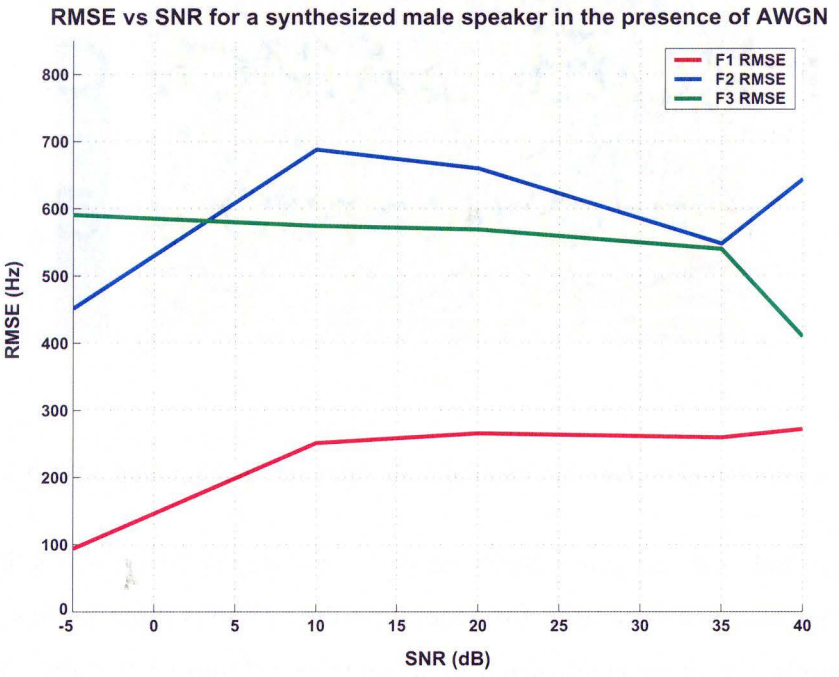

Figure 5 - 7 – RMSE vs. SNR for a synthesized male speaker in AWGN

The RMSEs drops sharply below 10 dB because below this SNR the formant frequencies are assigned an arbitrary value instead of being estimated from the cepstrally smoothed log spectrum. This occurs because the algorithm is unable to resolve distinct peaks for any of the formant frequencies at such a low SNR. Figure 5-8 shows a spectrogram for the same synthesized male speech sentence as in Figures 5-6 and 5-7 except that the SNR is 0 dB. It can be seen from the figure that the estimated formant frequencies are constant and remain at their arbitrary assigned values throughout most of the signal.

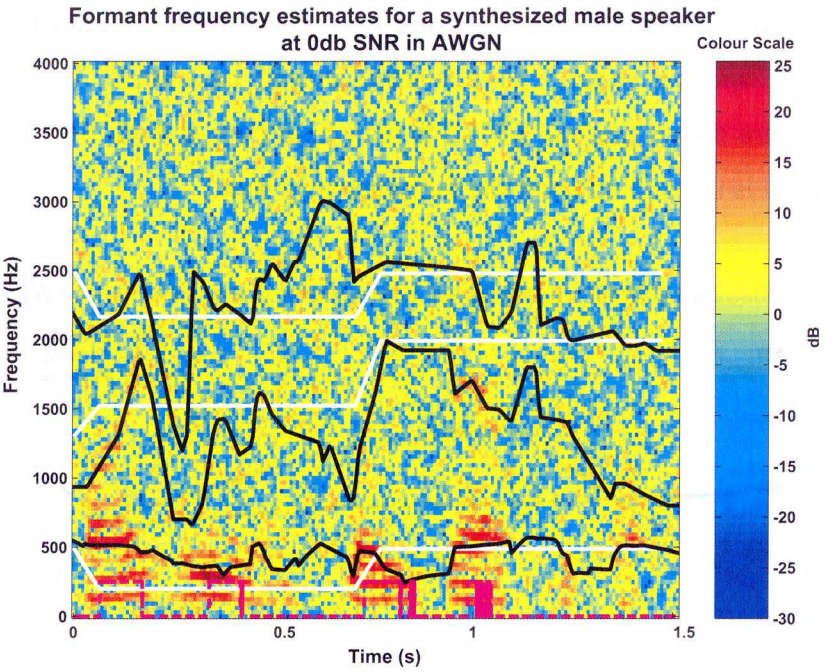

Figure 5 - 8 – Spectrogram for a synthesized male speaker in background AWGN at 0 dB SNR

To prove that that the poor performance of the algorithm was not limited to the synthesized speaker sentence used for generating the previous three figures, it was tested in a large number of other synthesized male sentences. Figure 5-9 shows the RMSE vs. SNR plot for the first three formant frequencies of a synthesized male speaker saying "Once upon a midnight" in the presence of AWGN. This figure shows similar trends to those observed in Figure 5-7 and confirms that the overall performance of the algorithm is poor.

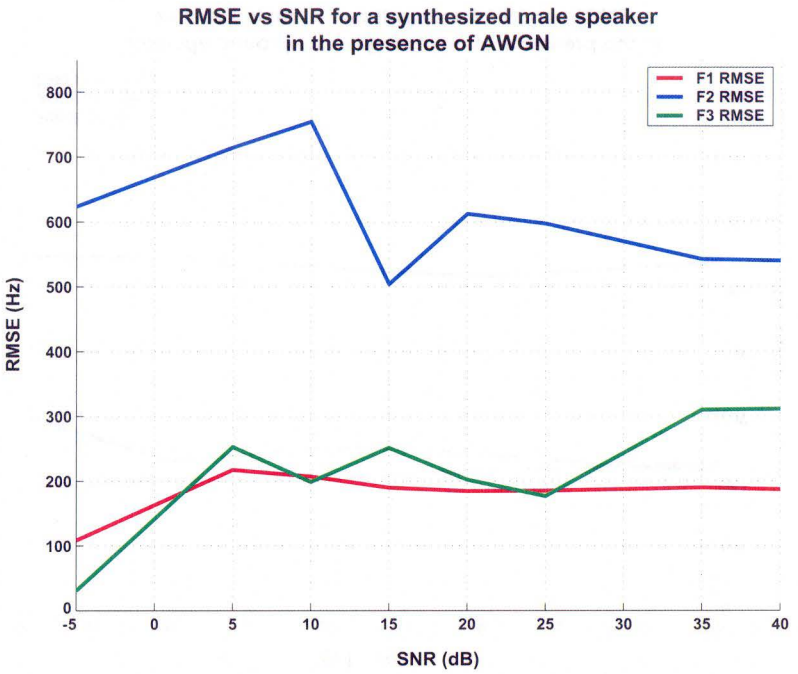

Figure 5 - 9 – RMSE vs. SNR for a synthesized male speaker in AWGN

The algorithm is also tested in the presence of a male single background speaker and in the presence of multiple background speakers for a wide range of SNRs. Figure 5-10 shows the RMSE vs. SNR plot for a synthesized male speaker saying "Five women played basketball" in the presence of a male single background speaker. Figure 5-11 shows the RMSE vs. SNR plot for the same synthesized male speaker in the presence of a multiple background speakers. From these figures it is clear that the performance of this algorithm is also poor in the presence of a male single background speaker as well as multiple background speakers.

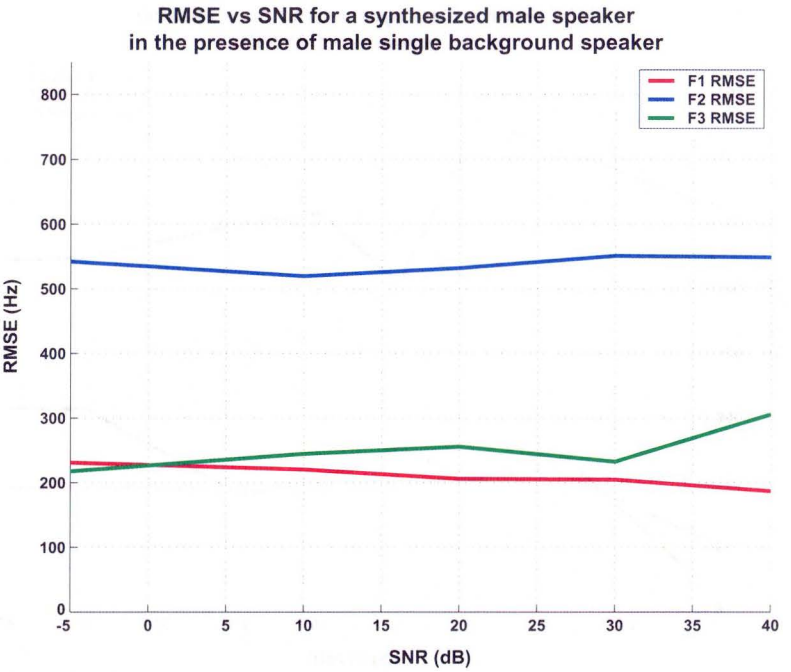

Figure 5 - 10 – RMSE vs. SNR for a synthesized male speaker in the presence of a male single background speaker

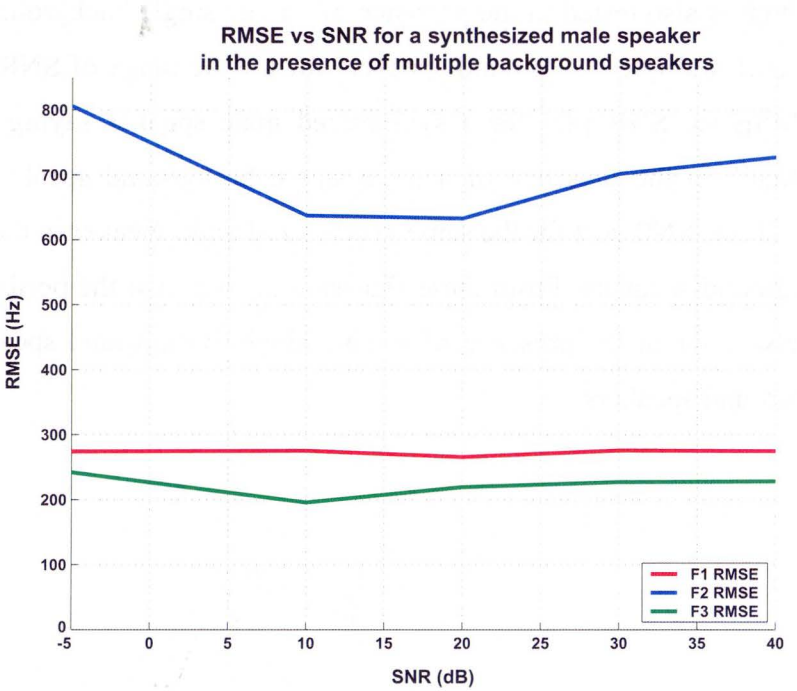

Figure 5 - 11 – RMSE vs. SNR for a synthesized male speaker in multiple background speakers

5.2. Formant Frequency Estimation using Linear Predictive Coefficients

Linear prediction analysis has been among the most popular methods for extracting spectral information from speech. Linear prediction of a speech signal involves modelling the next signal step as a linear combination of previous values in a statistically optimal way. The combination is a hypothetical proposal of the vocal tract impulse response. The parameters of the models are indicative of formant frequency positions hence, this is a parametric formant estimation technique. The solution of the linear prediction is a difference equation that expresses each sample of the original signal as a linear combination of the preceding samples. This difference equation is called the linear predictor and the coefficients of the equation are called the linear predictive coefficients (LPC). In the algorithm implemented [9], the first three formant frequencies are estimated from the peaks of the linear prediction spectra of the speech signal while minimizing the mean-square error between the predicted signal and the actual signal.

The proposed algorithm can only track formants in heavily voiced sounds and uses a pitch based voicing detector to determine whether a segment is voiced or unvoiced. However, the voicing detector that was built for the new formant tracker was supplemented to be used in this algorithm instead of the one proposed in the paper [9]. The speech is first pre-emphasised using a high pass filter (see Section 3.1) to remove the spectral tilt of the speech. Then the speech signal is segmented into non-overlapping segments of length 16 ms and then windowed using a Hamming window. Next, the 14th order LPCs of the signal are found and the frequency response of the all-pole filter described by those coefficients is computed. This frequency response is called the LPC spectrum of the signal. The formant frequencies are estimated from the peaks of the LPC spectrum. Figure 5-12 compares the 14^{th}-order LPC spectrum and the FFT of a segment of speech. From the figure it can be seen that the LPC spectrum is similar to a 'smoothed' version of the FFT spectrum of the speech segment.

