Performance Analysis of LPC and MFCC Techniques in Automatic Speech Recognition

A Thesis submitted in partial fulfilment of the requirements for the award of *Degree of*

Master of Engineering

In

Software Engineering

Submitted By

Manish Kumar

(Roll No. 801331013)

Under the supervision of

Ms. Ashima Singh

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

MAY 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled "Performance Analysis of LPC and MFCC Techniques in Automatic Speech Recognition" in partial fulfillment of award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala. is an authentic record of my own work carried out under the supervision of Ms. Ashima Singh Assistant Professor Computer Science and Engineering Department of Thapar University ,Patiala.

DATE

Manish Kumar
ROLL NO: 801331013

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Ms. Ashima Singh
Assistant Professor
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University ,Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENTS

My sincere thanks to all the people around me who helped me in completing this thesis work. First I wish to thank **Ms. Ashima Singh** (Assistant Professor), Computer Science and Engineering Department of **Thapar University , Patiala** for giving me an opportunity to work under her guidance . Her continued support, guidance and vision helped me to complete this thesis. It has been a pleasure working under her guidance.

I also express my sincere gratitude to all the faculty member of **Thapar University, Patiala** equipping me with the best of knowledge and providing me top class facilities and infrastructure.

Manish Kumar

Date:

Place: Thapar University, Patiala.

Manish Kumar

(801331013)

Abstract

The feature extraction is one of the most important issues in the field of speech recognition. There are two important measurements of speech signal. One is the parametric modeling approach, which is developed by human vocal tract that produces the corresponding speech sound. Generally it is derived from Linear Predictive analysis, such as Linear Prediction Cepstral (LPC) based cepstrum (LPCC). Another approach is non-parametric the modeling method that is based on the human auditory perception system. Mel-frequency cepstral coefficients (MFCCs) are utilized for this purpose. LPC is a technique used in the most of the speech recognition system to estimate the speech parameters like pitch and spectral envelope of the speech signals, which are used in linear predictive (LP) model. A brief survey of LPC and MFCC is initiate with modern phonetics and continuing through the current state of Large-Vocabulary Continuous Speech Recognition (LVCSR) . Experiments have been happening by help of MatLab and matlab tool for isolated word speech recognition in different environment. In the experiment we used different recognition algorithm and convert test data to trained data for better speech recognisation .Testing the data from two different vocabularies. Data is collected and recorded with different female and male voices. We have implemented LPC and MFCC algorithms applied on different type of wave (noise or without noise) and Analysis pulse positions, gain and error signal. We are trying to minimise the error and identified batter algorithms. LPC coefficients (LPCC) also estimated by applying some procedures on the speech signal. These procedures started with applying autocorrelation on the windowed frames and windowed frame is auto correlated by p`th order using by MATLAB.

Table of Contents

LIST OF TABLE

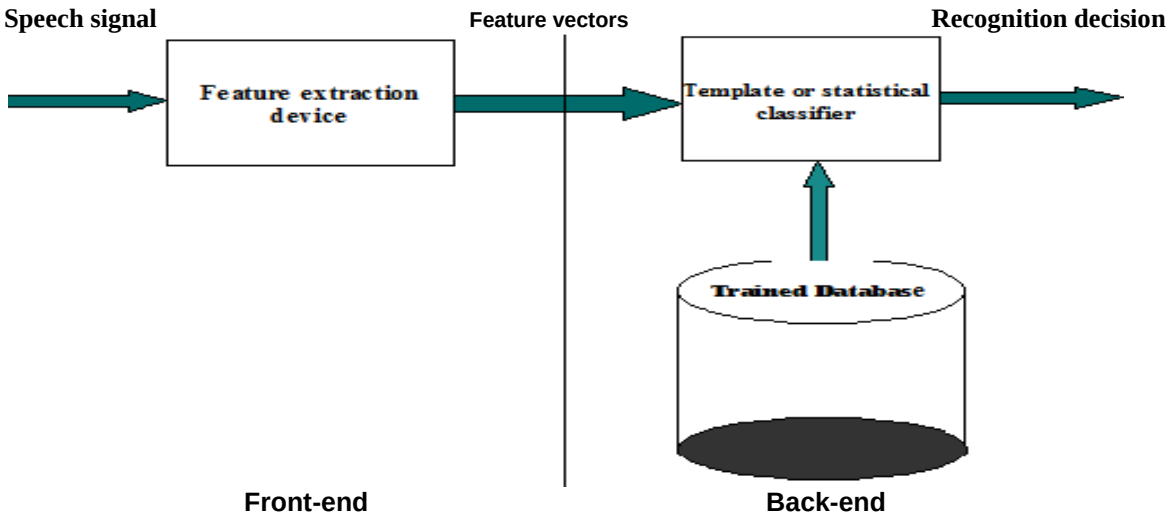
LIST OF FIGURES

Chapter 1

INTRODUCTION

Speech recognition system automatically recognizing whatever we are speaking on the basis of important information included in different sequence speech waves. This technique help us verify the identity of user and controlled access to different services such as Call Center voice dialling, Telephone Conversation in banking , telephone shopping, Robotic System, database access services , voice mail, control for confidential areas, and remote access [1].

Generally, a speech recogniser can be divided into two parts: front-end and back-end. The front-end of the recogniser is used for feature extraction device, while the back-end consists of a template or statistical classifier and a well-trained database. The following figure described the structure of a typical speech recogniser.



.

Figure 1.1 Speech recogniser Structure

The main task of the front-end is to observed features from a speech signal

Voice conversion have two phases first is training phase and second is conversion phase. In the training phase data are collected and processed or first one is referred to the enrolment, while the second one is referred to as the operational or testing phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or

train a reference model for that speaker. The second type of speech reorganisation, back-end is a statistical classifier. Statistical modeling method also called Hidden Markov Modeling (HMM) [9] for recognition. Basically, statistical models of phonemes are built during training. The classifier modifies the statistical parameters of these models when the system is trained with training utterances (set). After training, the statistical models are stored in the database. The classifier are decides the most likely sequence of words or phonemes by calculates the maximum likelihood based on the models. The front-end is basically a feature extraction module, while the back-end is primarily a classifier [10]. The back-end can either be a template matching classifier (DTW) or a statistical classifier (HMM). Back-end function is to make accurate decisions based on the speech features those are extracted by the front-end. so, the quality of a front-end algo perform important role in speech recognition systems. Many front-end algorithms have already been suggested by project and researchers. Some of the generally used feature extraction algorithms include,(LPCC) Linear Prediction Cepstral Coefficients , Perceptual Linear Prediction Coefficients (PLP) and Mel Frequency Cepstral Coefficients (MFCC). These algorithms good work in clean environment. However, they suffer severe performance decreased in noisy conditions, especially when there is a noise level not match between the training and testing environments

1.2.2 General Structure of feature extraction using LPC

The general structure of the speech recognition system is shown in figure 1.1. The input of the speech reorganising system is speech signal. After receiving signal the pre-processing of the speech is started. The pre-processing includes analysis and End points detection for this operation. After pre-processing the speech signal goes to the feature extraction block. In this paper, LP based features used like as LPC and LPCC have been considered. These features are used in next step for input into the neural network in the next block for classification. Finally the decision is taken if there is matching or not in the last decision block.

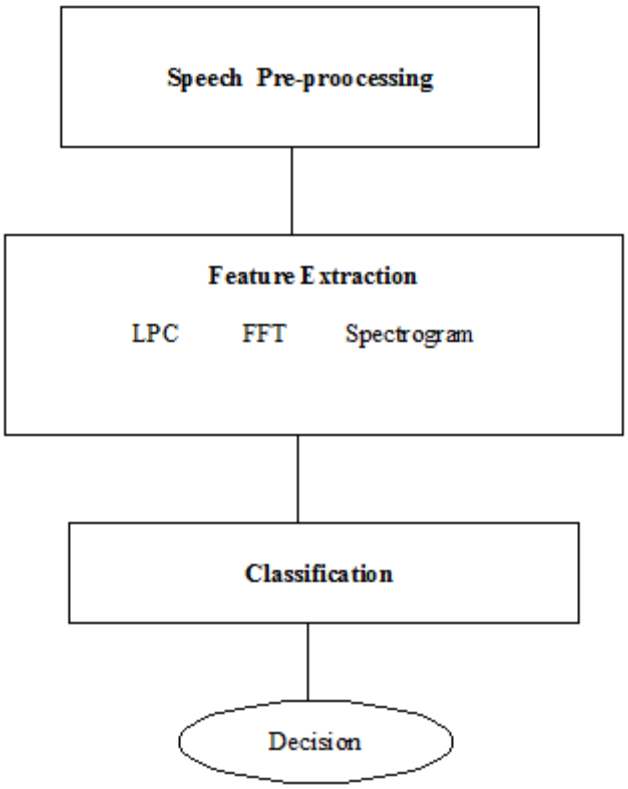


Figure 1.2 general structures for the LPC program

Chapter 2

Literature Survey

2.1 History

Beginning in the before 1980s, there had been a large curious in the use of feature sets that are analysis by computational models of the auditory periphery, generally based on physiological measurements of the responses of individual bers of the auditory nerve (Oded Ghitza 1986, Richard Lyon 1982, Stephanie Seneff 1986). John Makhoul (1975) gave an exposition of linear prediction in the analysis of discrete signals. The signal was modelled as a linear combination of its past values and present values of a hypothetical input to a system whose result was the given signal. Cepstral coefficients derived from linear prediction(LP) known as the (LPCC) Linear Prediction Cepstral Coefficients (LPCCs) (John Makhoul 1975, Bishnu Atal 1974,) . The LPCC featuren extraction used an all pole filter to model the vocal tract (human) with speech formants found by the poles of the all-pole filter. The model parameters were analysed by a least squares analysis based on time domain then derived in the frequency domain. The output spectral matching formulation analysed by modeling of selected part of a spectrum, for any spectral shaping in the frequency domain and we can modeling of continuous as well as discrete spectra. Linear Prediction Cepstral Coefficients features worked well in a clean environment. However, the linear predictive (LP) spectral envelope showed large spectral distortion in noisy environments (Qi Li et al. 2000, 2001). This resulted in significant performance degradation MFCC , the generally parameter use in speech recognition overall the advantages of the Cepstrum with a scale based frequency on critical bands (Paul Mermelstein and Steven Davis 1980). The Mel-frequency cepstra had a significant advantage over the linear frequency cepstral. Specifically, MFCC allowed better suppression of insignificant spectralvariation in the higher frequency bands. Through experiments , we can analysis several parametric representations of the speech signal were compared with according to word recognition performance in a syllable-oriented speech recognition system. The word templates were generated using an efficient, dynamic warping method and test data were time

