ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT

Non-Uniform Sampling based Feature Extraction for Automatic Speech Recognition

By

Bisrat Girma

Advisor

Dr.Eneyew Adugna

A Thesis Submitted to Addis Ababa University, School of Graduate Studies, Institute of Technology, in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Electrical Engineering

February, 2012

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY SCHOOL OF GRADUATE STUDIES

INSTITUTE OF TECHNOLOGY

ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT

Non-Uniform Sampling based Feature Extraction for Automatic Speech Recognition

By

Bisrat Girma

Approval by Board of Examiners

_____	_____
Chairman, Dept. Graduate Committee	Signature
Dr.Eneyew Adugna	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
Ato. Bisrat Derebssa	_____
External Examiner	Signature

Declaration

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or any other university, and all sources of materials used for the thesis have been fully acknowledged.

Bisrat Girma _____

Name Signature

Place: Addis Ababa

Date of Submission: _____

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Eneyew Adugna _____

Advisor's Name Signature

Abstract

In Automatic Speech Recognition (ASR) robustness to additive noise remains a large unsolved problem. As a result selecting a proper feature extraction method has been a key research area. So many feature extraction algorithms have been proposed that are designed specifically to have a low sensitivity to background noise. However, there are still some performance problems in noisy environments. This thesis is an attempt to develop a new feature extraction method based on a combination of non-uniform sampling and mel-frequency cepstrum coefficients (MFCCs) method since MFCC works very well under clean environment.

Non-Uniform sampling is used when fluctuations in sampling instants cannot be ignored or when signal samples can be obtained only at irregular or even random time intervals. It also sometimes deliberately introduced in order to see some useful effect such as the suppression of aliasing and to reduce the quantization noise which as a result improve the performance of Analog to Digital converter (ADC). Since improving ADC using non-uniform sampling method helps to increase the representation of the original signal in digital form and the non-uniformity of sampling as compared to uniform sampling efficiently improves the spectrogram which allows to determine true signal components at frequencies exceeding the half of mean sampling rate and also the fact that spectra of the non-uniform sampled signals are not uniform in frequency domain helps to represent the non-uniform spectral sensitivity of human hearing which might helps to autofocus on most reliable part of the spectrum in noisy cases, in this thesis we deliberately introduced non-uniform sampling in order to modify the front-end analyzer to better capture the speech information and incorporate the temporal characteristics in the feature set.

The step used for implementing the non-uniform sampling based speech recognition can be summarized by the following steps. The first step performs oversampling and end point detection. The second steps includes non-uniform sampling of the speech signal using sine-wave crossing method and speech segmentation by using short term temporal analysis. The third step includes finding the feature vectors using NU-MFCCs methods and vector quantizing (VQ) of the speech features.

Finally, by using the means and variance of the feature calculated in VQ as an input the Gaussian Mixture Models (GMM) is used for classifier or modeling purpose. Experimental results show the average performance of the recognition system based on NU-MFCCs is around 92.18% under normal external surrounding (>35dB) and 51.27% under additive white Gaussian noise(AWGN) condition (between -5 and 35 dB) whereas in MFCC case, 92.36% and 42% respectively. But when the system was trained with a mixture of normal external surrounding and 10 dB SNR (AWGN) condition the average performance of NU-MFCCs went up to 74.84%. Similarly in MFCC case, it increased to 65.87%.

Keywords: Non-Uniform Sampling, MFCC, ASR, GMM

Acknowledgement

During the course of this study, I have received the assistance of many people to whom I am very grateful.

First and foremost, I want to thank my advisor, Dr. Eneyew Adugna, for giving me his constant support, invaluable guidance and advice throughout my studies. Thank You.

I would like to convey my utmost gratitude to Prof F.Marvasti for sending me reference materials. And also I am grateful to all my friends for the great time we had together and for helping me on some materials staff.

I would like also to thank my parents for their endless support and love. Their encouragement during both bad times and good times always kept my spirits high.

Finally, I would like to dedicate this thesis for a memory of my best friend Fikru Yetbarek.