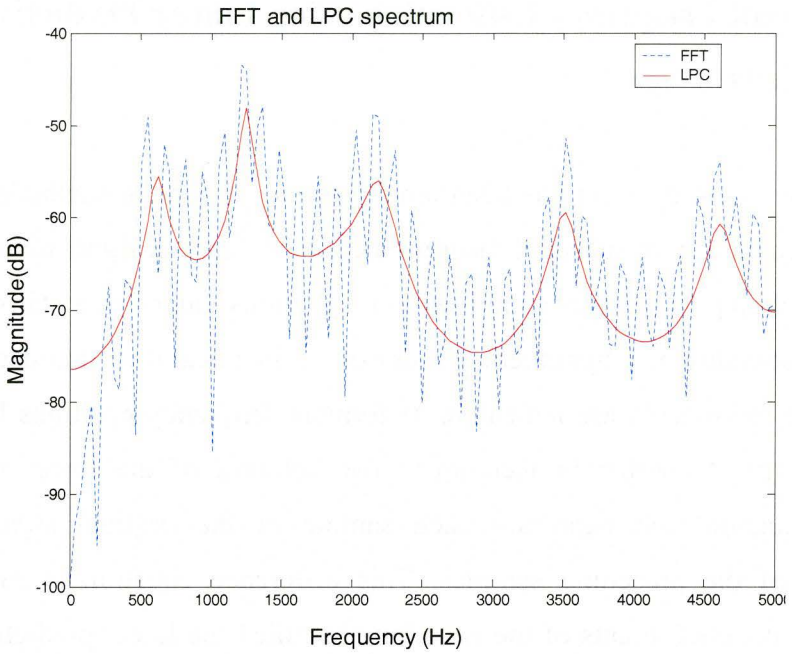

Figure 5 - 12 – The LPC spectrum and the FFT spectrum of a speech segment

The location and amplitude of all the peaks of the LPC spectrum are extracted and the peaks closest to the experimentally estimated anchor points are assigned as the best candidates for their respective formant frequency. For example, a peak at 500Hz would be closer to Est F1 than Est F2, and would therefore be assigned as the first formant frequency. The experimentally estimated anchor points for the formant frequencies for male and female speakers are shown in table 2 [9].

Anchor Points	Est F1 (Hz)	Est F2 (Hz)	Est F3 (Hz)	Est F4 (Hz)
Male	320	1440	2760	3200
Female	480	1760	3200	3520

Table 2: Formant Frequency anchor points estimated using experimental data

However, sometimes the peaks of the LPC spectrum are merged together and are too close to each other to be assigned as formant frequencies. In this case, the frequency spectrum is enhanced by using the CZT, which increases the frequency resolution at the expense of a decreased temporal resolution of the spectrum [8], [9]. If the peaks are still not resolved, the peak frequency being obtained for both formant values is compared to the anchor values and the peak is assigned to the formant closest to the anchor point. The other formant is either found by interpolation or by taking the mean of the already available data. Finally, successive formant frequency values are "smoothed" over time when a formant frequency is missing or is grossly out of range, by assigning that formant frequency its moving average value.

Figure 5-13 shows the spectrogram of a synthesized male speaker with the estimated and actual formant frequencies. From the figure it can be seen that the first formant frequency is tracked relatively well but the second and third formant frequencies are not accurately estimated using this algorithm.

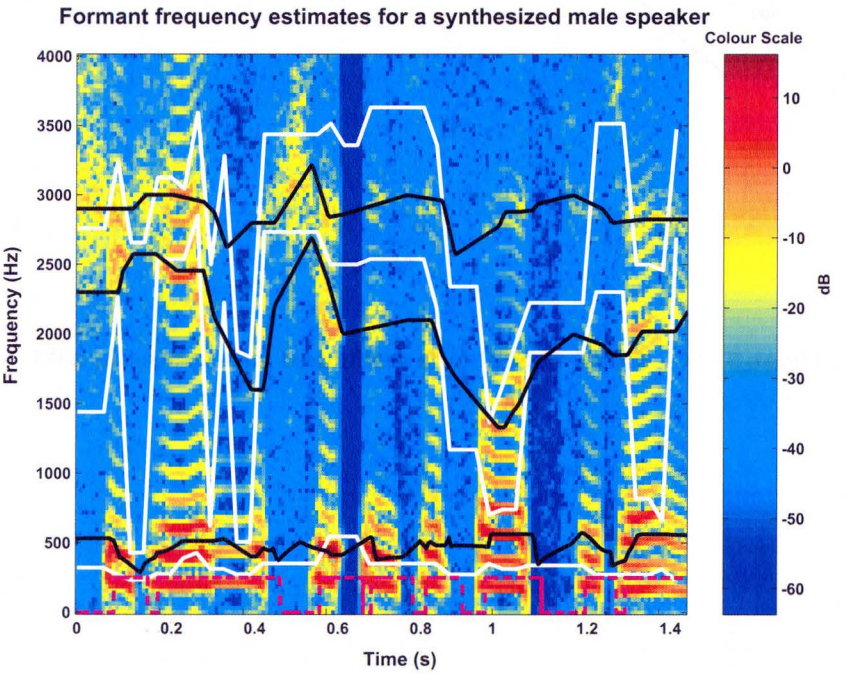

Figure 5 - 13 – Spectrogram for a synthesized male speaker

Figure 5-14 shows the RMSE vs. SNR plot for a synthesized male speaker saying "Five women played basketball" in the presence of AWGN. Figure 5-15 shows the RMSE vs. SNR plot for a synthesized female speaker saying the same sentence in the presence of a background AWGN. The figures illustrate that the performance of the algorithm is acceptable at high SNR levels for both male and female speakers. The RMSEs starts to drop below 20 dB because below this SNR level the formant frequencies are assigned an arbitrary value instead of being estimated from the spectrum. Overall, the algorithm performs poorly for both male and female speakers in AWGN below that SNR.

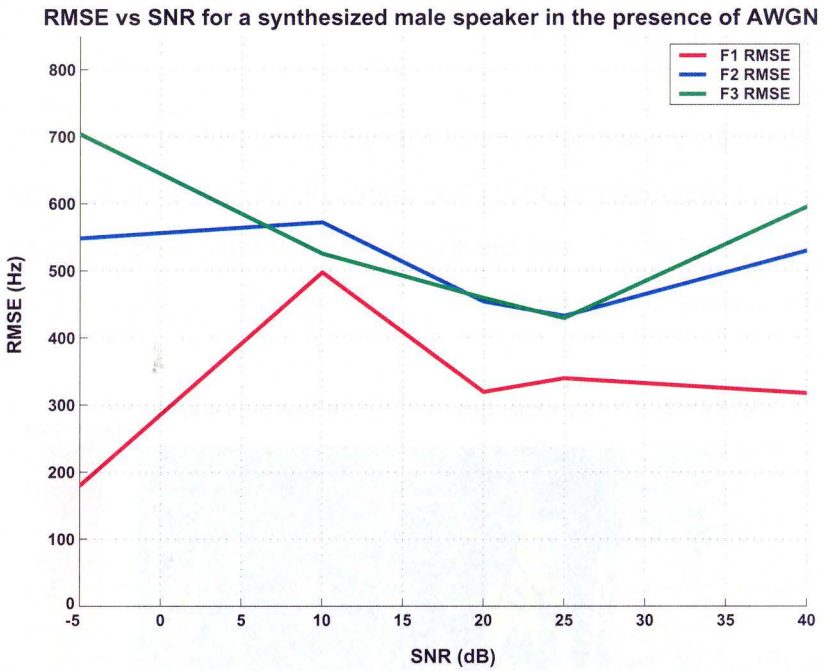

Figure 5 - 14 – RMSE vs. SNR for a synthesized male speaker in AWGN

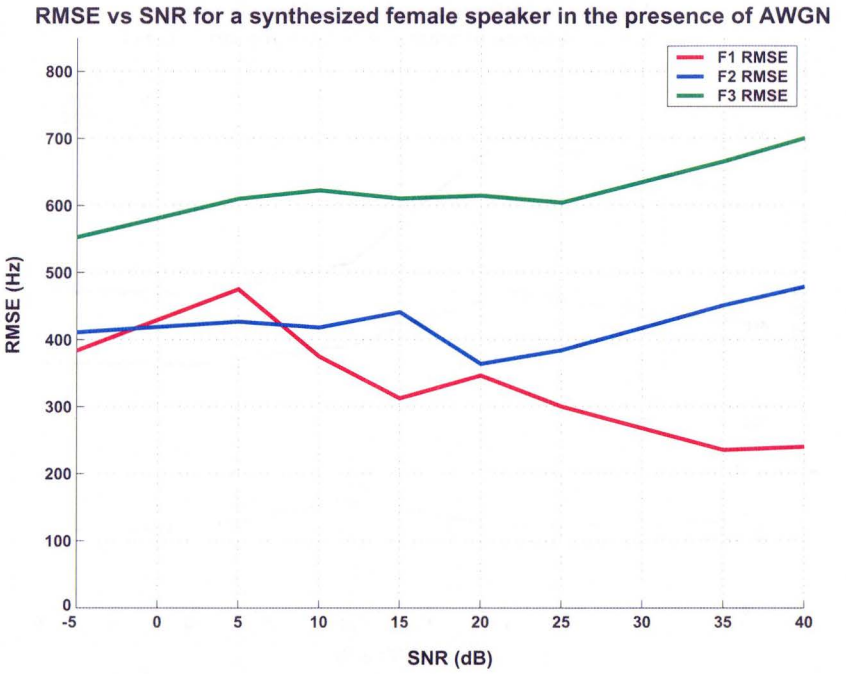

Figure 5 - 15 – RMSE vs. SNR for a synthesized female speaker in AWGN

The algorithm is also tested in the presence of single and multiple background speakers at varying SNRs. Figure 5-16 shows the RMSE vs. SNR for a female synthesized speaker saying "She gave the kitten to Budd today" in the presence of a male single background speaker. This figure shows that the second and third formant frequencies are affected greatly by the background speaker and as a result, their RMSEs are very high even at high SNRs. Figure 5-17 shows the RMSE vs. SNR for a synthesized male speaker saying "Five women played basketball" in the presence of multiple background speakers. This figure shows similar trends to those observed in the male single background speaker case and the second and third formant frequencies show an unusually large RMSE even at high SNRs.