Registered with the templates. A group of ten MFCC result found each 6.4 ms .result give better performance, and 95 percent or 96.5 percent recognition with each of two speakers. The great performance of Mel Frequency Cepstral Coefficients (MFCC) may be attributed to give that better output the perceptually relevant of the short-term speech spectrum. Even, like the Linear Prediction Cepstral Coefficients features, the MFCC features performed better in without noise environments, but not mismatched training and not in adverse environments or and testing conditions. Qi Li et al. (2000) was presented An auditory feature extraction algo for robust speech recognition in adverse acoustic environments. The feature analysis was comprised of an outer-middle-ear transfer function, FFT, conversion from linear to the Bark scale, nonlinearity, auditory filtering, and DCT. The feature was evaluated in two tasks: large vocabulary and connected-digit recognition in continuous speech recognition. The tested data were analysis under various noise conditions, including hands-free speech and handset data in wireless and landline communications with additive car and babble noise. While comparing with the MFCC, LPCC, MELLPCC, and PLP features, the auditory feature extraction algo has an average of 20 to 30 percentages of string error rate reduced by connected digit task, and of 8 to 14 percentage of word error rate reduced on the Wall Street Journal task in various additive noise conditions.

Kirandeep Kaur and Neelu Jain: Different techniques for the system have been discussed such as MFCC, LPC, LPCC, Wavelet break for feature extraction and VQ, DTW, HMM ,GMM, SVM, for feature Classification. All these techniques are also analysis with each other to find out best suitable among them. On the basis of the comparison done, MFCC has upper edge techniques for feature extraction as it is more consistent with speech recognisation. GMM comes out to be the best among classification models due to its good less memory usage and classification accuracy. Different feature extraction technique LPC Modal by all pole modal used this principle then output gets based on basic principle of different sound production, performance decreased in presence of noise [14]. Cepstral coefficients based on FFT principle than found tha result due to analysis Not much consistent with speech recognisation (human hearing) due to representation by linearly spaced filters [15].LPCC Modal by all pole modal analysis this principle then output gets Gives smoother

spectral envelope and stable representation as compared to LPC [15], drawback due to linearly spaced frequency bands.MFCC used Filter bank coefficients and get output More information about lower frequencies than higher frequencies due to mel spaced filter banks hence behaves more like a human ear as compared to other techniques , based on STFT which has fixed time-frequency resolution [16].

LIA / CERI and STEPMIND: He presented several methods for mapping speech recognition engine. That is required for mobile phone resource. The techniques are found the result digit recognition task using both English corpora and French. Analysis shows that speaker dependent lexicon outperforms strongly with respect to speaker independent one (1.69% of WER to compare to 5.60% of WER for 128. They show that LPC cepstral coefficients issued from the cellular phone embedded LPC algo get a better performance and cost ratio. Filter-bankbased parameters (MFCC and PLP) are outperforming significantly LPC and LPCC parameterization. His experiments shows that reducing the Gaussians per state number from 128 to1, them increase of WER remains less than 4%, in the specific context of less vocabulary and isolated word tasks Besides, approach based on GMM mapping obtains better results for very compact models .Never the small , this method was tested on speech-to-text conversion , and must be evaluated on isolated digit recognition task At last, they plan to explore various GMM learning strategies and adaptation methods.

Taabish Gulzar, Anand Singh and Sandeep Sharma:

They are show Mel Frequency Cepstral Coefficient (MFCC), Linear Prediction Cepstral Coefficient (LPCC), and Bark frequency Cepstral coefficient (BFCC) feature extraction techniques for recognition of Hindi Isolated, Paired and Hybrid words have been analysed and the corresponding recognition rates are compared. Artifical Neural Network (ANN) is used as back end processor. The experimental results show that the good recognition rate is obtained for MFCC as compared to LPCC and BFCC for all the three types of words .This surveys a comparative analysis of LPCC, MFCC and BFCC as feature extraction techniques and classifier as ANN. Hindi Isolated, Paired and Hybrid words are used for large database purpose. Experimental results demonstrate that MFCC shows better recognition rate with 99.78% for Isolated, 99.88% for paired and 99.82% for Hybrid words. LPCC and BFCC shows the

recognition rates of 95.82%, 97.02%, 96.62% and 95.68%, 95.56%, 97.62% for Isolated, Paired and Hybrid words respectively. Analyses from the group of experiments show that MFCC perform better than the conventional LPCC and BFCC method.

2.3 REVIEW OF AUDITORY PERCEPTUAL FEATURES AND SPEECH RECOGNITION

Reference/year	**Recognition process**	**Methodology perceptual processing**	**Results**	**Comments**
Tchorz *et al. [17]/ 1999*	Speaker independent, isolated word recognize using a HMM with additive noise.	Gamma tone filter bank, half wave rectification,	Adaptation system more robust in noise than mel scale cepstral eatures.	It was identified that the adaptive compare stage was the most important Processing stage.
M.Holm-berg et al.[18]/ 2006	Aurora 2 and 3 (added noise) with continuous density HMM recognition	Simplified synaptic short-term adaptation introduced in MFCC by high-pass first order IIR filter.	Improved ASR per - formance compared to baseline MFCC, MFCC with RASTA, CMS, Weiner filtering.	A filter time constant of 240 ms for adapta- tion was found appropriate for ASR.

D. S. Kim *et al.[19]/ 1999*	Isolated 50 Korean words using discrete density HMM and 256 word code-books.	Auditory model for speech recognition, FIR filterbank,single-level EIH, peak detector.	Improved robustness in various types of noise compared to LPCC, MFCC, PLP, EIHC.	ZCPA is more robust in white noise than in real-world noise types.
B. Strope et.al.[20]/ 1997	Isolated digits DTW (Itakura path constraint) and HMM recognizer with forced Viterbi alignment.	Dynamic model with a logarithmic adaptation stage (AGC) based on forward masking data.	Improvements in robustness to background noise Compared to the MFCC, LPC and RASTA based front-ends.	-
C. R. Jankowski et.al.[21]/1995	Tested on TI-105 isolated word database, HMM recognizer in clean and babble noise.	GSD and EIH auditory models compared with mel filter bank based cepstral front end.	The auditory model provided error rates as much as 4 % lower Than a mel filter bank.	The study also showed that mel filter bank cepstra outperformed LPC spectra.

Table 2.3 Review table

In this method the signal is observed by estimating the formants. We can change the effects of formants from the signal, and analysis the intensity and frequency from remaining buzz. This phenomenon is called inverse filtering, and the rest of signal is called the residue. In LPC method, sample of the signal is communicated as a linear combination of the previous samples. This equation is called a linear predictor equation and so, it is called as linear predictive coding. These formants are differentiating by the coefficients of the difference equation (prediction coefficients) [4].

We have used LP Residual (LPR) technique , these are contains mostly info about excitation source for emotions recognizing in speech signal. Signal is obtained from speech after extracting the vocal tract characteristics. Linear predictive coding method for speech synthesis and speech analysis is based on Vocal tract modelling the as a linear Pole filter having the system transfer function [5]

$$H(z) = \frac{G}{1 - \sum a_k z^{-k}}$$

H (z) = System transfer function

P = No of poles;

G = Filter Gain;

a[k] = Determine the pole , k=parameters

3.1 Identification of model voiced and unvoiced

There are two ways to identify model voiced and unvoiced speech sounds. According to short time-basis analysis, voiced speech is considered as periodic with a basic frequency of FO, and a period of pitch is 1/Fo, that is depends on the speaker. Hence, speech sound is generated by exciting and the polls filter model by a periodic impulse train. Or we can say that, unvoiced sounds are produce by exciting the all-pole filter by the o/p of a randomly noise generator. The basic difference between these two types of sounds comes from the way

that is produced. The vibrations of the vocal cords produce different voiced sounds. Because of rate at vocal cords vibrate identify the pitch of sound. On the other hand, unvoiced part of sounds does not identify the vibration of the vocal cords. The unvoiced sounds are created by the created by the vocal tract. If vocal cords until open so that constrictions of the vocal tract force air out to created the unvoiced sounds. For this reason use a short segment of a speech signal, lets say about 10 ms or 160 samples and sampling rate 16 KHz, the speech encoder at the transmitter **must** have some proper excitation function, the period of pitch for the voice speech, the gain, and the coefficients $a_p(k)$. [6] The parameters of the model are analysis data and modelled into a binary sequence form and transmitted data to the receiver. From the receiver side, the receive signal is the synthesized from the excitation signal and model. The parameters of the pole filter model are occupied from the speech data by means of linear prediction feature.

$$\hat{s}(n) = -\sum_{k=1}^{p} a_p(k)s(n-k)$$

$\hat{s}(n)$ =predicted value

S(n)=observed value

And the corresponding error between the observed by this equation -

$$e(n) = s(n) - \hat{s}(n)$$

Minimizing the sum of the squared error we want to identify the pole parameters $a_p(k)$. The differentiating the result we can sum above with respect to each of the parameters and equation the result to zero, is a Sep of p linear equations

$$\sum_{k=1}^{p} a_p(k)r_{ss}(m-k) = -r_{ss(m)}$$ Where m=1, 2……p

Where $r_{ss(m)}$ show the autocorrelation of the sequence $s(n)$.