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

ASR	Automatic Speech Recognition
AWGN	Additive White Gaussian Noise
CMC	Cepstral Mean Correction
CPU	Central Processing Unite
CSR	Continuous Speech Recognition
DASP	Digital Alias-free Signal processing
DCT	Discrete Cosine Transform
DSP	Digital Signal Processing
EM	Expectation Maximization
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
IWR	Isolated Word recognition
LBG	Linde-Buzo-Gray
LPC	Linear Predictive Coding
MAP	Maximum A *posteriori* Parameter
MFCCs	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
NDFT	Non-uniform Discrete Fourier Transform
NU-MFCCs	Non-Uniform Mel Frequency Cepstral Coefficients
SNR	Signal-to-Noise Ratio
VQ	Vector Quantization

CHAPTER ONE: INTRODUCTION

1.1. Background

In an ideal case, the features of a digital signal obtained as a result of analog-to-digital conversions would copy the features of the original analogue signal. The reality is different. The sampling and quantization operations of this kind of conversion impact on the characteristics of the digital signals obtained substantially. The characteristics of the analogue signal at the input of an ADC and of its digital output are representative rather than identical [1]. So in this thesis we gave much emphasis on sampling since it is the first important step used to discretize the analog signal.

Uniform sampling has been used for a number of decades for digital signal processing because of its simplicity and since it has straightforward processing techniques. But after 1960 a great deal of investigation on non-uniform sampling has been done [2-4].

The non-uniform sampling might take place either when signal sample values are obtainable and could be taken only at some random unpredictable time instants or when these sample values are taken non-uniformly in order to obtain some specifically targeted effects, like avoiding aliasing.

Now a day's non-uniform sampling is used for a number of applications like in Sensor Network Communication, in Time-Domain Dielectric Spectroscopy, in Image Processing and also in some Biomedical Engineering application.

Non-uniform sampling methods allows one to perform an experiment faster, collect more transients (to improve S/N) or acquire more data points (to improve resolution) in any indirect dimension. And also it opens up the possibility of distinguishing all spectral components of the signal, even if their frequencies substantially exceed the mean sampling rate. In this thesis, we introduce non-uniform sampling deliberately in order to see or get its advantage on Automatic Speech Recognition.

Technically speaking, Automatic Speech Recognition (ASR) refers to a mechanism (hardware and software combined) that stores some representations of distinguishing characteristics of speech with a source of input equipment, such as a microphone and further processes these representations to match them to incoming speech in an effort to interact with machines, computers and/or human users. The first primitive recognizer was developed at Bell Labs during the early 1950s.

Development of speech recognition systems has been accelerated in a mid half of 1960s' when a popular expectation-maximization (EM) algorithm forwarded, known as the *Baum-Welch Re-estimation Algorithm* (or Forward-Backward Algorithm), to estimate the parameters of a Hidden Markov Model (HMM) or Gaussian Mixture Models(GMM) iteratively [6-8].

Techniques for extracting feature parameters from the speech signal have also been studied besides the improvements in modeling. At the end of 1960s *Cepstral Analysis* introduced which performs deconvolution of the speech signal to separate an excitation sequence from an impulse response convolved with it. By using mel basis frequency scale, which imitates the human ear behavior, Mel-frequency Cepstrum Coefficients (MFCC) [8] have been used in many recognition applications successfully.

However, despite considerable progress, many aspects of ASR are still unsolved and problematic even today because the engineering view of human speech production and human ear perception is not yet fully conclusive. Hence, the research community believes that the field of speech recognition is still in its early infancy and will remain to pose a challenging problem for the near future [7, 9].

1.2. Statement of the Problem

There are several factors that render speech recognition methods complicated [9]. This factor summarizes as follow:

First , most speech recognition algorithms, in principle, can be used in either a "speaker-dependent" or "speaker-independent" mode, and the designation for a particular system depends upon the mode of training. A *speaker-dependent* recognizer uses the utterances of a single speaker to learn the parameters (or models) that characterize the system's internal model of the speech process. The system is then used specifically for recognizing the speech of its trainer. Accordingly, the recognizer will yield relatively high recognition results compared with a *speaker-independent* recognizer, which is trained by multiple speakers and used to recognize many speakers (who may be outside of the training population). Although more accurate, the apparent disadvantage of a speaker-dependent system is the need to retrain the system each time it is to be used with a new speaker. Beyond the accuracy/convenience trade-off is the issue of necessity.

Second, the *vocabulary size* is another key factor that affects the dimension of difficulties in recognition. Considering the number of ambiguous utterances (e.g.,“know” and “no”) and acoustic confusability (e.g., “beer” and “bear”) in the vocabulary of interest naturally, the performance of a particular recognizer is expected to degrade with the increasing vocabulary size [9-10]. Small vocabulary (less than 100 words) recognizers can perform relatively simpler tasks such as destination sorting systems for shipping tasks, credit card number or telephone number recognition. In these examples, specific models for each word in the vocabulary can be stored in the system and recognition is achieved by an exhaustive search through the whole vocabulary. As the vocabularies become larger, the recognition task creates increasing memory requirements. When training and modeling for each word in a larger vocabulary becomes impractical, models of sub word units like syllables and phonemes are preferred over models of words.