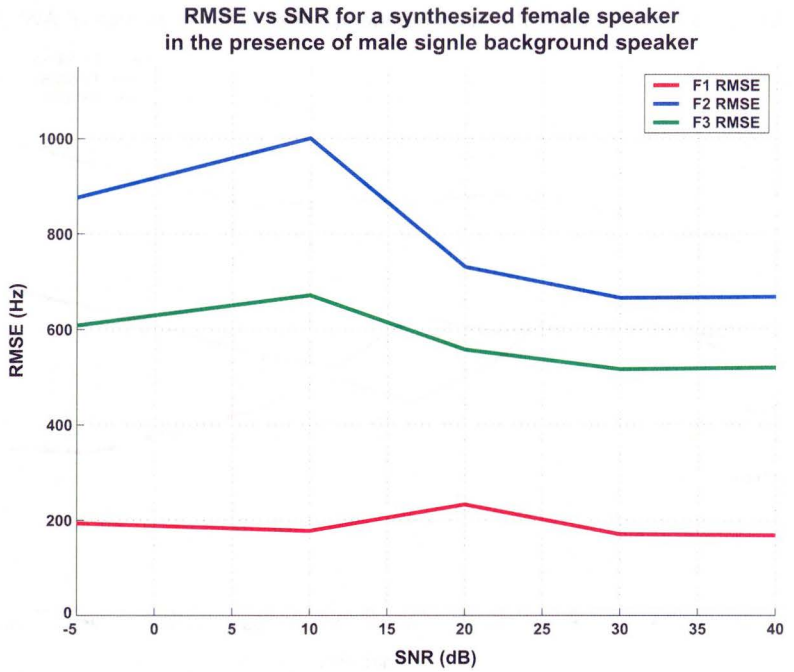

Figure 5 - 16 – RMSE vs. SNR for a synthesized female speaker in male single background speaker

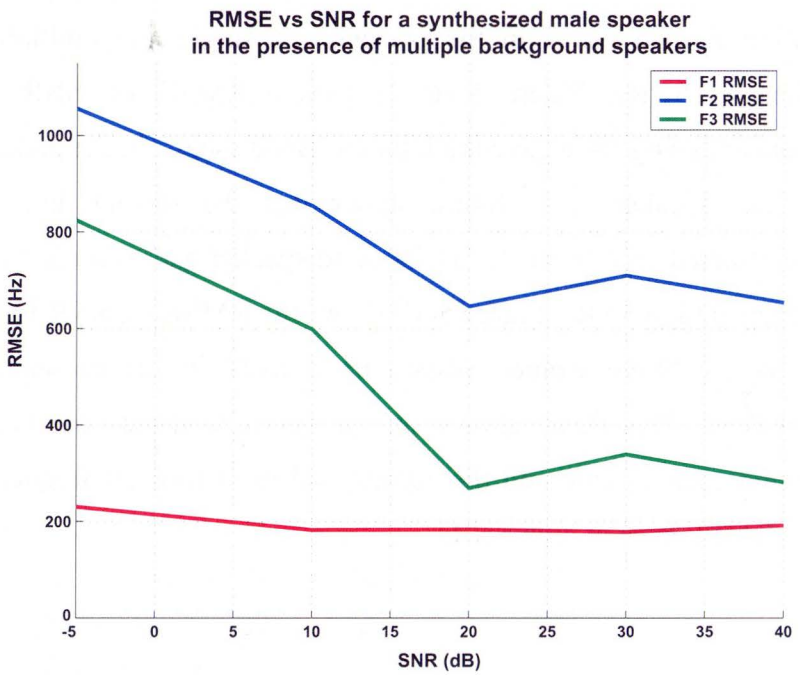

Figure 5 - 17 – RMSE vs. SNR for a synthesized male speaker in multiple background speakers

5.3. Formant Frequency Estimation using Physiological Models of the Ear

The ability of the human auditory system to process speech in noisy environments makes it superior to any human-designed speech processing system to date. Spectral estimation using auditory models has been shown to be efficient and robust but the success of the system depends on the accuracy and robustness of the auditory model it uses. A formant tracking algorithm that uses a human auditory model has been proposed by Metz et al. [10] and has been implemented using MATLAB.

The auditory model consists of stages for the outer, middle and inner ears. The output of the auditory model is the ensemble interval histogram (EIH), which shares similarities to the auditory nerve response of the mammalian ear. The algorithm proposes using the peaks of the EIH for estimating formant frequencies from voiced speech. The three highest peaks of the EIH for each short-time speech segment are designated as the three formant frequencies of that segment. Figure 5-18 shows the auditory model of the inner ear and cortical processing used by Metz et al. for formant tracking [10].

In this algorithm formant tracking is broken down into three main steps. First is the spectrum estimation of the speech signal (obtaining the EIH), second is the determination of the peaks from the spectrum and the last is the picking of the proper peaks. The speech signal is first pre-emphasised using a HPF (see Section 3.1) to remove the spectral tilt of the signal and then normalized so that the maximum amplitude of the signal is slightly above 40dB. Normalization is necessary for proper use of the level crossing detectors (LCDs) and will be discussed in detail later. Next, the signal is broken into non-overlapping 40 ms segments. Formant frequencies of each segment are estimated by passing it through each of the band pass filters (BPFs) and picking peaks from the EIH.

5.3.1. The BPFs

The band pass filters (BPF) are designed to simulate the psycho-acoustic tuning curves (PTCs) [10]. The model is made up of 90 uniformly distributed 2^{nd}-order Butterworth IIR filters. The bandwidth of each filter is constant at 200 Hz and the centre frequencies of the filters are separated by 50Hz. The BPFs are distributed evenly to cover a frequency range from 200 Hz to 5 kHz. Figure 5-19 shows the frequency and phase responses of the 2nd, 20th, 50th, 60th and 90th BPF.

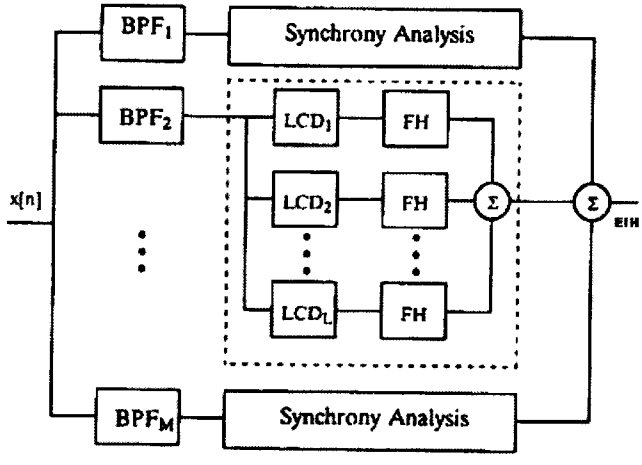

Figure 5 - 18 – The Auditory Model used by Metz et al. (Reprinted from Metz et al. [10])

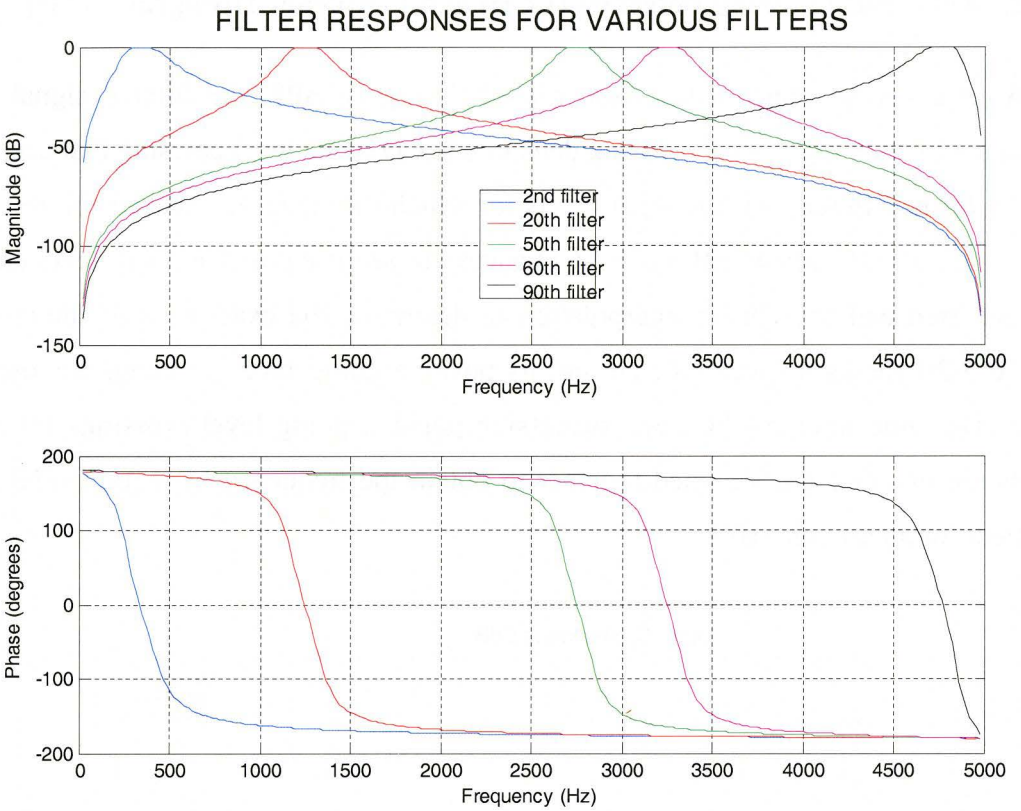

Figure 5 - 19 – The filter response of various BPFs

5.3.2. The Level Crossing Detectors (LCD) and Frequency Histograms (FH)

After a speech segment has been passed through a BPF, the filtered signal is put through a bank of eight level crossing detectors (LCD). The level crossing that each LCD checks for is different and the eight levels for which the signal is checked for are: 5, 10, 15, 20, 25, 30, 35, and 40 dB. Each LCD checks to see if a positive going level crossing has occurred and uses linear interpolation to determine the exact time of the crossing. Figure 5-20 shows a speech signal and its positive going level crossings for the 5 dB LCD. The time intervals between successive positive going level crossings for all the LCDs are inverted and summed together to obtain the frequency histogram (FH) of the segment for a particular BPF.

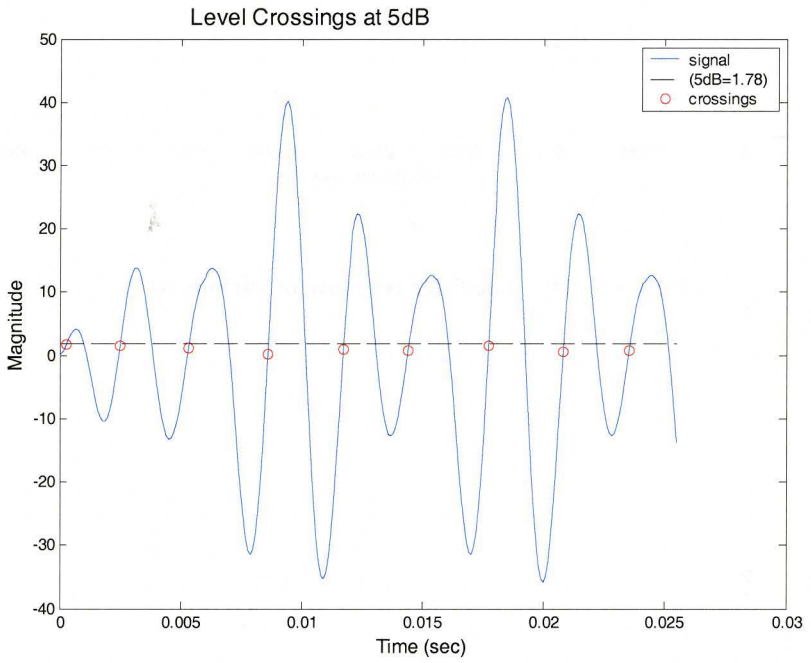

Figure 5 - 20 – The filter response of various BPFs

The segment is passed through all 90 of the BPFs and the above process is repeated. Finally, the FHs from each BPF are summed to obtain the EIH for the segment [10].

Figure 5-21 shows a sample EIH of a segment of speech. Notice that the first three formant frequencies of the segment of speech have the three highest peaks of the EIH.

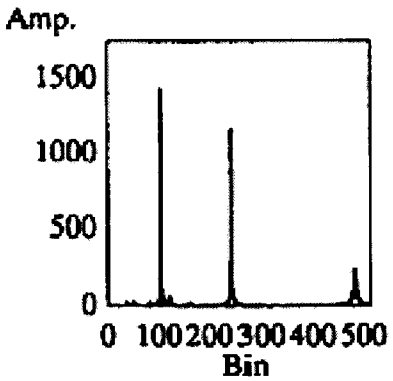

Figure 5 - 21 – The EIH of a segment of speech (Reproduced from Metz. et al. [10])

A peak picking algorithm picks the top three peaks of the EIH for every segment of speech and these are assigned to be the first three formant frequencies of that segment.

5.3.3. Results

The Metz et al. algorithm works well for estimating the first three formant frequencies of sustained vowels whose formant frequencies remain constant. However, the parameters of the algorithm have to be modified for each sustained vowel in order for it to work properly. Figure 5-22 shows the spectrogram of a sustained vowel /a/ with constant actual formant frequencies. It is clear from the spectrogram that the algorithm is able to predict the three formant frequencies very accurately. However, the algorithm performs very poorly for speech signals where the formant frequencies are non-stationary. Figure 5-23 shows the spectrogram of a synthesized female speaker saying "Five women played basketball". From this figure it is clear that the algorithm is not able to estimate formant frequencies accurately even at high SNRs.