$r_{ss(m)} = \sum_{n=0}^{N} s(n)s(n+m)$ The above equation can be represent as in matrix form as

$$R_{ss}a = -r_{ss(m)}$$

Where $R_{ss}a$ is a pxp autocorrelation matrix, r_{ss} is a px1 autocorrelation vector, and a is a px1 vector of model parameters. The gain parameter of the filter can be observed by the input-output relationship as follow

$$s(n) = -\sum_{k=1}^{p} a_p(k)s(n-k) + Gx(n)$$

where X (n) represent the input sequence. We can further manipulate this equation and in terms of the error sequence we have

$$Gx(n) = s(n) + \sum_{k=1}^{p} a_p(k)s(n-k) = e(n)$$ Then

$$G^2 \sum_{n=0}^{N-1} x^2(n) = \sum_{n=0}^{N-1} e^2(n)$$

Input excitation is normalized to unit energy by design, then

$$G^2 \sum_{n=0}^{N-1} x^2(n) = \sum_{n=0}^{N-1} e^2(n) = r_{ss}(0) + \sum_{k=1}^{p} a_p(k)r_{ss}(k)$$

Where G^2 is set equal to the residual energy resulting from the least square optimization. If we identify the result LPC coefficients, we can easily determine any frame is voiced, and if it is indeed voiced sound, then what is the pitch. We can determine the pitch by computing the following sequence in matlab.

$$r_e(n) = \sum_{k=1}^{p} r_a(k) r_{ss}(n-k)$$ Where $r_a(k)$ is defined as follow

$$r_a(n) = \sum_{k=1}^{p} a_a(k) a_p(i+k)$$

Which is defined as the autocorrelation sequence of the prediction coefficients. The pitch id detected by finding the peak of the normalized sequence

$\frac{r_e(n)}{r_e(0)}$ In the time interval corresponds to 3 to 15 ms in the 20ms sampling frame. If the value of this peak is at least 0.25, the frame of speech is considered voiced with a pitch period equal to the value of $n = N_p$, where $\frac{r_e(N_p)}{r_e(0)}$ is a maximum value.

If the peak value is less than 0.25, the frame speech is considered unvoiced and the pitch would equal to zero. The coefficients value of the LPC, the pitch period, and the type of excitation are then transmitted to the receiver. The decoder synthesizes the speech signal by passing the proper excitation through the all pole filter model of the vocal tract.

Chapter 4

LPC Methods and Estimation Issues

4.1 Definition of LPC method

Linear prediction methods are the most widely used in, speech synthesis, speech coding, speech recognition, speaker recognition and verification and for large speech storage. LPC methods provide accurate estimates of speech parameters, and does it extremely efficiently. The idea of Linear Prediction: current speech sample can be closely approximated as a *linear* combination of past samples [22].

4.2 LPC Methods

1. Covariance method
2. Autocorrelation method
3. Lattice method
4. Inverse filter formulation
5. Spectral estimation formulation
6. Maximum likelihood method
7. Inner product method

4.2.1. Autocorrelation Method signal is windowed by a tapering window in order to minimize discontinuities at beginning (predicting speech from zero-valued samples) and end (predicting zero-valued samples from speech samples) of the interval the matrix $\phi_{n(i,k)}$ is shown to be an autocorrelation function, the resulting autocorrelation matrix can be readily solved using standard matrix solutions.

4.2.2. Covariance method the signal is extended by p samples outside the normal range of $0 \leq m \leq L-1$ to include p samples occurring prior to m=0 (they are available) and eliminates the need for a tapering window; resulting matrix of correlations is symmetric different method of solution with somewhat different set of optimal prediction coefficients, $\{a_k\}$.

4.2.3. Lattice Method

Covariance and autocorrelation methods use two step solutions

1. computation of a matrix of correlation values
2. efficient solution of a set of linear equations

another class of LP methods called lattice methods, has evolved in which the two steps are combined into a recursive algorithm for determining LP parameters begin with Durbin algorithm--at the *i th* stage the set of coefficients $\{a_j^{(i)}, j = 1,2, \ldots . . i\}$ are coefficients of the *i th* order optimum LP .

4.3 PLP based Analysis

PLP analysis differs from LPC analysis in the sense that we approximate an auditory Spectrum by the spectrum of an all-pole model. This auditory spectrum differs from the power spectrum in the sense that we use a nonlinear frequency axis, that we do a critical band analysis with asymmetric weighting coefficients (with low-frequency slopes less steep than high-frequency ones). Also the idea of the non equal sensitivity of hearing at different frequencies and the intensity- loudness power law is included in this more perceptually based LP analysis.

4.4 RASTA Analysis

RASTA PLP, which is an extension of the previously described PLP analysis, applies an IIR filtering on the logarithm of the critical band spectrum. The IIR Filter is equivalent to a derivative-reintegration process as to filter out the long-term spectral tilts due to the tele-phone lines. After the two psycho acoustical steps the inverse logarithm is taken, followed by the "traditional" all pole modeling and cepstral recursion.

5.1 LPCC

Linear Prediction Cepstral Coefficients (LPCC) has been generally used in many speech recognition applications and speech analysis for many years. The idea behind LPCC is to model the human vocal tract by digitalize all-pole filter. The following block diagram show the process for analyse the LPCC feature vectors.

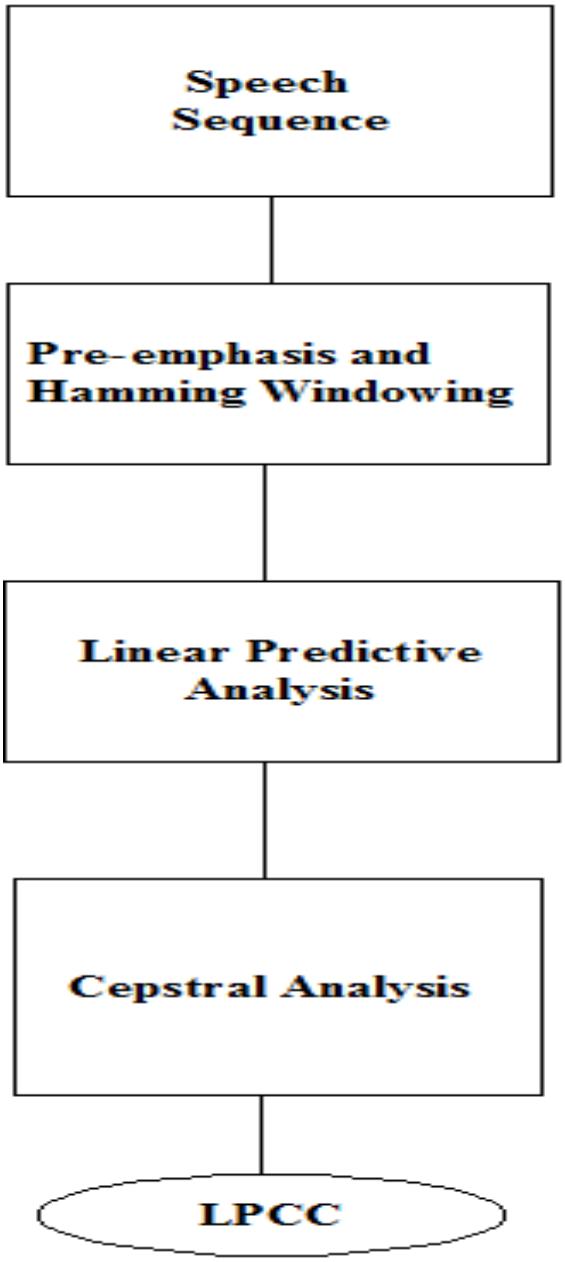


FIGURE 5.1 Block diagram of the LPCC processor

5.2 Pre-emphasis and Hamming windowing

The first step of the lpcc algorithm is pre-emphasis.so,The idea of pre-emphasis is to spectrally flatten the speech signal and equalise the inherent spectral tilt in speech [7].

Pre-emphasis is used by a first order FIR digital filter. The below equation shows that the transfer function of the pre-emphasis digital filter, (Equation -1)

$$\boldsymbol{H_P(Z) = 1 - az^{-1}}$$

Where *a* is a constant, which value is, a= 0.97.

After pre-emphasis, the signal is subdivided into frames. This process is the same as multiplying the entire speech sequence by a windowing function.

$$\boldsymbol{S_m[n]=s[n]\ w\ [n-m]}$$

Where *s*[*n*] is the entire speech sequence, $s_m[n]$ is a windowed speech frame at time *m* and *w*[*n*] is the windowing function. The frame length is about 10-page15
30 ms in the above equations; *m* is the time shift or the step size of the windowing function. A new frame is analysis by shifting the window function to a subsequent time. The amount of shifting is typically 10 ms.

The shape of the windowing function is very important. I can analysis on Rectangular window but I observed that Rectangular window is not recommended since it causes severe spectral distortion or leakage to the speech frames [8]. Other types of window function, those are minimise the spectral distortion, should be used. One of the most commonly used and very useful windows is the Hamming window for spectral distortion.

$$\mathbf{W[n]} = \begin{cases} \mathbf{.54 - .46}\boldsymbol{COS}\left(\frac{2\pi N}{N-1}\right) & \mathbf{0 \le n \le N-1} \\ \mathbf{0} & \boldsymbol{Otherwise} \end{cases}$$

Where N is the length of the windowing function. After Hamming windowing, the speech frame is goes to the next stage for further processing.

5.3 Cepstral Analysis

Cepstral analysis is the next stage after Linear predictive analysis refers to the process of finding the cepstrum of a speech sequence. Cepstrum, whose spelling is formed by shuffling the characters of the word "spectrum", is a based on "time-domain" representation of a signal. Even, a special term of "quefrency", which is also created by shuffling the spelling of the word "frequency", is generally used instead of "time". Cepstrum is defined as the inverse Fourier transform (FFT) of the logarithm of a signal's spectrum [8]. The equation gives the definition of a signal cepstrum,

$$\widehat{\boldsymbol{S}}[\boldsymbol{n}] = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\pi}} \int_{-\boldsymbol{\pi}}^{\boldsymbol{\pi}} \mathbf{ln}[\boldsymbol{s}(\boldsymbol{\omega})] \boldsymbol{e}^{\boldsymbol{j}\boldsymbol{\omega}\boldsymbol{n}} \, \boldsymbol{d}\boldsymbol{\omega}$$

In the above equation *S* (ω) is the Fourier spectrum of a signal, in our case frame and $\widehat{\boldsymbol{S}}[\boldsymbol{n}]$ is the cepstrum.