Third, isolated word recognition (IWR) systems assumed that the speaker deliberately utters sentences with sufficiently long pauses between words (typically, a minimum of 200 msec is required) so that silences are not confused with weak fricatives and gaps in plosives.

The most complex recognition systems are those which perform *continuous-speech recognition* (CSR), in which the user utters the message in a relatively (or completely) unconstrained manner. The recognizer must be capable of somehow dealing with unknown temporal boundaries in the acoustic signal and performs well in the presence of all the coarticulatory effects and sloppy articulation (including insertions and deletions) that accompany flowing speech. As a result, IWR is expected to have higher recognition accuracy. Generally speaking, CSR systems require more CPU power and memory than ISR systems. Further, inter-speaker and intraspeaker variations of the training population in articulation, pronunciation and intonation make it even harder for CSR systems to determine speech boundaries, thus yielding lower classification accuracy compared to that obtained with ISR systems.

Fourth, another important parameter in ASR system performances is whether any linguistic information is built into the recognizer to fine-tune algorithms. Speech recognizers are trained on basic speech unit models such as phones, phonemes, syllables or words. Linguistic (or grammar) constraints deal with how these basic units should be concatenated to form a meaningful message in a particular language. Such linguistic information may be embedded into more sophisticated recognizers.

Finally, one of the major challenges of the speech recognition problem is to make the system robust to background noise. The waveform of a speech signal is very susceptible to the variations in channel, microphone characteristics, room reverberation or background noise. Nonlinear effects of noise and channel distortion can be very destructive for recognition tasks, especially when no a-prior knowledge is available about their characteristics.

All the above factors have a major impact on the success of a particular ASR system and add up to determine the necessary level of system complexity in design phase. Among all of the above-mentioned factors, performance degradation in noisy real world environments is probably the most significant factor limiting the progress of ASR technology today [7].

Suppose that *s(t)* is the clean speech signal, *n(t)* is additive noise , then the degraded signal *y(t)* can be computed as Eq. (1.1)

$$y(t) = [s(t) + n(t)] \tag{1.1}$$

Assuming that the additive noise is stationary and uncorrelated with the clean speech then the power spectrum of *y(t)* is found by Eq. (1.2).

$$P_y(f) = [P_s(f) + P_n(f)] \tag{1.2}$$

In log domain we have the Eq. (1.3)

$$\log[P_y\,(f)] = \left\{\log[P_s(f)] + \log\left[1 + {}^{1}/_{SNR(f)}\right]\right\} \tag{1.3}$$

where SNR(f) indicate Signal to Noise ratio in frequency. From this equation we see that at low *SNR* values the degradation is very high [11]. As a result noise robustness is an important challenge for automatic speech recognition (ASR).

In order to remove the additive noise from speech signal a number of different researches have been conducted. Like when the noise is additive and stationary, and if one can estimate the average noise spectrum, a widely used technique for noise removal is linear spectral subtraction (SS). SS attempts to remove noise effects by subtracting the average magnitude spectrum of the noise$\overline{N}\ (e^{jw})$ from that of the observed signal $Y(e^{jw})$[12]:

$$\left|S\left(e^{jw}\right)\right| = \left|Y\left(e^{jw}\right) - \overline{N}\left(e^{jw}\right)\right| \tag{1.4}$$

where $S\left(e^{jw}\right)$ is the estimate of the speech spectrum. Clearly, this is not valid when the signal and noise are not in phase. Nonlinear spectral subtraction (NSS) is an alternative method which can improve ASR noise robustness by using a subtraction factor that is a function of SNR[12]:

$$\left|S\left(e^{jw}\right)\right| = \left|Y\left(e^{jw}\right) - \frac{\overline{N}\left(e^{jw}\right)}{1+\gamma SNR\left(e^{jw}\right)}\right| \tag{1.5}$$

where $\boldsymbol{\gamma}$ is a tunable parameter.

In general, noise robustness is typically handled by the acoustic model often a hidden Markov model (HMM)], and/or at the front-end (feature extraction) [12]. As a result in this thesis we attempt to improve robustness to additive noise by proposing non-uniform sampling based feature extraction method for robust ASR. Discussion about the non-uniform sampling effect, advantages and usage is covered in chapter two.