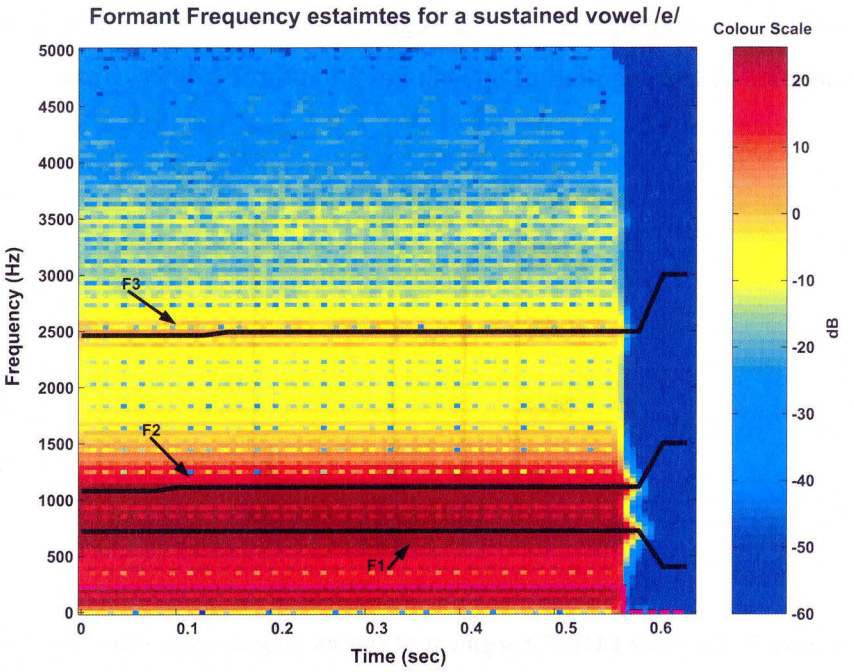

Figure 5 - 22 – Spectrogram and estimated formant frequencies for a sustained vowels

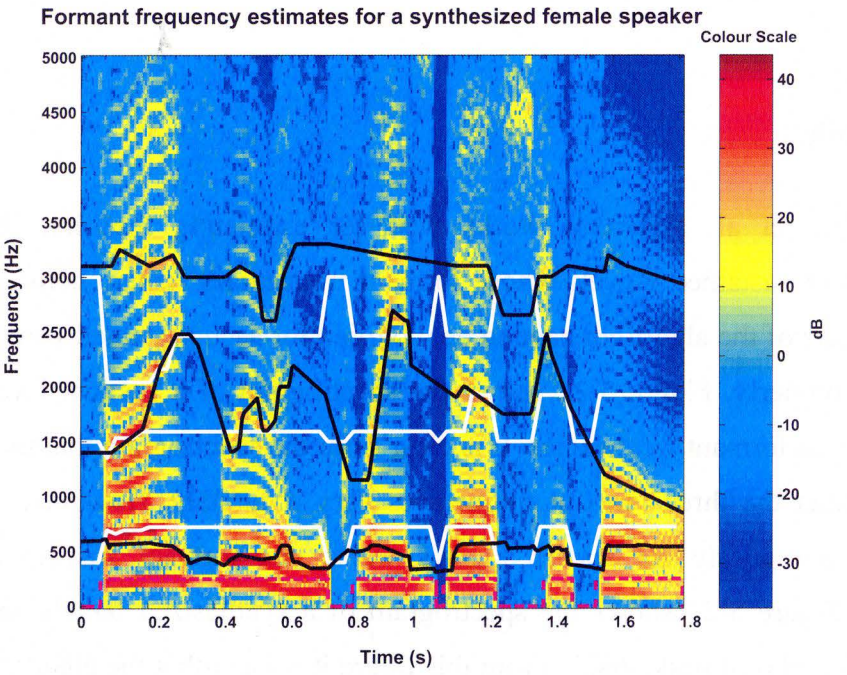

Figure 5 - 23 – Spectrogram for a synthesized female speaker

Figure 5-24 shows the RMSE vs. SNR for a synthesized male speaker saying "Five women played basketball" in the presence of AWGN. This figure shows that the RMSEs are constant across the different SNRs. This is because (as seen from Figure 5-23) the estimated formant frequencies remain constant throughout the signal. Therefore, the difference between the actual and estimated formant frequencies also remains constant.

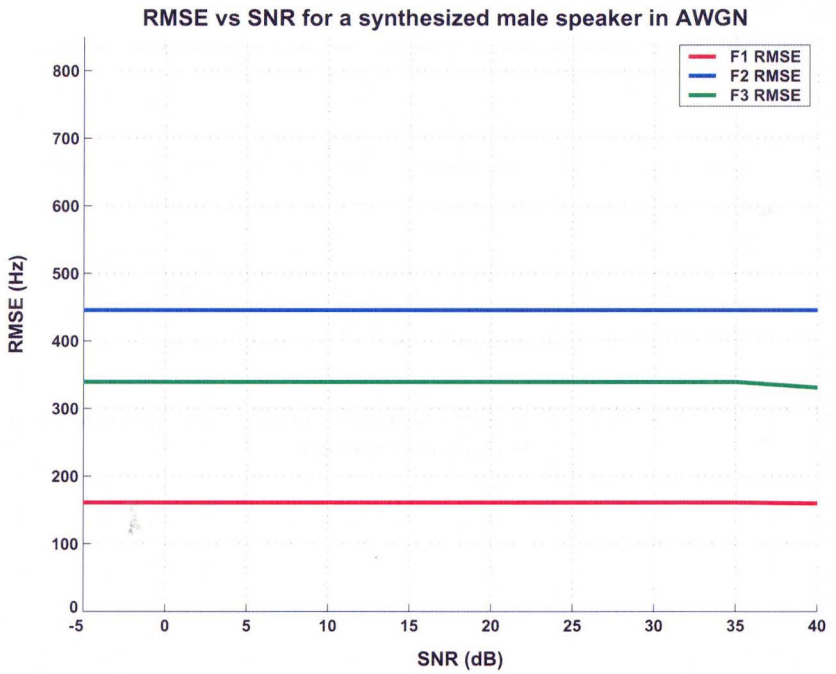

Figure 5 - 24 – RMSE vs. SNR for a synthesized male speaker in AWGN

Figure 5-25 shows the RMSE vs. SNR for a synthesized female speaker saying "Five women played basketball" in the presence of a male background speaker. Overall, the algorithm performs poorly for all types of sounds except sustained vowels.

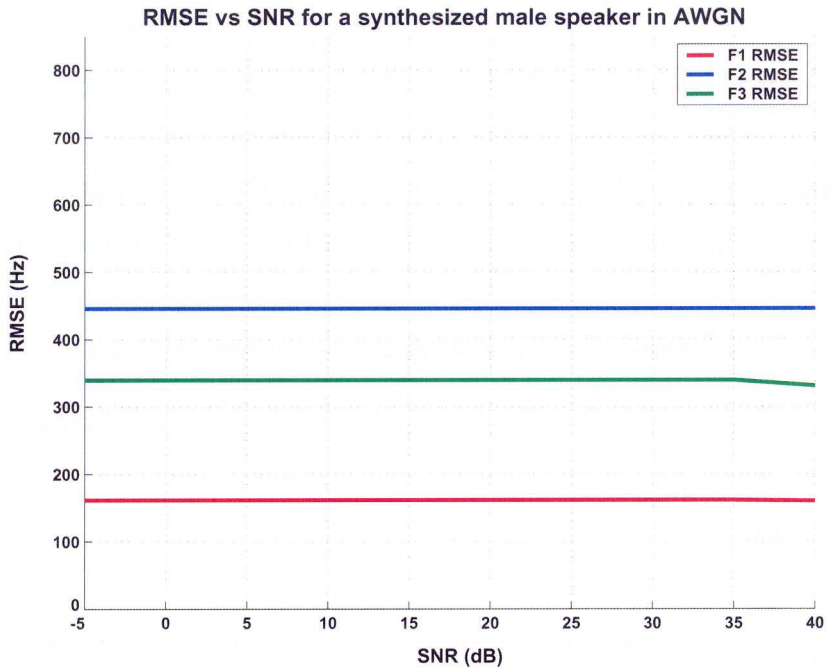

Figure 5 - 25 – RMSE vs. SNR for a synthesized female speaker in the presence of a male single background speaker

6. CONCLUSIONS

Quantitive analysis of the formant tracking algorithm described in this thesis has shown that it provides accurate formant frequency estimates for both male and female speakers for a wide range of SNRs in real-life noise conditions such as AWGN, a single competing background speaker (male and female), multiple background speakers, and reverberant acoustic environments. The algorithm provides mostly smooth formant frequency estimates. The formant tracker is also robust and recovers quickly after erroneous estimates to go back to tracking the actual formant frequencies in the speech signal. There have been some problems identified with the formant tracker. The algorithm occasionally gives 'choppy' and oscillating formant frequency estimates. This is an undesirable result because the actual formant frequencies of speech normally vary slowly with time and have smooth transitions. This problem is only encountered when the SNR is very low (typically below 5 dB as in Figure 4-21) and occurs due to the algorithm tracking the excess energy added outside the formant frequency regions from the background noise source. However, the overall performance of the proposed formant tracking algorithm is still much better than those of traditional formant estimation techniques.

The algorithm developed in this thesis is geared primarily towards use for CEFS amplification. It was identified earlier that in order to apply CEFS to continuous speech the second formant frequency has to be estimated accurately and in real-time. Furthermore, the estimated formant frequencies have to be smooth and the algorithm has to be able to identify formant transitions accurately so that the proper frequency-dependent amplification is applied to the speech signal. Testing on the algorithm has shown that the formant frequency estimates are smooth and the formant frequency transitions are tracked accurately. The algorithm has been designed to operate in real-time and estimate formant frequencies from continuous speech for both male and female

speakers. Therefore, the formant tracking algorithm developed in this thesis can be used to implement CEFS amplification.

Limitations of Traditional Formant Frequency Estimation Techniques

The traditional formant frequency estimation techniques that were implemented have shown that they do not meet the criteria to be able to provide formant frequency estimation for CEFS amplification. The formant frequency estimation based on peak picking of the cepstrally smoothed spectrum provides good formant frequency estimates at high SNRs for voiced speech segments of male speakers. However, its performance degrades considerably in background noise (AWGN and single competing speaker) at lower SNRs and it cannot track formant frequency transitions accurately even at moderate SNRs. The algorithm is not designed to work for female speakers and many of the parameters have to be adjusted manually for the algorithm to function for female speakers. The technique also requires a large number of logic operations (eg. Figure 5.2, 5.3, 5.5) in order to constrain and refine the formant frequency estimates and is therefore computationally complex.

The formant frequency estimation method based on LPCs provides good first formant frequency estimates for both male and female speakers at high SNRs. However, the estimates for the second and third formant frequencies are poor even at high SNRs and the algorithm is not able to track formant frequency transitions. The formant frequency estimates that the algorithm provides are not smooth and have large jumps. The overall performance of the algorithm deteriorates significantly in the presence of background noise (AWGN, single competing speaker, or multiple background speakers) at moderate SNRs. The algorithm also requires a large number of logic operations in order to constrain and refine the formant frequency estimates making it computationally complex.

The physiological model based formant frequency estimation technique was unable to provide accurate formant frequencies for continuous speech for either male or female speakers.

Other Applications of the Formant Tracking Algorithm

Although the algorithm developed in this thesis is primarily designed to meet the criteria for CEFS amplification, other applications were identified earlier. Speaker identification systems often use front-end phonetic segmentation prior to feature extraction from phonemes [21]. For such an application, the algorithm no longer needs to obtain formant frequencies from continuous speech and only needs to operate on voiced speech segments. The algorithm can be modified to remove the voicing detector and the moving average decision maker so that the formant frequencies are always estimated by spectral estimation. The algorithm should perform very well in this environment and will be able to provide accurate formant frequency estimates for the voiced speech segments.

Another application identified for the formant tracking algorithm is for speech coding. For this application it may be more important that the algorithm provides actual formant frequency estimates of the speech even if the estimates are not smooth and that the algorithm is able to recover quickly. The formant tracking algorithm can be modified to remove the moving average decision maker so that the algorithm is able to quickly recover and provide erratic but accurate formant frequency estimates. The algorithm will also be able to identify voiced speech segments for which it is able to provide accurate estimates.