The basic idea behind cepstral analysis is deconvolution. In the speech analysis field, the page16

main use of cepstral analysis .That is extracting the vocal tract behaviour from a speech spectrum. It can be analysis by speech sequence that can be modelled by the bending the vocal tract impulse response and the glottal excitation,

$$\mathbf{s[n] = v[n]*u[n]}$$

in the above equation *v*[*n*] is the impulse of the vocal tract filter.

u[*n*] is the glottal excitation, which is generally a quasi-periodic impulse sequence for speech. In frequency domain, the above equation can be expressed as

$$\boldsymbol{S}(\boldsymbol{\omega}) = \boldsymbol{V}(\boldsymbol{\omega})\boldsymbol{U}(\boldsymbol{\omega})$$

In the equation $\widehat{\boldsymbol{S}}[\boldsymbol{n}] = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\pi}} \int_{-\boldsymbol{\pi}}^{\boldsymbol{\pi}} \mathbf{ln}[\boldsymbol{s}(\boldsymbol{\omega})] \boldsymbol{e}^{\boldsymbol{j}\boldsymbol{\omega}\boldsymbol{n}} \, \boldsymbol{d}\boldsymbol{\omega}$, we can see that logarithm and the integral splits the product of two spectra into a summation. Because of integration is a linear operation, the cepstrum of the speech sequence is actually the sum of the vocal tract cepstrum and the glottal excitation cepstrum.

$$\widehat{\boldsymbol{S}}[\boldsymbol{n}] = \widehat{\boldsymbol{V}}[\boldsymbol{n}] + \widehat{\boldsymbol{U}}[\boldsymbol{n}]$$

Where $\widehat{\boldsymbol{S}}[\boldsymbol{n}], \widehat{\boldsymbol{V}}[\boldsymbol{n}], \widehat{\boldsymbol{U}}[\boldsymbol{n}]$ are the cepstrum of the speech sequence, vocal tract impulse response and glottal excitation respectively.

Cepstrum possesses have two important properties [8]

(1) The cepstrum of a periodic sequence is always a periodic sequence.

(2) The cepstrum of an arbitrary filter impulse response is broken the sequence, which tends to zero when quefrency *n* tends to infinity. Since *v*[*n*] is the impulse response and *u*[*n*] is a quasi-periodic sequence, it can be analysis that the low-quefrency part of $\widehat{\boldsymbol{S}}[\boldsymbol{n}]$ belongs to $\widehat{\boldsymbol{V}}[\boldsymbol{n}]$ or the high-quefrency part is mainly constituted of $\widehat{\boldsymbol{U}}[\boldsymbol{n}]$. . Generally it is taken the first few coefficients from the low-quefrency part of the cepstrum that contains the characteristics of the vocal tract. These coefficients are called cepstral coefficients.

Generally, the cepstral coefficients is obtain in order, it is required to solve the Equation $\widehat{\boldsymbol{S}}[\boldsymbol{n}] = \frac{\boldsymbol{1}}{\boldsymbol{2\pi}} \int_{-\pi}^{\pi} \mathbf{ln}[\boldsymbol{s}(\boldsymbol{\omega})] \boldsymbol{e}^{\boldsymbol{j\omega n}}\, \boldsymbol{d\omega}$ However, cepstral coefficients are also be calculated from the Linear Prediction Coefficients (LPC) via a set of recursive procedure [8]. The cepstral coefficients result found in this way are called Linear Prediction Cepstral Coefficients

(LPCC). The following equations show the recursive procedure,

$$\widehat{\boldsymbol{V}}[\boldsymbol{n}] = \mathbf{ln}\ \mathbf{(G)} \qquad \text{for n=0}$$

$$\widehat{\boldsymbol{V}}[\boldsymbol{n}] = \boldsymbol{a_n} + \sum_{\boldsymbol{k-1}}^{\boldsymbol{n-1}} \left(\frac{\boldsymbol{k}}{\boldsymbol{n}}\right) \widehat{\boldsymbol{V}}[\boldsymbol{n}]\ \boldsymbol{a_{n-1}} \qquad \text{for } 1 \le n \le p$$

In speech recognition systems, we get the value from *p* number of LPC (i.e., the order of LPC is *p*). Then the LPC is converted into LPCC, for $1 \le n \le p$, by recursion from the above equation. There will be *p* number of LPCC. The LPCC will be grouped together to from the speech signal one feature vector for a particular speech frame. Often the log energy of the speech frame is also added to the feature vector. Let's assume *s*[*n*] is a speech frame of length *N* for $0 \le n \le N-1$, the log energy is given by the following equation,

$$Energy_{log} = \ln\left(\sum_{n=0}^{N-1} s^2[n]\right)$$

Get Together with the log energy, the length of one feature vector will be *p*+1.

5.4 Hamming windowing and FFT

Hamming window (See the LPCC analysis for details of Hamming windowing. Fast Fourier Transform is performed on the window frame in order to obtain the speech spectrum. The resultant FFT spectrum, $S[k]$, is passed to the next stage for next processing.

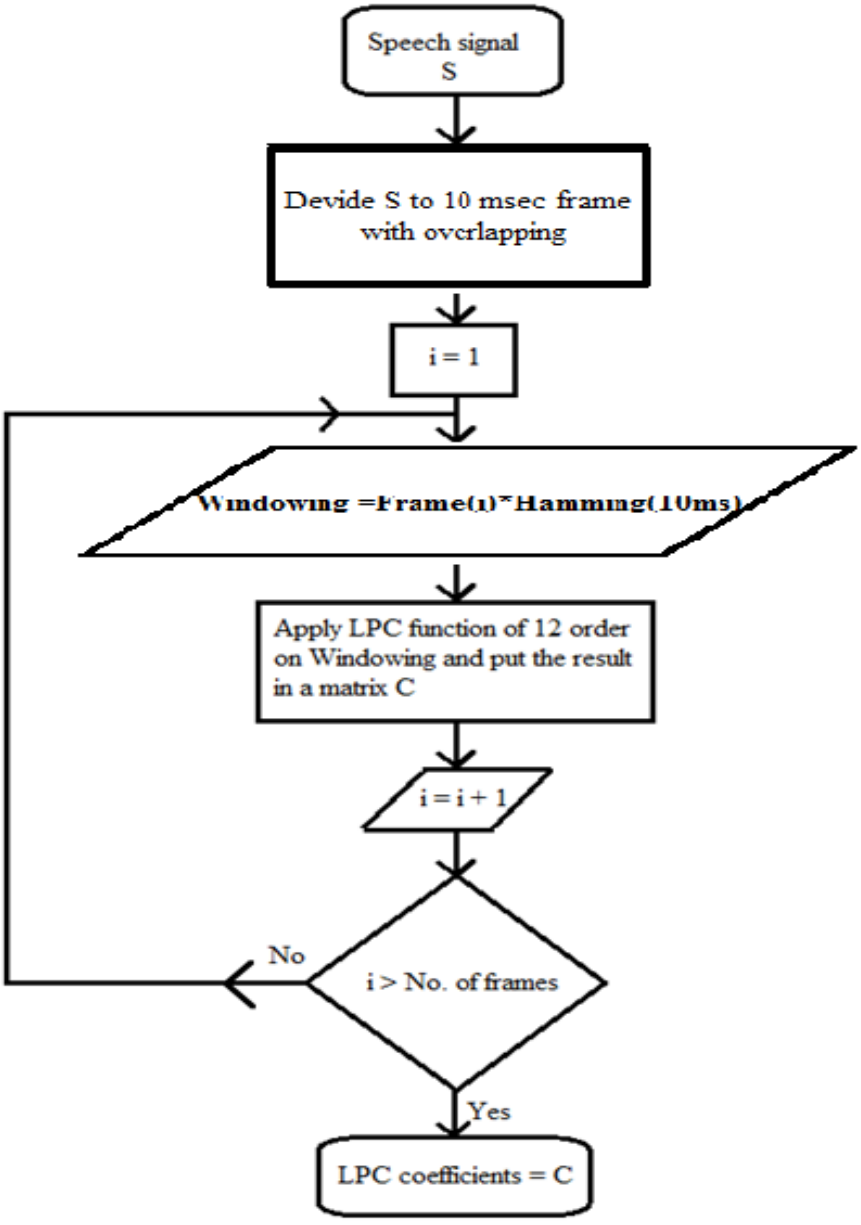


FIGURE 5.2 LPCC FLOW CHART ALGORITHUM

5.5 Mel-Scale Frequency Cepstral Coefficients (MFCC) Feature Extraction

Mel Frequency Cepstral Coefficients (MFCC) [11] is one of the commonly used feature extraction front-ends in speech recognition systems. The technique based on FFT, that means feature vectors are extracted from the frequency spectra of the windowed speech frames.

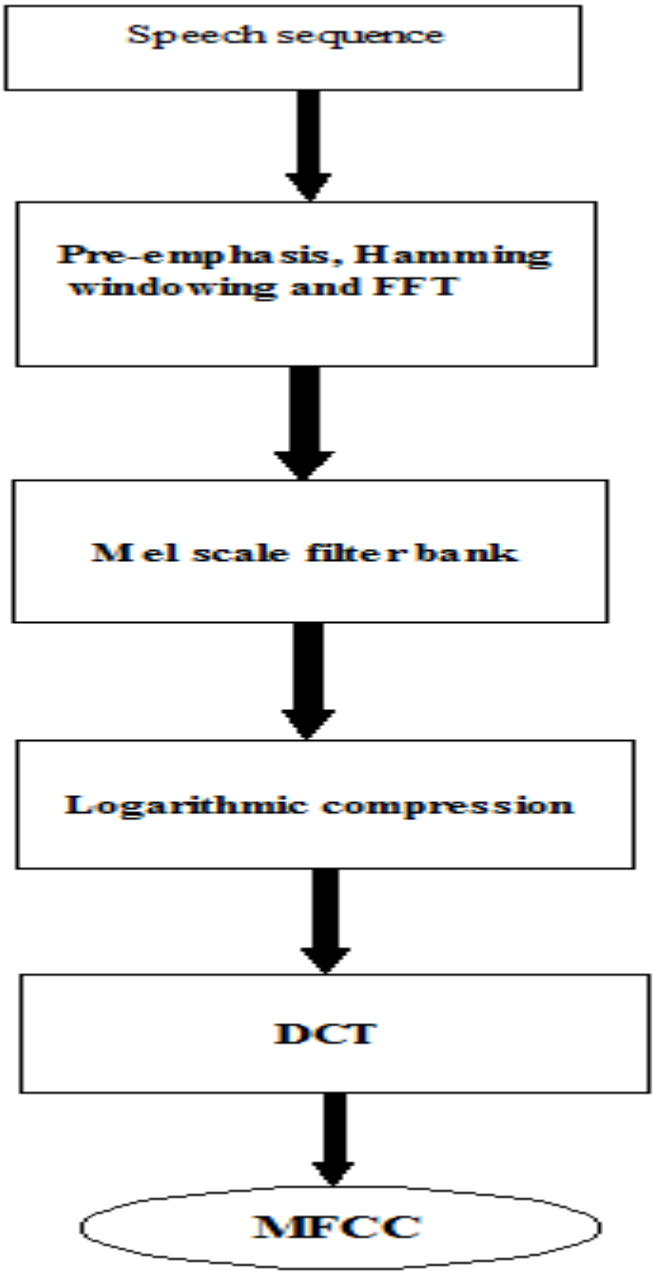


FIGURE 5.3 Block diagram of the MFCC processor

5.6 Mel scale Filter Bank

It is filter bank is a series of triangular band pass filters, which copy the human auditory system. The filter bank is based on a non-linear frequency scale that called the Mel scale. The Mel frequency scale is a intension of specking measure of pitches judged by human. According to Stevens et al. [12], a 1000Hz tone, with 40dB above the listener's threshold, is defined as having a pitch of 1000 mels. Below 1000Hz, the Mel scale is approx linear to the linear frequency scale. More than 1000Hz reference points; listeners tend to perception the same pitch increments with longer frequency intervals as much required. Hence the relationship between the Mel scale and the linear frequency scale is non-linear and approximately logarithmic more than 1000Hz. Mathematical relationship between the Mel scale and the linear frequency scale is

$$\textbf{Mel (f)} = 2595 log_{10} \left(1 + \frac{f}{700}\right)$$

Where $f_{Mel}$the Mel frequency in meal and f is is the linear frequency in Hz.

As previously say that Mel frequency consists series of triangular band pass filters. The filters

are overlapped in this way that the lower boundary of one filter is situated at the centre frequency of the previous filter and the upper boundary is situated at the centre frequency of the next filter. The max response of a filter, i.e. The highest vertex of the triangular filter is located at the filter's centre frequency and is normalised to unity.

The filter of centre frequency increased evenly along with Mel frequency scale.

It can be shown that the centre frequency of the m^{th} filter in the filter bank can be found by evaluating the following equation,

$$f_{cm(Mel)} = f_{L(Mel)} + \frac{m(f_{H(Mel)} - f_{L(Mel)})}{M+1} \qquad 1 \leq m \leq M$$

Where $f_{cm(Mel)}$ is the centre frequency of m^{th} filter in Mels. $f_{H(Mel)}$ and $f_{L(Mel)}$ is lower and upper ends in mels, respectively, entire filter bank. There are *M* triangular filters between the range between $f_{L(Mel)}$ and $f_{H(Mel)}$.

Suppose that $H_m[k]$ show the frequency magnitude response of the m^{th} filter .where k is the discrete frequency index in the digital domain. The filter o/p of the m^{th} filter, X_m, can be show below the equation

$$X_m = \sum_{k=0}^{\frac{N}{2}-1} |S[k]|^2 \; |H_m[k]| \qquad 1 \leq m \leq M$$

S[*k*] is the *N*-point FFT spectrum of the windowed speech frame, as described in Pre-emphasis, Hamming windowing and FFT .It is Notice that the number of summation points is just *N*/2. It is because half of the FFT spectrum is a mirror image of the other half.

5.7 *DCT (*Discrete Cosine Transform)

DCT is a Fourier-related transform same like the discrete Fourier transform, but using only real numbers. It is equivalent to DFTs of roughly twice the length, operating on real data. (Since the Fourier transforms give same real and even function is real and even). A Discrete p Cosine Transform computes a sequence of data points at various frequencies give summation of cosine functions oscillating. Discrete Cosine Transform on Mel Scale is motivated by speech frequency domain characteristics. The module of Discrete Cosine Transform reduces the speech signal's repeated information, and reaches the speech signal into feature coefficients with minimal dimensions.

The final step of the algorithm is to decorrelate the filter outputs. Discrete Cosine Transform (DCT) is applied to the filter outputs and the first few coefficients are grouped together as a feature vector of a particular speech frame.

Suppose *p* is the order of the Mel scale cepstrum. The feature vector is obtained by considering the first *p* Discrete Cosine Transform coefficients. Filter bank energies is

$$c_d = \frac{1}{M} \sum_{m=0}^{M-1} c_m \; Cos \left(\frac{\pi(2d+1)m}{2M} \right)$$

In the above equation c_d is the d-th cepstral coefficient, M is the total number of filter banks and c_m denotes the log energy for filter bank m. typically, c1 – c12 constitute the MFCCs.

Chapter 6

6.1 Relative Execution Time (Algorithm)

Every person use self implementation of feature extraction algorithm. We are analysed different algorithms of times complexity with respect to the algorithm. We take the least time, so we check relative execution time for each algo. We analyse the test algorithms on an actual wireless sensor node: the Jennic JN5148 module. In the experiment we use a 32-bit RISC CPU and have 128 kb available for program code and data. By choosing single specific device to perform our benchmark test, we do not deliver extensive proof of the validity of the results for Wireless sensor nodes in general.

Program is compiled on the Jennic JN5148 chipset. The nodes have no additional operating system (os). During the test , there is no other processes are running. All calculations are analysis using floating point numbers. Algo use the same sound fragments. We take Sampling frequency is 8 kHz, because that is both feasible for the Jennic analog-digital converter and holds information of the recorded sound for the different analysis and calculation of the features, we are taking frame size is 20–30 ms for easy calculation. If the signal is small , then the number of samples for a spectral analysis will be too low; if the signal is high, then the signal changes more change throughout the frame, so that analysis of the feature extraction algo to become less value . Fourier transform, which is used by the frequency domain features and MFCCs, requires a frame length equal to a power of two; we take a frame length of 256 samples. This frame length corresponds to 30 ms, which fits in interval of 20–30 ms.

The Relative Execution Time (RET) is analysis by measuring the how much time taken for each feature extraction algorithm to finish. We calculate that respective executions and different result show same measurements, then we did running the same test on two different nodes. The difference in processing time between a signal of one second and a signal of 30 ms was less than 0.1% for all different algorithms. The result shows Relative Execution Time (RET) for a subset of the different features that we have tested. The features that are not present in this figure have computation times that are of the same magnitude as the other

features of the same category. Difference exception occurs at the time of long-term features (shimmer and jitter). Analysis of MFCC and LPCC is comparable to frequency domain features, whereas LPCC are found result even faster than MFCC. The reason for this is that the analysis the LPCC requires less multiplication operations.

We show the Relative Execution Time for each group of features. We analysis the value one for the least Computationally-intensive feature group. The value for this category is derived from the most computationally-intensive feature. The abbreviations in Table will be used in the remainder of this article.

Table 6.1 Relative execution time (RET).

Feature	Abbreviation	RET
Frequency domain features	FD	15
LPCC features	LPCC	31
MFCC features	MFCC	77

6.2 Orders of Complexity

In order to gain a first notion of the algorithms' calculation cost, we consider their orders of Complexity. See also for more details about the calculation of the features.

(I)Time domain features (STE, ZCR, sound amplitude (SA), PD) time complexity is O (N)

(II)The features time complexity is O (N) for each filter h_m, the values are multiplied with the signal values s(i). If the number of filters is f, this results in f * N multiplications.

(III)Frequency domain features: Press et al. [13] argue that the FFT can be obtained in O(Nlog2N) complexity. For all frequency domain functions require an additional O(N) step because after calculation of the frequency spectrum, they each have the same order of magnitude as the result of FFT.

(IV)MFCC and LPCC both use the Fast Fourier Transform (FFT) as one of the main steps. Other steps that are performed in both algorithms have lower complexity than the calculation of the FFT. The complexity of both algorithms can be obtained as O(Nlog2N).

(V)Jitter, shimmer: small more calculations required after calculation of F0 and SA, respectively. See the consecutive frame and the order of this is O(N), therefore jitter and shimmer have orders equal to the FFT (O(Nlog2N)) and time domain (O(N)), respectively.

6.3 The Details of Research Work

We have analysed two algorithms that are MFCC and LPCC. Both algorithms are used for feature extraction. The performances of the two algorithms are compared for better performance with high recognition rate and low computational complexity and the major advantage of comparing these two algorithms is that they improves the reliability of the system.

Hear we here are trying to recognise the English numerical digits from 'zero' to nine'. All the programming implemented here is done in matlab due to obvious reasons of it being the most efficient tool for mathematical and signal analysis.

At first we are present a small description of the words used:

Word	Sounds	APRABET
Zero	/z I r o/	Z-IH-R-OW
One	/w ΛΛ/	W-AH-N
Two	/t u/	T-UW

THREE	/θ r i/	TH-R-IY
FOUR	/f o r/	F-OW-R
FIVE	/f a^{y} v/	F-AY-V
SIX	/s I k s/	S-IH-K-S
SEVEN	/s ε v ∂ n/	S-EH-V-AX-N
EIGHT	/e^{y}t/	EY-T
NINE	/n a^{y}n/	N-AY-N
OH	/o/	OW

TABLE 6.3 Description of digit (phonetic)

before doing any speech recognition work we have to convert the speech into digital format.