The formant tracking algorithm can also be used for concatenation synthesis of speech [18]. For this application the formant frequencies have to be accurately identified during phoneme transitions. Since the formant tracker is already able to identify formant

transitions accurately, it should be able usable for this application without extensive modifications.

Future Work

The oscillating formant frequency problem may be solved in future updates to the formant tracker by either smoothing the formant frequency estimates or by incorporating additional logical limitations to prevent abnormal jumps in the formant estimates. Another future improvement may be to modify the formant pre-filters to have variable bandwidths that are dependent on the magnitudes of the poles estimated by the linear prediction coefficients. This may further improve the formant estimates during rapid formant transitions at high SNRs, but the performance at low SNRs would likely remain unchanged.

7. BIBLIOGRAPHY

[1] Sachs, M. B., Bruce, I. C., Miller, R. L., and Young, E. D. "Biological Basis of Hearing-Aid Design," *Annals of Biomedical Engineering* 30:157–168, 2002.

[2] Quatieri, T. F. *Discrete-Time Speech Signal Processing.* Prentice Hall, Upper Saddle River, NJ, 2002

[3] Deller, J. R. Jr., Proakis, J. G. and Hansen, J. H. L. *Discrete-Time Processing of Speech Signals*, 1st edition, Macmillan Publishing Company, New York, 1993

[4] Pols, L. van der Kamp, C., L. J. and Plomp, R. "Perceptual and physical space of vowel sounds," *Journal of the Acoustical Society of America* 46:458–467, 1969.

[5] Miller, R. L., Calhoun, B. M. and Young, E. D. "Contrast enhancement improves the representation of /ɛ/ like vowels in the hearing-impaired auditory nerve," *Journal of the Acoustical Society of America* 106:2693–2708, 1999.

[6] Liberman, M. C. and Dodds, L. W. "Single-neuron labeling and chronic cochlear pathology. III. Stereocilia damage and alterations of threshold tuning curves," *Hearing Research* 16:55–74, 1984.

[7] Pickett, J. M. *The Sounds of Speech Communication.* Pro-Ed Inc., Austin, Tx, 1980.

[8] Schafer, R.W., Rabiner, L.R. "System for automatic formant analysis of voiced speech," *Journal of the Acoustical Society of America* Vol.47, No.2, 1970, pp 634–650.

[9] McCandles, S.S. "An algorithm for automatic formant extraction using linear prediction spectra," *IEEE Transactions on Acoustics, Speech, and Signal Processing* 22: 135–141, 1974.

[10] Metz, S.W., Heinen, J.A., Niederjohn R.J., Sreenivas T.V. "Auditory modeling applied to formant tracking of noise-corrupted speech," in *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation - 1991 (IECON '91),* vol.3, pp. 2120–2124.

[11] Rao, A. and Kumaersan, R. "On decomposing speech into modulated components," *IEEE Transactions on Speech and Audio Processing* 8:240–254, 2000.

[12] Bruce, I. C., Karkhanis, N. V., Young E. D., and Sachs, M. B. "Robust formant tracking in noise," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing - 2002*, Vol. I, pp. 281–284.

[13] Mustafa, K. and Bruce, I. C. "Robust formant tracking for continuous speech in speaker variability," in *Proceedings of the International Symposium on Signal Processing and its Applications - 2003*, Vol. 2, pp. 623–624.

[14] Rabiner, L.R. and Schafer, R.W. "On the behaviour of minimax FIR digital Hilbert transformers," *The Bell System Technical Journal* Vol. 53, No. 2, 1974.

[15] *Programs for digital signal processing*, IEEE Press, New York, 1979.

[16] Rabiner, L.R. and Schafer, R.W. *Digital processing of speech signals*. Prentice Hall, Englewood Cliffs, 1978.

[17] Sondhi, M. M. "New methods of pitch extraction," *IEEE Transactions on Audio and Electroacoustics* Vol. AU-16, No. 2, pp. 262–266, 1968.

[18] Ding, W and Campbell, N. "Optimising unit selection with voice source and formants in the CHATR speech synthesis system," in *Proceedings of Eurospeech 1997*, Rhodes; pp. 537–540.

[19] Lincoln, M., Cox, S. and Ringland, S. "A fast method of speaker normalisation using formant estimation," in *Proceedings of Eurospeech 1997*, Rhodes; pp. 2095–2098.

[20] Högberg, J. "Data driven formant synthesis," in *Proceedings of Eurospeech 1997*, Rhodes; pp. 565-568.

[21] Park, A. and Hazen, T. J. "ASR dependent techniques for speaker identification," in *Proceedings of the International Conference on Spoken Language Processing 2002*, pp. 478-479.

[22] Peterson, G. E. and Barney, H. L. "Control methods used in a study of the vowels," *Journal of the Acoustical Society of America* 24:175–184, 1952.

APPENDIX I - CODE

The MATLAB code used to implement the formant tracking algorithm is also publicly available from this webpage:

http://www.ece.mcmaster.ca/~ibruce/

Formant Tracker Frontend

```
%function F = formant_tracker_frontend

% FORMANT_TRACKER_FRONTEND
%
%  F = FORMANT_TRACKER_FRONTEND
%
%  This function acts as the frontend for the 'FORMANT_TRACKER_BACKEND.m' function, which
it calls.
%   It provides similar functionality to the 'FORMANT_TRACKER.m' function but all the
parameters for
%    the function are passed on to the 'FORMANT_TRACKER_BACKEND.m' function by this
function. All adjustments
%  to the formant tracker parameters can be made here.
%
%
%   See also FORMANT_TRACKER_BACKEND, FORMANT_TRACKER, and FORMANTFILTERS.

% Inputs
% none
%
% Outputs
% none
%
% Author: Kamran Mustafa
% E-mail: mkamran@hotmail.com  or  mkamran@ieee.org
%
% Modification List:
%
% June 13, 2002 - First Created
% June 16, 2002 - Modified to add more parameters
% June 17, 2002 - Added more parameters
% June 18, 2002 - Added more parameters
% June 22, 2002 - Modified function to accomodate independent RMS threshold Ratios for
each formant
% June 24, 2002 - Made changes to the RMS threshold levels to see if a moving average of
RMS thresholds is feasible
%                 for each of the formants.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Clear MATLAB Memory
clear all;
clc;


%Setup and load the signal and the correstaponding Actual Formant Frequncies
[X, Fs] = wavread('fwpb_female_energy_modified_1.wav');     %<---change filename here
load 'fwpb_female_energy_modified_1hl.mat';
```

```
%downsample to 8 Khz
X = resample(X,8e3,Fs);

%New Sampling Frequency (after downsampling)
Fs = 8e3;


% %Add noise to the original signal
% X = addnoise(X,30);


%LPC Window size assignment - 20 ms
lpc_window_size = 0.02*Fs;

%Declare the number of filters to create the window for
num_of_filters = 4;
%LPC Window - of duration LPC windowsize
window = repmat(hamming(lpc_window_size,'periodic'),1,num_of_filters);

%%%%%%%%%%%%%%%%%%VOICING DETECTOR INITIALIZATIONS%%%%%%%%%%%%%%%%%%%%

%Cut-ff frequency for the voicing detector HPF and the LPF
Filter_Cutoff = 700;

%The Hysterisis Log Ratio Threshold1 for switching from unvoiced speech to voiced speech
(0 --> 1)
Log_ratio_threshold1 = 0.2;

%The Hysterisis Log Ratio Threshold1 for switching from voiced speech to unvoiced speech
(1 --> 0)
Log_ratio_threshold2 = 0.3;

%The Threshold Level for use with the Autocorrelation vs. logratio calculation for white
noise consideration
Autocorrelation_Threshold_Level = 0.4;

%Set RMS Ratio threshold value
RMS_Ratio_F1 = -35;
RMS_Ratio_F2 = -40;
RMS_Ratio_F3 = -45;
RMS_Ratio_F4 = -50;

%%%%%%%%%%SET ALL PARAMETERS BEFORE THIS POINT%%%%%%%%%%%%%%%%%%%%%%%%

%Call the Formant Tracking function
[F, Voice, Y_RMS, Avg_Y_RMS, Gender, Pitch] = formant_tracker_backend (X, Fs,
lpc_window_size, window, Filter_Cutoff, Log_ratio_threshold1, Log_ratio_threshold2,
Autocorrelation_Threshold_Level, RMS_Ratio_F1, RMS_Ratio_F2, RMS_Ratio_F3, RMS_Ratio_F4);


%%%%%%%%%%%%%%%%%%%%%%PLOT THE RESULTS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Spectrogram
figure;
specgram(X,256,Fs,256,192)
caxis ([-65 5]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
hold on;
plot(time/1000-(256/2)*(1/Fs),f1, 'k -','linewidth',2.5);
plot(time/1000-(256/2)*(1/Fs),f2, 'k -','linewidth',2.5);
plot(time/1000-(256/2)*(1/Fs),f3, 'r - ','linewidth',2.5);
plot(time/1000-(256/2)*(1/Fs),f4, 'r -','linewidth',2.5);
plot([0:1/Fs:(length(X)-1)/Fs]-
(((lpc_window_size+256)/2)+10)*(1/Fs),F,'w','linewidth',2.5)
plot([0:1/Fs:(length(X)-1)/Fs]-(((lpc_window_size+256)/2)+10)*(1/Fs),250*Voice,'m       --
','linewidth',2)
```

Formant Tracker Backend

```
function [F, Voice, Y_RMS, Avg_Y_RMS, Gender, Pitch] = formant_tracker_backend (X, Fs,
lpc_window_size,   window,   Filter_Cutoff,   Log_ratio_threshold1,   Log_ratio_threshold2,
Autocorrelation_Threshold_Level, RMS_Ratio_F1, RMS_Ratio_F2, RMS_Ratio_F3, RMS_Ratio_F4)

% FORMANT_TRACKER_BACKEND
%
%  F = FORMANT_TRACKER_BACKEND f
%
%  This function acts as the backend for the 'FORMANT_TRACKER_FRONTEND.m' function, which
calls it.
%   It provides similar functionality to the 'FORMANT_TRACKER.m' function but all the
parameters for
%  the function are passed on to it from the 'FORMANT_TRACKER_FRONTEND.m' function.
%
%
%   See also FORMANT_TRACKER_FRONTEND, FORMANT_TRACKER, and FORMANTFILTERS.

% Inputs
% none
%
% Outputs
% F = The 4 formants frequencies
%
% Author: Kamran Mustafa
% E-mail: mkamran@hotmail.com  or  mkamran@ieee.org
%
% Modification List:
% June 13, 2002 - First Created from formant_tracker.m function
% June 16, 2002 - Changed the method to calculate the initial formant frequencies
% June 17, 2002 - Modified method to calculate initial formant frequency assignments
% June 18, 2002 - Added the formant filter parameters to the list of modifiable
parameters.
% June 24, 2002 - Modified to adjust to negative frequency predictions by the LPC
% June 25, 2002 - Added an adaptive RMS_Threshold Level detector based on a 'moving
average' RMS_Threshold Level
% July 8, 2002  - Added Moving average based RMS Threshold Levels
% July 9, 2002  - Modified teh Moving Average based RMS Threshold Levels to decay when
below the threshold
% July 16, 2002 - Fixed problems with the initializing of the formant frequencies before
the LPC starts
% September 10, 2002 - Started playing around with the Pre-Empahsis.
% September 11, 2002 - Added a second order Pre-Emphasis filter with a higher spectral
tilt below the cut-off freq.
% September 13, 2002 - Selected a 1st order Buterworth IIR Pre-Emphasis filter after
trying different types of Pre-Emphasis filters.
% September 25, 2002 - Started integrating the pithch based gender detector into the
tracker.
% September 26, 2002 - Completed integrating the gender detector and started testing the
results
% October 12, 2002 - Made changes to the gender detector so that it only polls for voiced
speeech.
% October 22, 2002 - Modified the function to inculde Pitch based calculations when
calculatinh formants.
% April 19, 2003 - Cleaned Code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Turn off warnings
warning off;

%Equalize the input file to have an RMS of 1
X = X./rms(X);
```

```
%Replicate the origianl signal to be sent to the voicing detector
X_VD = X';

%Apply the Pre-Amphasis filter to the input signal
Xhf_preemphasis = filter([0.2618 -0.2618],[1 0.4764],X); %Butterworth IIR filter, Order
1, stable

%Hilbert Transform the signal into a complex signal using the FIR Hilbert function.
X = firhilbert(Xhf_preemphasis);
X=X.';

%Generate 4 uniformly distributed random numbers between 0 and the Nyquist Frequency -
sort them in ascending order
Q = (sort(rand(1,5)));

% %Assign initial formant frequency values to each of the formants based on experimental
values (randn*std + mean)
Pitch_Initial = Q(1)*50 + 175;
F1 = Q(2)*115.9433 + 397.3253;
F2 = Q(3)*461.5834 + 1.49E+03;
F3 = Q(4)*381.7358 + 2.49E+03;
F4 = Q(5)*258.653 + 3.55E+03;

%Intialize F_freq
F_freq = [F1;F2;F3;F4];

%Check for any non-Nyquist and non-zero initial formant frequency assignments
F_freq_Bad = find( (F_freq <= 0) | (F_freq >= Fs/2) );

if (any(F_freq_Bad))

    %If the first formant is less then or equal to zero
    if (F_freq(1) <= 0)

        %Set the first formant to it's (MEAN - 2*STD)
        F1 = 397.3253 - 2*115.9433;

    end %endif (F_freq(1) <= 0)

    %If the fourth formant is more than or equal to the Nyquist freq.
    if (F_freq(4) >= Fs/2)

        %Set the fourth formant frequency to it's (MEAN + 2*STD)
        F4 = 3.55E03 + 2*258.653;

    end %endif (F_freq(4) >= Fs/2)

    F_freq = [F1;F2;F3;F4];

end %endif (any(F_freq_Bad))


%Initial Filter assignments
[B,A] = formantfilters(Fs, Pitch_Initial, [F_freq(1), F_freq(2), F_freq(3), F_freq(4)]);
B = B.';
A = [A.';zeros(3,4)]; %To equalize A into a 5 x 5 matrix as well (like B).

%Initialize the Avg. Formant Frequency
Avg_F_freq = [F1; F2; F3; F4];

%Initialize the Moving Average of the RMS Levels
Avg_Y_RMS = [RMS_Ratio_F1; RMS_Ratio_F2; RMS_Ratio_F3; RMS_Ratio_F4];
Avg_Y_RMS = repmat(Avg_Y_RMS,1,lpc_window_size+1);

%Set the previous formant frequency
Last_F_freq = F_freq;
```

```
%Order of the filter - doesn't change
[tmp, num_of_filters] = size(B);
filter_order =tmp-1;

%Zero Pad the input signal to deal with the initialization
Y = zeros(num_of_filters,size(X,2));
X_VD = [(zeros(1,filter_order)) X_VD];
X = [(zeros(1,filter_order)) X];
Y_temp = zeros(num_of_filters,size(X,2));

%Frequency movement indicators (they are only here to keep track of the movement)-
initialize
F1_Mov(1:lpc_window_size+1) = F1;
F2_Mov(1:lpc_window_size+1) = F2;
F3_Mov(1:lpc_window_size+1) = F3;
F4_Mov(1:lpc_window_size+1) = F4;

%%%%%%%%%%%%%%VOICING DETECTOR and GENDER DETECTOR INITIALIZATIONS%%%%%%%%%%%%%%%%%

%Initialize the return variable: 1 for voiced speech; and 0 for unvoiced speech
Voice = zeros(1,length(X_VD));

%Initialize the gender 0 for male and 1 for female and -1 for no change from last (hold
at previous)
Gender = zeros(1,length(X_VD));
Pitch = zeros(1,length(X_VD));

%Initialize the Avg. Pitch
Avg_Pitch = Pitch_Initial;

%Setup Jump counter to an inital value of 20 ms - Determines how often the Pitch/Gender
is checked
Jump_Counter = Fs/20;

%The HPF Parameters - 20th order high-pass Butterworth filter.
[B_HPF,A_HPF] = butter(20,Filter_Cutoff/(Fs/2), 'high');

%The LPF Parameters - 20th order low-pass Butterworth filter
[B_LPF,A_LPF] = butter(20,Filter_Cutoff/(Fs/2));

%Delay Filter Parameters
B_Delay = [zeros(1,10),1,zeros(1,10)];
A_Delay = 1;

%High-pass part - for log ratio calculation - Voicing Detector
X_HPF= filter(B_HPF,A_HPF,X_VD);

%Low-pass part - for log ration calculation - Voicing Detector
X_LPF= filter(B_LPF,A_LPF,X_VD);

%Delayed part - for Autocorrelation
X_Delayed= filter(B_Delay,A_Delay,X_VD);


%Setup the waitbar.
wb1 = waitbar(0,'Running...');
set(wb1,'name','Formant Tracker - 0%');


%Compute over entire signal
for n=filter_order+1:length(X)

    %Compute filtered signal for each filter (each formant)
    for c = 1:num_of_filters

        Y_temp(c,n) = [X(n:-1:(n-filter_order))]*[B(1:(filter_order+1),c)] - [Y_temp(c,n-
1:-1:(n-filter_order))]*[A(2:(filter_order+1),c)];
```

```
    end %inner loop endfor c = 1:num_of_filters

    %Calculate the RMS values (complex) of the filetred signals
    Y_RMS(:,n) = rms(Y_temp(:,max(n-(lpc_window_size-1),1):n).').';

    %%%%%%%%%%%%%%%%% VOICING DETECTOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Calculate Autocorrelation on a moving square window of the same size as the LPC
window - 20 ms
    if (n > lpc_window_size+1)
        %autocorrelation of the delayed signal on a moving 20 ms square window - after
LPC kicks in
        autocorr = xcorr(X_Delayed(n-(lpc_window_size-1):n));
    else
        %set the autocorrelation to zero for the length of the signal before the LPC
kicks in.
        autocorr = zeros(2*lpc_window_size-1,1);
    end %endif (n > lpc_window_size+1)

    %Calculate the logratio of the LPF and the HPF
    Log_ratio(n)          =          log((rms(X_LPF(max(1,          n-(lpc_window_size-
1)):n))/sqrt(Filter_Cutoff))/(rms(X_HPF(max(1,      n-(lpc_window_size-1)):n))/(sqrt(Fs/2-
Filter_Cutoff))));

    %- HYSTERISIS - Assign voiced/unvoiced to each of the data points in the sample
    %If the previous sample was unvoiced AND the current sample had a log ratio of MORE
than the
    %log ratio threshold1 (for switcing from 0 --> 1), then the current sample is voiced.
ie.
    %The switch from unvoiced to voiced occurs only if the log ratio threshold 1 is
crossed.
    if ((Voice(max(1,n-1)) == 0) & (Log_ratio(n) > Log_ratio_threshold1))

        %set current sample to be voiced speech
        Voice(n) = 1;

        %If the previous sample was voiced AND the current sample had a log ratio of LESS
than the
        %log ratio threshold2 (for switcing from 1 --> 0), then the current sample is
unvoiced. ie.
        %The switch from voiced to unvoiced occurs only if the log ratio threshold 2 is
crossed.
    elseif ((Voice(max(1,n-1)) == 1) & (Log_ratio(n) < -(Log_ratio_threshold2)))

        %set current sample to be unvoiced speech
        Voice(n) = 0;

        %If the log ratio threshold is NOT crossed then assign the current sample to be
like the last one
    else
        Voice(n) = Voice(max(1,n-1));

    end %endif Hysterisis.

    %Use the results for the autocorrelator AND the Hysterisis to make a final decisions
regarding voiced/unvoiced speech.
    %The Sample is voiced if the logratio says it voiced AND the the autocorrelation at
atleast one point in the window
    %is greater than 0.25 (Autocorrelation_Level) times the autocorrelation at the centre
of the window AND
    %there is at least one point in the window whose autocorrelation is greater than 0.

    Voice(n)        =        (Voice(n)        &        any(abs(autocorr([1:lpc_window_size-1
lpc_window_size+1:2*lpc_window_size-1]))                                          >=
Autocorrelation_Threshold_Level*autocorr(lpc_window_size)) & any(abs(autocorr) > 0));
```

```
    %%%%%%%%%%%%%%%%%%%%%%% GENDER DETECTOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Satisfy indexing and length requirements for the Gender Detector
    %Check Gender iff sample is voiced AND the check hasn't been done in a while
(determined by Jump_Counter)
    if ( (Voice(n) == 1) & (n >= Jump_Counter) )

        %Setup Jump counter to setup how often the pitch is going to be checked - every
20 ms
        Jump_Counter = Jump_Counter + (Fs/50);

        %Setup the windowed data to send to the Gender Detector
        X_GD = X_VD((n-399):n);

        %Call the Gender Detector and obtain pitch as well
        [Gender(n), Pitch(n)] = gender_detector(X_GD, Fs);

        %Round the Pitch to the closest integer
        Pitch(n) = round(Pitch(n));

        %Move Voicing Detector threshold levels depending on the result from the Gender
Detector - Males
        if (Gender(n) == 0)

            %It's MALE AND we need to change ANY one of the parameters any further
            if    (Filter_Cutoff    >    700)    |    (Log_ratio_threshold1    >    0.1)    |
(Log_ratio_threshold2 > 0.2) | (Autocorrelation_Threshold_Level < 0.6)

                %Change filter_cutoff gradually, if requied
                if (Filter_Cutoff > 700)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Filter_Cutoff = Filter_Cutoff - 10;

                    %If there has been a change in the parameters re-calculate the LPF
and the HPF for the Voicing Detector
                    %The HPF Parameters - 20th order high-pass Butterworth filter.
                    [B_HPF,A_HPF] = butter(20,Filter_Cutoff/(Fs/2), 'high');

                    %The LPF Parameters - 20th order low-pass Butterworth filter
                    [B_LPF,A_LPF] = butter(20,Filter_Cutoff/(Fs/2));

                    %High-pass part - for log ratio calculation - Voicing Detector
                    X_HPF= filter(B_HPF,A_HPF,X_VD);

                    %Low-pass part - for log ration calculation - Voicing Detector
                    X_LPF= filter(B_LPF,A_LPF,X_VD);

                end %endif (Filter_Cutoff > 700)

                %Change Log_ratio_threshold1 gradually, if requied
                if (Log_ratio_threshold1 > 0.1)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Log_ratio_threshold1 = Log_ratio_threshold1 - 0.0025;

                end %endif (Log_ratio_threshold1 > 0.1)

                %Change Log_ratio_threshold2 gradually, if requied
                if (Log_ratio_threshold2 > 0.2)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Log_ratio_threshold2 = Log_ratio_threshold2 - 0.0025;
```

```
                end %endif (Log_ratio_threshold2 > 0.2)

                %Change Autocorrelation_Threshold_Level gradually, if requied
                if (Autocorrelation_Threshold_Level < 0.6)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Autocorrelation_Threshold_Level  =  Autocorrelation_Threshold_Level  +
0.00875;

                end %endif (Autocorrelation_Threshold_Level < 0.6)

            end  %endif  (Filter_Cutoff  >  700)  |  (Log_ratio_threshold1  >  0.1)  |
(Log_ratio_threshold2 > 0.2) | (Autocorrelation_Threshold_Level < 0.6)

        elseif (Gender(n) == 1)%- Females

            %It's FEMALE AND we need to change ANY one of the parameters any further
            if   (Filter_Cutoff   <   1120)   |   (Log_ratio_threshold1   <   0.2)   |
(Log_ratio_threshold2 < 0.3) | (Autocorrelation_Threshold_Level > 0.25)

                %Change filter_cutoff gradually, if requied
                if (Filter_Cutoff < 1120)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Filter_Cutoff = Filter_Cutoff + 10;

                    %If there has been a change in the parameters re-calculate the LPF
and the HPF for teh Voicing Detector
                    %The HPF Parameters - 20th order high-pass Butterworth filter.
                    [B_HPF,A_HPF] = butter(20,Filter_Cutoff/(Fs/2), 'high');

                    %The LPF Parameters - 20th order low-pass Butterworth filter
                    [B_LPF,A_LPF] = butter(20,Filter_Cutoff/(Fs/2));

                    %High-pass part - for log ratio calculation - Voicing Detector
                    X_HPF= filter(B_HPF,A_HPF,X_VD);

                    %Low-pass part - for log ration calculation - Voicing Detector
                    X_LPF= filter(B_LPF,A_LPF,X_VD);

                end %endif (Filter_Cutoff < 1120)

                %Change Log_ratio_threshold1 gradually, if requied
                if (Log_ratio_threshold1 < 0.2)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Log_ratio_threshold1 = Log_ratio_threshold1 + 0.0025;

                end %endif (Log_ratio_threshold1 < 0.2)

                %Change Log_ratio_threshold2 gradually, if requied
                if (Log_ratio_threshold2 < 0.3)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
                    Log_ratio_threshold2 = Log_ratio_threshold2 + 0.0025;

                end %endif (Log_ratio_threshold2 < 0.3)

                %Change Autocorrelation_Threshold_Level gradually, if requied
                if (Autocorrelation_Threshold_Level > 0.25)

                    %Decay fast enough so that it can change from one end to the other in
40 ms
```

```
                    Autocorrelation_Threshold_Level = Autocorrelation_Threshold_Level -
0.00875;

                end %endif (Autocorrelation_Threshold_Level > 0.25)

            end %endif (Filter_Cutoff < 1120) | (Log_ratio_threshold1 < 0.2) |
(Log_ratio_threshold2 < 0.3) | (Autocorrelation_Threshold_Level > 0.25)

        end %endif (Gender == 0)

    end %endif ( (Voice(n) == 1) & (n >= Jump_Counter) )

    %%%%%%%%%%%%%%%%%%%%%% Pitch Calculations %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Smooth out the pitch - avoid it from going to zero and jumping around during voiced
speech
    if ( (Pitch(n) == 0) & (Voice(n) == 1) )

        %Hold Pitch at previous value
        Pitch(n) = Pitch(n-1);

    end %endif ( (Pitch(n) == 0) & (Voice(n) == 1) )

    %Set the Pitch to the moving average during unvoiced sections
    if ((Voice(n) == 0) )

            %Decay Pitch to a moving avg.
            Pitch(n) = Avg_Pitch;

    end %endif ( (Pitch(n) == 0) & (Voice(n) == 0) )

    %Update the moving average of the Pitch
    Avg_Pitch = ((((n-1) .* Avg_Pitch) + Pitch(n)) ./ n);


    %%%%%%%%%%%%%%%% Moving Average Calculations and LPC %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Calcualte as long as the last sample has not been reached
    if ((n > lpc_window_size+1) & (n <= length(X)))

        %If the entire window is voiced
        if(all(Voice(n-(lpc_window_size-1):n)) == 1)

            %1st order LPC formant region calculations ON WINDOWED DATA
            F_lpc = lpc (window.*Y_temp(:,(n-lpc_window_size+1):n).',1);

            %convert LPC values into formant frequencies
            F_freq = sort(angle(-F_lpc(:,2))/(2*pi))*Fs;

            %Check to see if any of the formant results from the LPC are invalid
frequencies
            if(isempty(find(isfinite(F_freq) ==0)) == 0)

                %Set all 'NaN' frequencies to the last valid Formant Frequency
                F_freq(find(isfinite(F_freq) == 0)) = Last_F_freq;

            end %endif (isempty(find(isfinite(F_freq) ==0)) == 0)

            %Deal with any Negative Frequencies by Decaying to Average
            if (any(F_freq < 0))

                %Find the Formant frequencies that are less than ZERO
                F_freq_Bad = find(F_freq < 0);

                %Decay Each of the Negative Frequencies
                F_freq(F_freq_Bad)          =          (Last_F_freq(F_freq_Bad)          -
(0.002*(Last_F_freq(F_freq_Bad) - Avg_F_freq(F_freq_Bad))));

            end %endif
```

```
            %Update the RMS_Thresholds based on a moving average (within set limits) if
the sentence is VOICED
            Avg_Y_RMS(:,n) = ((((n-1) .* Avg_Y_RMS(:,n-1)) + db(Y_RMS(:,n))) ./ n);


            %Check to see if any of the formant frequencies are below the Threshold
Levels
            %If the Formant Frequency is less then the threshold level (moving threshold
parameter - Avg_Y_RMS) then decay ONLY that formant
            if (db(Y_RMS(1,n)) < (Avg_Y_RMS(1,n) -6))

                F_freq(1) = Last_F_freq(1) - (0.002*(Last_F_freq(1) - Avg_F_freq(1)));

            end %endif (db(Y_RMS(1,n)) < Avg_Y_RMS(1,n))

            if ( (db(Y_RMS(2,n)) < (Avg_Y_RMS(2,n) -8)) | (abs(F_freq(2)-Last_F_freq(2))
> 900) )

                F_freq(2) = Last_F_freq(2) - (0.002*(Last_F_freq(2) - Avg_F_freq(2)));

            end %endif (db(Y_RMS(2,n)) < Avg_Y_RMS(1,n))

            if ( (db(Y_RMS(3,n)) < Avg_Y_RMS(3,n) - 10) | (abs(F_freq(3)-Last_F_freq(3))
> 900) )

                F_freq(3) = Last_F_freq(3) - (0.002*(Last_F_freq(3) - Avg_F_freq(3)));

            end %endif (db(Y_RMS(3,n)) < Avg_Y_RMS(3,n))

            if (db(Y_RMS(4,n)) < Avg_Y_RMS(4,n) - 14)

                F_freq(4) = Last_F_freq(4) - (0.002*(Last_F_freq(4) - Avg_F_freq(4)));

            end %endif (db(Y_RMS(4,n)) < Avg_Y_RMS(4,n))

            %Update the moving average Formant Frequency
            Avg_F_freq = ((((n-1) .* Avg_F_freq) + F_freq) ./ n);

        else %elseif(all(Voice(n-(lpc_window_size-1):n)) == 1) %If the entire window is
NOT voiced

            %Decay the Formant Frequency for ALL formants
            F_freq = (Last_F_freq - (0.002*(Last_F_freq - Avg_F_freq)));

            %Decay the Avg_Y_RMS value based on the current RMS values and the Average
RMS value.
            if (all(db(Y_RMS(:,n)) > (Avg_Y_RMS(:,n-1) - 5)))

                Avg_Y_RMS(:,n)    =    Avg_Y_RMS(:,n-1)    -    (0.002*(Avg_Y_RMS(:,n-1)    -
db(Y_RMS(:,n))));

            else

                Avg_Y_RMS(:,n) = Avg_Y_RMS(:,n-1);

            end %endif (all(db(Y_RMS(:,n)) > (Avg_Y_RMS(:,n-1) - 5)))

        end %endif (all(Voice(n-(lpc_window_size-1):n)) == 1)

        %Limit how close the formants are allowed to come to each other
        %F1 should not get closer than 150 Hz to the Pitch
        if (F_freq(1) < (Pitch(n)+150))

            F_freq(1) = F_freq(1) + 200;

        end %endif (F_freq(1) < (Pitch(n)+150))
```

```
        %F2 should not get closer than 300 Hz to F1
        if (F_freq(2) < (F_freq(1)+300))

            F_freq(2) = F_freq(2) + 400;

        end %endif (F_freq(2) < (F_freq(1)+300))

        %F3 should not get closer than 400 Hz to F2
        if (F_freq(3) < (F_freq(2)+400))

            F_freq(3) = F_freq(3) + 400;

        end %endif (F_freq(3) < (F_freq(2)+400))

        %F4 should not get closer than 500 Hz to F3
        if (F_freq(4) < (F_freq(3)+500))

            F_freq(4) = F_freq(4) + 400;

        end %endif (F_freq(4) < (F_freq(3)+500))

        %Set the previous formant frequency to the final assignment for the current
formant frequency
        Last_F_freq = F_freq;

        %Re-Calculate Formant Filter Parameters
        [B,A]   =   formantfilters(Fs,   Pitch(n),   [F_freq(1),   F_freq(2),   F_freq(3),
F_freq(4)]);
        B = B.';
        A = [A.';zeros(3,4)]; %To equalize A into a 4 x 4 matrix as well (like B).

        %re-assign Formant Frequency Tracking info. based on the Voicing Detector Info.
        F1 = F_freq(1);
        F2 = F_freq(2);
        F3 = F_freq(3);
        F4 = F_freq(4);

        %Frequency movement indicators - update
        F1_Mov(n) = F1;
        F2_Mov(n) = F2;
        F3_Mov(n) = F3;
        F4_Mov(n) = F4;

    end %end if (n > lpc_window_size+1)

    %UPDATE WHITE-BAR every 1%
    if (mod(n,(round(length(X_VD)/100))) == 0)
        waitbar(n/length(X_VD),wb1)
        set(wb1,'name',['Voicing Detector - ' sprintf('%2.1f',n/length(X_VD)*100) '%'])
    end

end %end for - outter loop

%Close Waitbar
close(wb1)

%Turn warnings back on
warning on;

% Remove the extra padded info. form the matrix
Y = Y_temp(:,(filter_order+1):end);
X = X(:,(filter_order+1):end);
X_VD = X_VD(:,(filter_order+1):end);
Voice = Voice(:,(filter_order+1):end);
F1_Mov = F1_Mov(:,(filter_order+1):end);
F2_Mov = F2_Mov(:,(filter_order+1):end);
```

```
F3_Mov = F3_Mov(:,(filter_order+1):end);
F4_Mov = F4_Mov(:,(filter_order+1):end);
Gender = Gender(:,(filter_order+1):end);
Pitch = Pitch(:,(filter_order+1):end);

%Formant Frequencies
F = [F1_Mov; F2_Mov; F3_Mov; F4_Mov];


keyboard

figure
subplot(5,1,1);
specgram(X,256,Fs,256,round(0.85*256))
[caxis_low_lim caxis_up_lim] = caxis;
caxis([caxis_up_lim-80 caxis_up_lim]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
title('\bf Spectrogram of the original signal');
subplot(5,2,1);
specgram(Y(1,:),256,Fs,256,round(0.85*256))
[caxis_low_lim caxis_up_lim] = caxis;
caxis([caxis_up_lim-80 caxis_up_lim]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
title('\bf Spectral region of estimation for the first formant filter');
subplot(5,3,1);
specgram(Y(2,:),256,Fs,256,round(0.85*256))
[caxis_low_lim caxis_up_lim] = caxis;
caxis([caxis_up_lim-80 caxis_up_lim]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
title('\bf Spectral region of estimation for the second formant filter');
subplot(5,4,1);
specgram(Y(3,:),256,Fs,256,round(0.85*256))
[caxis_low_lim caxis_up_lim] = caxis;
caxis([caxis_up_lim-80 caxis_up_lim]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
title('\bf Spectral region of estimation for the third formant filter');
subplot(5,5,1);
specgram(Y(4,:),256,Fs,256,round(0.85*256))
[caxis_low_lim caxis_up_lim] = caxis;
caxis([caxis_up_lim-80 caxis_up_lim]);
xlabel('\bf Time (s)');
ylabel('\bf Frequency (Hz)');
title('\bf Spectral region of estimation for the fourth formant filter');
```