In order to show the performance of the different steps involved in LPC extraction process, the following figures were executed for save S1test.wav file. In Figure 6.1, the original speech waveform of zero and how is affected after the pre-emphasis filter is illustrated. Figure 6.2 presents the effect of speech waveform before the preemphasis filter using a Hamming window, and Figure 6.4 show FFT and LPC spectrum of the zero (0) spectrum of one frame as compared with its magnitude spectrum.

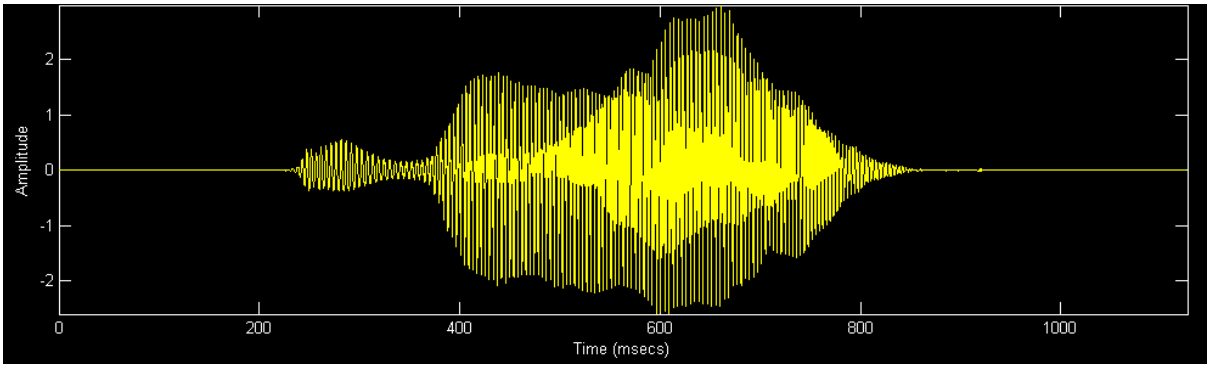


Figure 6.1: Plot of time waveform and frequency spectra of spoken 'zero'

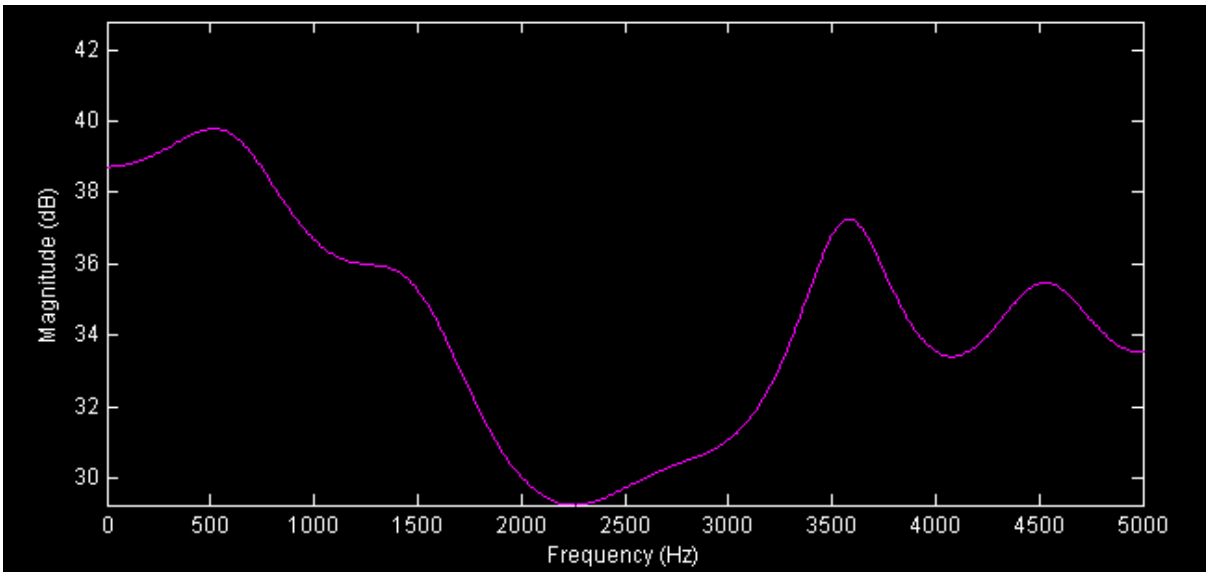


Figure 6.2: Original speech waveform before the preemphasis filter

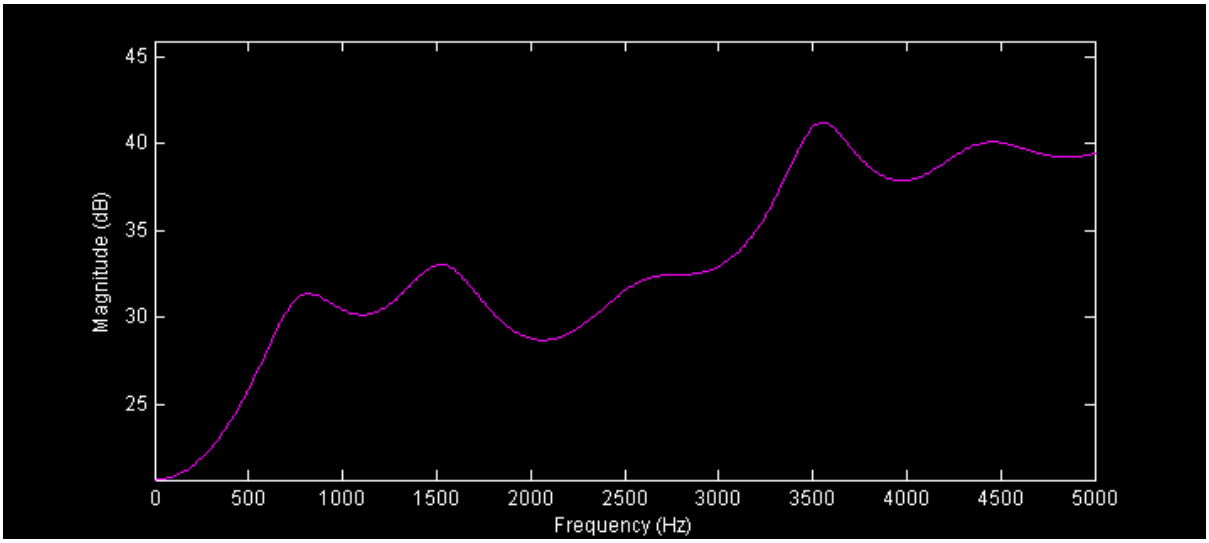


Figure 6.3: Speech waveform after the pre emphasis filter with coefficient equal to 0.97.

example above (Fig. 6.2) the frequency range was 0-8000 Hz, that is, it was 0-Fs/2, where Fs is the sampling frequency. Now I want to see the spectrum in the range, say, 0- 5kHz,

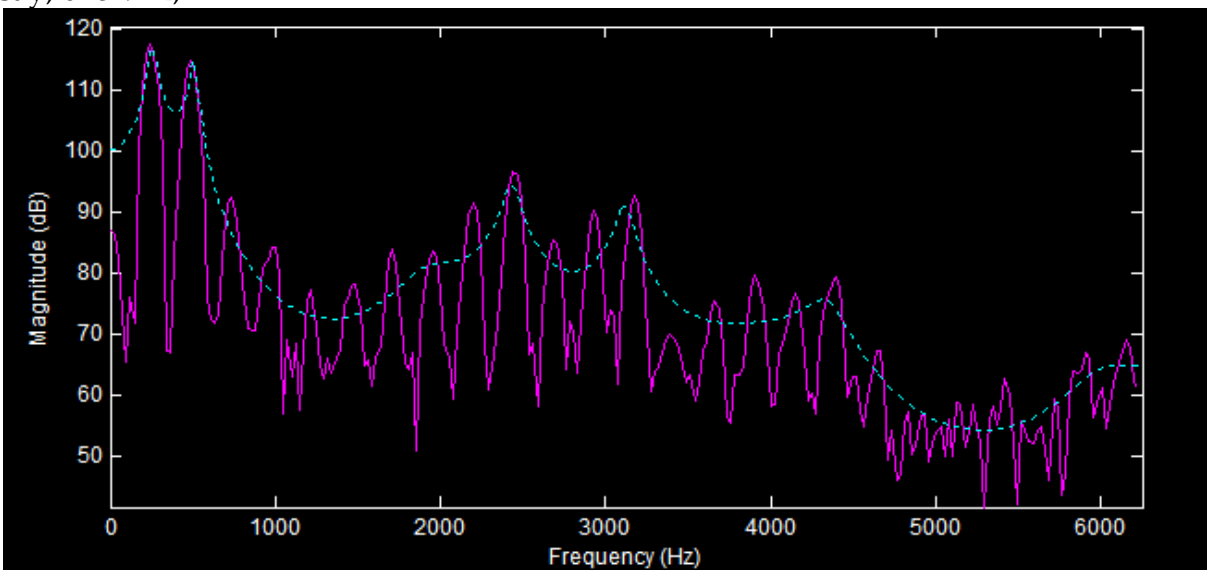


Figure 6.4 FFT and LPC spectrum of the zero (0)

Estimates Frequencies and Formant Amplitudes (in dB)

The formant frequencies are computed by peak-picking the LPC spectrum. To get accurate estimates of the formant frequencies, one needs to choose the LPC order properly depending on the sampling frequency. Although 12-pole LPC analysis is typically adequate for telephone speech, it is not adequate for speech recorded at sampling frequencies of 16 kHz or above. In the example above (Fig6. 2) the LPC order was 12, and the third formant (F3) had a value of F3=3064 Hz, which is suspiciously high for a third formant (for an adult male speaker). So we increasing the LPC order to 18 will yield a better estimate of the second and third formants for this example.

we take the frequency range set 0 to 5 kHz and insert the label for three formants f1, f2 and f3.

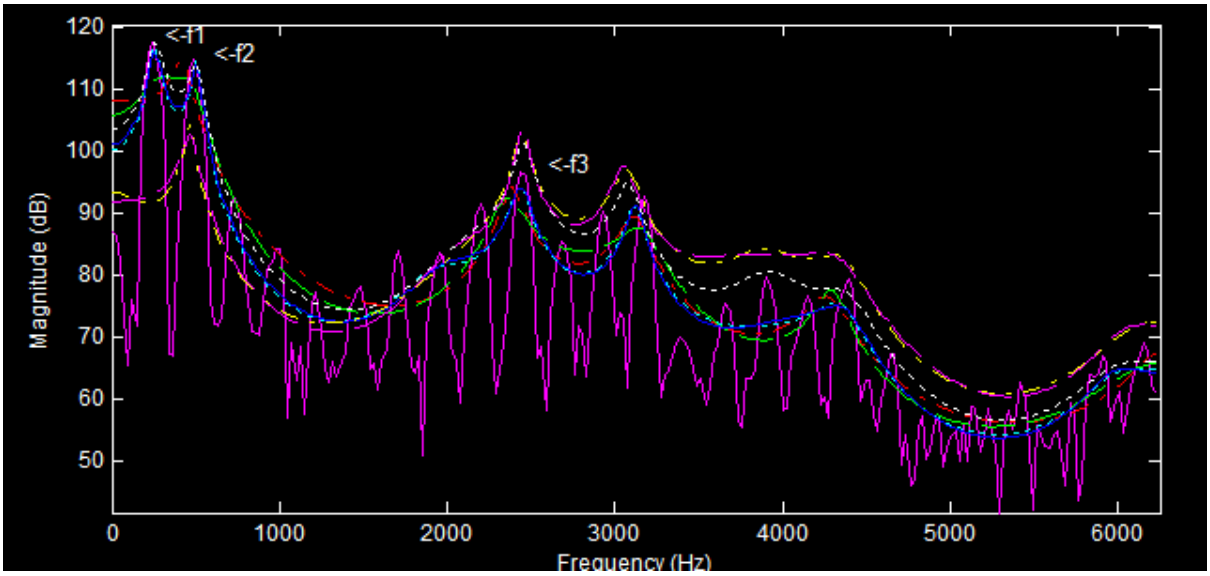


Figure 6.4The FFT and LPC spectrum for word zero with different order

In this spectrum we use LPC analysis such as using pre-emphasis FIR filter of the form H (*z*) = 1- 0.97*z* -1, and using Hamming window and frequency range was set to 0-5 kHz. We Increasing the LPC order 12 to 18 will yield a better estimate.

6.5 Error Analysis:

LPC error analysis using a wave frame of speech analyzed by both the STFT and the LPC autocorrelation method. The upper graphics(figure 6.5) panel shows the window-weighted voice speech frame; the second graphics(figure 6.5) panel shows the LPC error signal; the third graphics(figure 6.5) panels shows the log magnitude spectrums of the STFT and the LPC system and the bottom graphics(figure 6.5) panel shows the log magnitude spectrum of the LPC error signal.

Using the speech file 's2test.wav' set the frame length to Lm = 20 msec. Run the LPC error analysis using the autocorrelation method of LPC. Starting sample for the word 'zero'. analysis at p = 18 samples at the beginning and end of the prediction error signal. We can analysis different question -answer from this output .Why is the prediction error large in these intervals? What is the spacing between impulses ? Why does the prediction error contain" impulses" ? How is the spacing between impulses related to the pitch period of the waveform, as seen in the top graphics plot ? Also Compare the STFT of the signal to the

STFT of the prediction error. How do they differ? In both cases you should see the same shape repeated periodically in frequency. Now switch to the covariance method of LPC analysis, using the same speech frame as above. Is the prediction error more or less impulsive than tha t obtained from the LPC autocorrelation analysis? Why is the prediction error not large at the beginning and end of the window? The output figure is

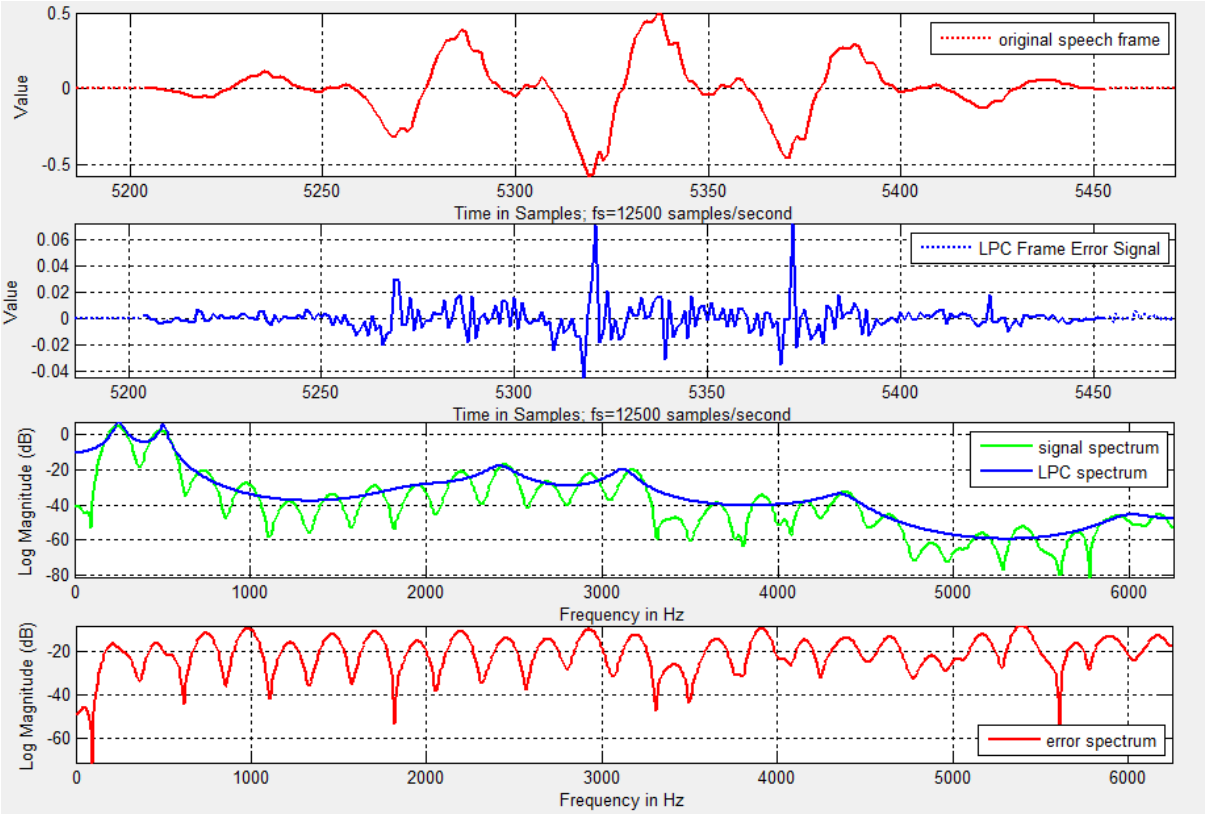


Figure 6.5 Error analysis

MFCCs are commonly analysis as follows:

1. Found the Fourier transform of (a windowed excerpt of) a signal

2. Map the log amplitudes of the spectrum obtained from Fourier transform signal onto the Mel scale, using triangular overlapping windows.

3. Observed Discrete Cosine Transform of the list of Mel log-amplitudes, as if it were a signal.

4. The MFCCs are the amplitudes of the resulting spectrum.

A set of matlab modules were written to find the above mentioned coefficients and the

Corresponding graps for letters ‘zero’ to ‘nine’ are given below.

For Zero:

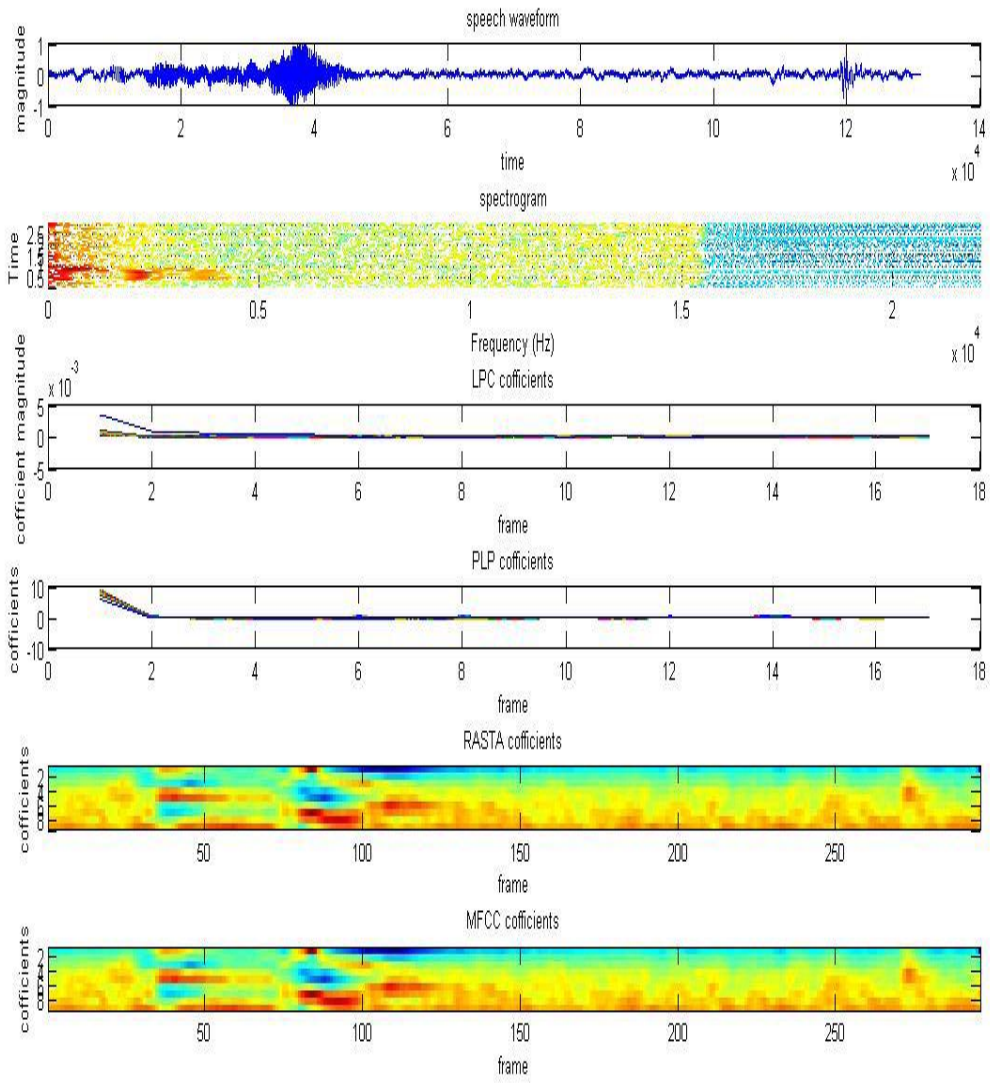


Figure 6.6 Coefficients for zero

6.3.4 Distance calculation

After extracting the feature vector the next step which is important in speech recognition is distance calculation between the feature vectors which were calculated by the last step. Here we have a nalysed the two most prominent method of distance measure which are Euclidean distance and the itakura-saito distortion measure (likelihood distortion measure). Here we have taken two wave form segmented in 10 msec frames.