Formant Tracking Filters

```
function [BT, AT] = formantfilters(Fs, Pitch, formant_frequencies)

% FORMANTFILTERS2
%
%   [BT, AT] = formantfilters(Fs, Pitch, formant_frequencies)
%
%
%   This function calculates and returns the filter coefficients, BT, AT for a set of 4
%   formant tracking filters.
%
%   Fs: Sampling Frequency of the signal to be filtered.
%
```

```
%    PITCH: The value in (Hz) of the Pitch, to be able to add an extra Zero in the F1
filter.
%
%    FORMANT_FREQUENCIES: Each component of this vector contains the locations (in Hz) of
the
%    formant frequency estimates (pole locations for each of the formant filters).
%
%    Each row of the filter coefficients returned (BT and AT) contain the values for one
%     Formant Tracking Filter, so BT and AT both have 4 rows - for 4 different tracking
filters.
%
%     THIS FUNCTION CALCULTAES AND RETURNS THE FILTER COEFFICIENTS OF 4 FORMANT TRACKING
FILTERS
%    SIMULTAENOUSLY!
%
%    See also FORMANTFILTER_COEFFICIENTS, MATRIXFILTER, FILTER, and FILTER2

% Inputs
%
% Fs: Sampling Frequency of the signal to be filtered
%
% PITCH: Location in (Hz) of the Pitch for the first Formant
%
% FORMANT_FREQUENCIES: Each component of this vector contains the locations (in Hz) of
the
% formant frequency estimates (pole locations for each of the formant filters).
%
%
%
% Outputs
% BT: Filter coefficients a 4 x 4 matrix - each row represents one filter
% AT: Filter coefficients a 4 x 2 matrix - each row represents one filter
%
% Author: Kamran Mustafa
% E-mail: mkamran@hotmail.com
%
% Modification List:
% March 25, 2002 - First Created
% October 21, 2002 - Started modifications to include an extra AZF at the location of the
Pitch in the F1 filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initialize the return variables
AT = zeros(4,2);
BT = zeros(4,5);

%Setup the general DTF parameters
Rp = 0.9;
K = 1-Rp; %The DC gain

%set-up the PITCH DTF pole at the location of the Pitch
Fk0 = Pitch;
P0 = Rp*exp(j*2*pi*Fk0/Fs); %Location of the pole - Pitch dependent

%set-up the 1st DTF pole at the first Formant Frequency location
Fk1 = formant_frequencies(1);
P1 = Rp*exp(j*2*pi*Fk1/Fs); %Location of the pole

%set-up the 2nd DTF pole at the second Formant Frequency location
Fk2 = formant_frequencies(2);
P2 = Rp*exp(j*2*pi*Fk2/Fs); %Location of the pole

%set-up the 3rd DTF pole at the third Formant Frequency location
Fk3 = formant_frequencies(3);
P3 = Rp*exp(j*2*pi*Fk3/Fs); %Location of the pole

%set-up the 4th DTF pole at the fourth Formant Frequency location
```

```
Fk4 = formant_frequencies(4);
P4 = Rp*exp(j*2*pi*Fk4/Fs); %Location of the pole


%set-up the 4 AZF Parameters
Rz = .98;
Fl0 = Pitch;
Fl1 = formant_frequencies(1);
Fl2 = formant_frequencies(2);
Fl3 = formant_frequencies(3);
Fl4 = formant_frequencies(4);

%set-up the Pitch dependent AZF
Z0 = [Rz*exp(j*2*pi*Fl0/Fs)]; %The single zero at the pitch location

%set-up the first AZF
Z1 = [Rz*exp(j*2*pi*Fl1/Fs)]; %The single zero location

%set-up the second AZF
Z2 = [Rz*exp(j*2*pi*Fl2/Fs)]; %The single zero location

%set-up the third AZF
Z3 = [Rz*exp(j*2*pi*Fl3/Fs)]; %The single zero location

%set-up the fourth AZF
Z4 = [Rz*exp(j*2*pi*Fl4/Fs)]; %The single zero location


%FIRST TRACKING FILTER COEFFICIENTS
%Construct the first formant tracking filter with a pole at the 1st formant frequency
estimate
%setup the appropriate all zero filter DC gain values
Kn0 = 1/(1-Rz*exp(j*2*pi*((Fl0-Fk1)/Fs))); %The zero filter DC gain - for the pitch
Kn2 = 1/(1-Rz*exp(j*2*pi*((Fl2-Fk1)/Fs))); %The zero filter DC gain
Kn3 = 1/(1-Rz*exp(j*2*pi*((Fl3-Fk1)/Fs))); %The zero filter DC gain
Kn4 = 1/(1-Rz*exp(j*2*pi*((Fl4-Fk1)/Fs))); %The zero filter DC gain
%The overall Filter Function
PT1 = P1; %pole location
ZT1 = [Z0; Z2; Z3; Z4]; %combine the zero locations
KT1 = K.*Kn0.*Kn2.*Kn3.*Kn4; %combine the DC gain terms of the AZF and the DTF
%Obtain the filter coefficients for the 1st filter
[BT(1,:), AT(1,:)]   = zp2tf_complex (ZT1,PT1,KT1); %Final filter values at this pole
location


%SECOND TRACKING FILTER COEFFICIENTS
%Construct the second formant tracking filter with a pole at the 2nd formant frequency
estimate
%setup the appropriate all zero filter DC gain values
Kn1 = 1/(1-Rz*exp(j*2*pi*((Fl1-Fk2)/Fs))); %The zero filter DC gain
Kn3 = 1/(1-Rz*exp(j*2*pi*((Fl3-Fk2)/Fs))); %The zero filter DC gain
Kn4 = 1/(1-Rz*exp(j*2*pi*((Fl4-Fk2)/Fs))); %The zero filter DC gain
%The overall Filter Function
PT2 = P2; %pole location
ZT2 = [Z1; Z3; Z4]; %combine the zero locations
KT2 = K.*Kn1.*Kn3.*Kn4; %combine the DC gain terms of the AZF and the DTF
%Obtain the filter coefficients for the 2nd filter
[BT(2,1:4), AT(2,:)]   = zp2tf_complex (ZT2,PT2,KT2); %Final filter values at this pole
location


%THIRD TRACKING FILTER COEFFICIENTS
%Construct the third formant tracking filter with a pole at the 3rd formant frequency
estimate
%setup the appropriate all zero filter DC gain values
Kn1 = 1/(1-Rz*exp(j*2*pi*((Fl1-Fk3)/Fs))); %The zero filter DC gain
Kn2 = 1/(1-Rz*exp(j*2*pi*((Fl2-Fk3)/Fs))); %The zero filter DC gain
```

```
Kn4 = 1/(1-Rz*exp(j*2*pi*((Fl4-Fk3)/Fs))); %The zero filter DC gain
%The overall Filter Function
PT3 = P3; %pole location
ZT3 = [Z1; Z2; Z4]; %combine the zero locations
KT3 = K.*Kn1.*Kn2.*Kn4; %combine the DC gain terms of the AZF and the DTF
%Obtain the filter coefficients for the 3rd filter
[BT(3,1:4), AT(3,:)]   = zp2tf_complex (ZT3,PT3,KT3); %Final filter values at this pole
location


%FOURTH TRACKING FILTER COEFFICIENTS
%Construct the fourth formant tracking filter with a pole at the 4th formant frequency
estimate
%setup the appropriate all zero filter DC gain values
Kn1 = 1/(1-Rz*exp(j*2*pi*((Fl1-Fk4)/Fs))); %The zero filter DC gain
Kn2 = 1/(1-Rz*exp(j*2*pi*((Fl2-Fk4)/Fs))); %The zero filter DC gain
Kn3 = 1/(1-Rz*exp(j*2*pi*((Fl3-Fk4)/Fs))); %The zero filter DC gain
%The overall Filter Function
PT4 = P4; %pole location
ZT4 = [Z1; Z2; Z3]; %combine the zero locations
KT4 = K.*Kn1.*Kn2.*Kn3; %combine the DC gain terms of the AZF and the DTF
%Obtain the filter coefficients for the 4th filter
[BT(4,1:4), AT(4,:)]   = zp2tf_complex (ZT4,PT4,KT4); %Final filter values at this pole
location
```