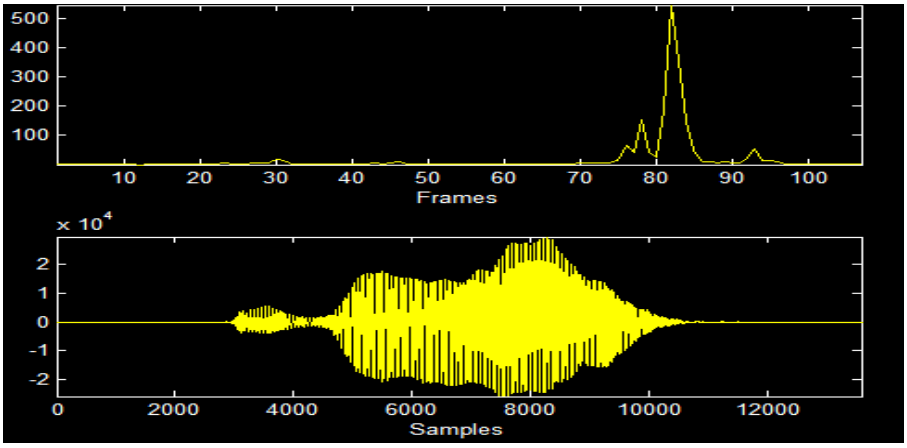


Figure 6.8 Spectral domain measures using Itakura-Saito for s2test.wav

Thus it can be easily seen that even though itakura-saito distance is a very good form of distance measure its performance for the case of isolated word recognition with very little database is very poor. Thus we have decided to use Euclidean distance and the itakura- saito Distortion measure (likelihood distortion measure). Codebook can extracted the features a from each speaker was build clustering the feature vectors using the itakura-saito algorithm. Codebook from all the speakers was collected in a database. A distortion measure based on minimizing the Euclidean distance was used when matching the unknown speaker with the speaker database. The experiments conducted using database, showed that it was possible to achieve MFCC get 80% and LPCC identification 60 % using itakura –saito algorithm .

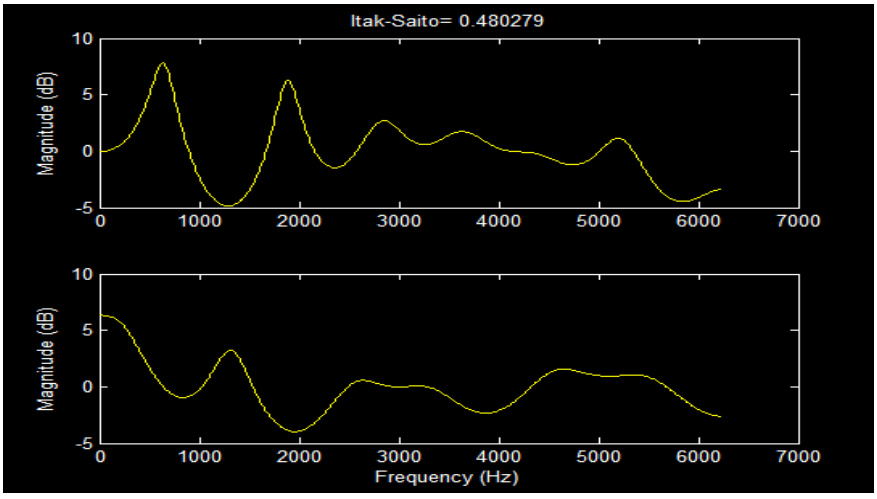


Figure 6.7 comparing between test “zero” and train“zero” one using Itakura-Saito measure. Top is testing zero (original) and bottom is train zero.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Speech recognisation System operate in two modes that is Enrollment mode and Recognition mode.

The first one is to create a data base of templates for once spoken words for example 'zero' to 'nine' and create the well trained database. Second task is to recognise speech signal. The MFCC feature coefficient is used here for reasons stated earlier Euclidean distance is used to measure the distance between the feature vectors (Itakura-Saito).The major part of this work is the implementation and analysis of the Automatic Speech Recognition algorithms using MATLAB and observed their individual performance. The speech reorganisation system model has been developed for compare the two algorithms that is MFCC and LPCC. The performance has been evaluated by considering the sets of speech signal.

It is analysis that MFCC used in Automatic speech Recognition system (ASP) provide 80 percentage accuracy where as LPCC used in Automatic Speech Recognition (ASP) given 60 percentages accuracy. Results and analysis show that MFCC algorithm gives better result than LPCC algorithm. From the simulation results we also add that MFCC algorithm, which require more calculation but perform better than LPCC in terms of efficiency and accuracy.

7.2 Future work

We are considering an indoor environment(less noise). We want to see that the classification of sounds into global categories can be performed with very low calculation effort. For gender recognition algorithms that required the low cost and frequency domain features achieve results. It may better than algorithms that uses for heavy-weight MFCCs for classification but LPCC and MFCC order of complexity of both algorithms can be expressed as O(Nlog2N). So we are trying to crate batter algorithm which have Orders of Complexity should be less then O(Nlog2N). We are seeing for the different categories (Global, Gender and indoor sound classification), that use of low-cost algorithms can be equally effective for deployment indoors as the use of high-cost algorithms.

References

[1] Jainath Yadav and Anshu Kumari," Emotion Recognition using LP Residual at Sub-segmental, Segmental and Supra-segmental levels" International Conference on Communication, Information & Computing Technology (ICCICT), 2015.

[2] N. Thapliyal and G. Amoli, "Speech based emotion recognition with Gaussian mixture model"*International Journal of Advanced Research in Computer Engineering & Technology,* 2012

[3] Kevin D'souza and K.T.V Talele "Voice Conversion Using Gaussian Mixture Models" *International Journal of Advanced Research in Computer Engineering & Technology*, 2015.

[4] Vibha Tiwari, "MFCC and its applications in speaker recognition", International Journal on Emerging Technologies, pp 19-22, 2010.

[5] Md Touseef Sumer and Pathan Md. Aziz khan"A Review on Speaker Recognition Using Gaussian Mixture model", *International Conference on Computer & Communication Technologies 2K14*

[6] Paliwal, K.K. and Kleijn, W.B. (1995) Speech synthesis and coding, chapter quantization of LPC parameters. Elsevier Science Publication, Amsterdam, 433-466.

[7] Rabiner, L. and Schafer, R., *Digital Processing of Speech Signals*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[8] Rabiner, L. and Schafer, R., *Digital Processing of Speech Signals*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[9] Furui, S., Digital Speech Processing, Synthesis, and Recognition Second Edition, Revised and Expanded, Marcel Dekker, Inc., New York, 2000.

[10] Rabiner, L. and Juang, B., *Fundamentals of speech recognition*. Prentice Hall, Inc., Upper Saddle River, New Jersey, 1993.

[11] Davis, S. and Mermelstein, P., Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Trans. ASSP-28*, pp 357-366, 1980.

[12] Stevens, S., Volkmann, J., and Newman, E., "A Scale for the Measurement of the Psychological Magnitude Pitch." *Journal of the Acoustical Society of America* 8: 185–190, 1937.

[13] Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. Numerical Recipes in C, 2nd ed.; The Art of Scientific Computing; Cambridge University Press: New York, NY, USA, 1992; pp. 504–508.

[14] S. Malik, F. A. Afsar,"Wavelet Transform based Automatic Speaker Recognition", IEEE 13th International Multitopic Conference, INMIC, Islamabad, pp. 1-4, 14-15 December, 2009.

[15] S. Tripathi, S. Bhatnagar. "Speaker Recognition", IEEE Third International Conference on Computer and Communication Technology (ICCCT), Allahabad, pp. 283-287, 23-25 November, 2012.

[16] P. Kumar, M. Chandra, “Hybrid of Wavelet and MFCC Features for Speaker (WICT), Verification”, IEEE World Congress on Information and Communication Technologies Mumbai, pp. 1150-1154, 11-14 December, 2011

[17] J. Tchorz and B. Kollmeier, A model of auditory perception as front end for automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 106(4), Pt. 1, Oct. 1999.

[18] M. Holmberg, D. Gelbart, and W. Hemmert, Automatic Speech Recognition with an adaptation model motivated by auditory processing," *IEEE Trans. Audio, Speech, Lang. Processing*, vol. 14, no. 1, pp. 44-49, Jan. 2006.

[19] D. S. Kim, S. Y. Lee, and R. M. Kil, Auditory processing of speech signals for robust speech recognition in real world noisy environments," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 1, pp. 55-69, Jan. 1999.

[20] B. Strope and A. Alwan, A model of dynamic auditory perception and its application to robust word recognition," *IEEE Trans. Speech Audio Processing*, vol. 95. no. 5, pp. 451-464, Sep. 1997.

[21] C. R. Jankowski Jr., H. H. Vo, and R. P. Lippman, A comparison of signal processing front ends for automatic word recognition," *IEEE Trans. Speech Audio Processing*, vol. 3. pp. 286-293, Jul. 1995.

[22] Linear Predictive Coding, Jeremy Bradbury, December 5, 2000.