Gender Detector

```
function [gender, avgF0] = gender_detector (X,Fs)

% GENDER_DETECTOR
%
%  gender = gender_detector(X,Fs)
%
%    This function will use a pitch detection algorithm to decide if the speaker is
MALE(0) or FEMALE (1).
%   It is designed to work with short speech samples (up to or greater than 50 ms). The
function returns a
%   '0' if X contains male speech and a '1' if it contains female speech.
%
%   X is the speech sample and Fs is the sampling frequency.
%
%    Note that the function uses an average pitch based approach, where the pitch is
calculated using an autocorrelation
%    based method (with centre clipping and median filtering). The method is similar to
the Pitch estimation algorithm used by
%   Philip Loizou (loizou@utdallas.edu) in COLEA.
%
%   See also PITCH_DETECTOR and TEST_PITCH_DETECTOR.


% Inputs
% X: Speech Signal
% Fs: Sampling Frequency
%
% Outputs
% gender: Gender of the speaker in speech sample X
%
% Author: Kamran Mustafa
% E-mail: mkamran@hotmail.com  or  mkamran@ieee.org
%
% Modification List:
%
% September 17, 2002 - First Created
```

```
% September 18, 2002 - Adapted to work with the Formant tracker over one window length of
50 ms
% September 25, 2002 - Modified the function to be able to work independently and retrun
gender values instead of pitch est.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Turn Warnings off
warning off;

%Figure out the number of samples in X
n_samples = length(X);

% Window update rate - Window spacing - 5 ms
updRate=floor(5*Fs/1000);

%Window size - 20 ms
fRate=floor(20*Fs/1000); %Use a 20 ms window

%Number of frames in this sample
nFrames=floor(n_samples/updRate)-1;

%Initialize variables
avgF0=0;
f01=zeros(1,nFrames);
k=1;
m=1;


%Calculate over all the frames in the sample
for t=1:nFrames

    %Make sure that the pitch estimations don't run over the index limits of the sample
    if (k+fRate-1) > length(X)

        %Select the window (20 ms) over which to do the pitch estimation
        X_Win = X((length(X)-fRate):length(X));

    else

        %Select the window (20 ms) over which to do the pitch estimation
        X_Win= X(k:k+fRate-1);

    end %endif (k+fRate-1) > length(X)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  Do  the  Pitch  Estimation  over  one  frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Remove the DC bias
    X_Win = X_Win - mean(X_Win);

    %LPF the speech at 900 Hz to remove upper freq. (since Pitch info. will be below 900
Hz)
    [bf0,af0]=butter(4,900/(Fs/2));
    X_Win = filter(bf0,af0,X_Win);

    %Perform Centre Clipping and find the Clipping Level (CL)
    i13=fRate/3;
    maxi1=max(abs(X_Win(1:i13)));

    i23=2*fRate/3;
    maxi2=max(abs(X_Win(i23:fRate)));

    %Choose the appropriate clipping level
    if maxi1>maxi2
        CL=0.68*maxi2;
    else
        CL= 0.68*maxi1;
```

```
    end %endif maxi1>maxi2

    %Perform Center clipping
    clip=zeros(fRate,1);
    ind1=find(X_Win>=CL);
    clip(ind1)=X_Win(ind1)-CL;

    ind2=find(X_Win <= -CL);
    clip(ind2)=X_Win(ind2)+CL;

    engy=norm(clip,2)^2;

    %Compute the autocorrelation
    RR=xcorr(clip);
    g=fRate;


    %The pitch estimates are limited to being between 60 and 320 Hz
    %Find the max autocorrelation in the range 60 Hz <= F0 <= 320 Hz
    LF=floor(Fs/320);
    HF=floor(Fs/60);

    Rxx=abs(RR(g+LF:g+HF));
    [rmax, imax]= max(Rxx);
    imax=imax+LF;

    %Estimate raw pitch
    pitch=Fs/imax;

    %Check max RR against V/UV threshold
    silence=0.4*engy;

    %Make Voiced/Unvoiced descions
    if (rmax > silence)  & (pitch > 60) & (pitch <=320)
        pitch=Fs/imax;
    else
        % It is a unvoiced segment
        pitch=0;
    end %endif (rmax > silence)  & (pitch > 60) & (pitch <=320)

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  End  of  Pitch  Estimation  per
Window %%%%%%%%%%%%%%%%%%%%%%%%%

    %Assign Pitch estimate
    f0(t)=pitch;

    % Do median filtering
    if t>2 & nFrames>3

        z=f0(t-2:t);
        md=median(z);
        f01(t-2)=md;

        if md > 0
            avgF0=avgF0+md;
            m=m+1;
        end %endif md > 0

    elseif nFrames<=3

        disp('# of frames less than 3');
        f01(t)=pitch;
        avgF0=avgF0+pitch;
        m=m+1;

    end %endif t>2 & nFrames>3
```

```
    %Put next window 'updRate' appart from where the last one was.
    k=k+updRate;

end %endfor t=1:nFrames


%Calculate the avg. pitch estimate for the whole sample X.
if m==1
    avgF0=0;
else
    avgF0=avgF0/(m-1);
end %endif m==1



%Find Gender (The pitch being used is the avg. pitch - avgF0)
if (avgF0 >= 180)

    %It's female
    gender = 1;

else

    %It's male
    gender = 0;

end %endif

%Turn Warnings back on
warning on;
```