1.3. Thesis Objectives

The general objective of this research is to study, analyze and test the effect of non-uniform sampling based feature extraction method for speech recognition system for noisy condition.

Specific Objective

The specific objectives of this thesis are:-

- To investigate and propose new feature extraction method for robust speech recognition system using non-uniform sampling method.
- To compare the proposed method performance in the speech recognition system with the MFCCs based speech recognition system under
    - ✓ Normal external surrounding condition
    - ✓ Additive white Gaussian noise condition
    - ✓ Additive Pink noise condition
    - ✓ Additive Volvo noise condition, for small vocabulary.

1.4. Previous Related Research

Research has been conducted in order to modify the front-end analyzer to better capture the speech information and incorporate the temporal characteristics in the feature set.

Previous studies have not been found related to non-uniform sampling of a signal at time domain for speech recognition application. But a number of researches have been done on non-uniform sampling and robust speech recognition separately. Some of this are:-

Bilinskis [2] showed the use of non-uniform sampling for Digital Alias Signal Processing (DASP). He build ADC based on the hybrid of non-uniform sampling and uniform sampling and showed the use of non-uniform application at wider frequency range.

Marvasti[4] used non-uniform samples that he found by using zero crossing and reconstructed the original signal using iterative algorithm like by spline methods.

Shrawankar and Thakare [13] studied different feature extraction methods of speech recognition. In their paper, they discussed some feature extraction techniques with their pros and cons and developed Autocorrelation Mel Frequency Cepstral Coefficients (AMFCC) by combining different techniques which can be used when MFCC fail on the noisy environment .

The following researchers used the general concept of sampling non-uniformly at different meaning in their researches for different application.

F.Soong [14] proposed nonuniform sampling approach for creating word reference patterns for speech recognition. This work showed a non-uniform sampling approach seems to be a natural choice for potentially improving the recognizer performance and stated non-uniform sampling suffers high uncertainty in its spectral estimate since the high variance regions are sampled more often.

Karnjanadecha [15] investigated a non-uniform sampling method for spectral/temporal feature extraction in speech recognition. In his paper, he proposed a variable block length and/or blocks spacing and modeled the signal with non-uniform sampling of feature and got a better result as compared to fixed block length.

In general, there was not much previous research done to investigate the effect of non-uniform sampling at time domain for feature extraction stage of speech recognition, we were motivated to work on this topic in addition to the advantage of non-uniform sampling.

1.5. Methodology and Scope of the Thesis

The methods used on working this thesis were as follow. First, literature review about non-uniform sampling and speech recognition was conducted. Second, we combined non-uniform sampling method into speech recognition and modeled the system mathematically then we programmed the system in Matlab 7.0. After that we trained and tested the speech recognition system and finally we collected the results and gave conclusion.

Since the objective of this thesis is to see the effect of non-uniform sampling on speech recognition performance, an isolated word speech recognition using Gaussian Mixture Model was build. It was trained with small vocabulary for words between (zero to ten) at normal external surrounding and at additive 10 dB SNR white Gaussian noise taking 440 utterances from 10

persons. The testing was performed for the eleven words at normal external surrounding and for different SNR at additive white Gaussian noise and also for the word 'four' and 'five' the test was done at additive pink and Volvo noise for both training cases.

1.6. Thesis Organization

This initial chapter of the thesis provided an introduction to ASR technology and non-uniform sampling. It focused on the main challenges in the ASR field so as to present the fundamentals of ASR. Then the chapter introduced the main objective of this thesis work and explained the earlier related research. Finally, the scope of thesis discussed with the methods used on constructing this thesis.

Chapter Two first presents an overview of non-uniform sampling. Next, the chapter covers the approaches used for non-uniform sampling and approaches to processing of the non-uniform samples follow. Finally, the chapter deals with the application on which the non-uniform sampling method is used.

Chapter Three introduces the popular speech feature coefficients used to extract spectral information from acoustic signals. The computation of Mel-frequency Cepstral Coefficients (MFCCs) is explained using a step-by-step approach. Then, the dynamic delta parameters (delta-MFCC) are derived from the MFCC parameters to supplement information of MFCC parameters with information on its rate of change over time.

Chapter Four discusses about the proposed Non-Uniform Mel-frequency Cepstral Coefficients (NUMFCCs) step by step from pre-processing to computation of Cepstral coefficients. The rest of the chapter deals with the vector quantization (VQ) and Gaussian Mixture Model (GMM).

Chapter Five provides overview of the system implementation of the speech recognition and presents the results.

Chapter Six presents conclusions and recommendations for future work.

CHAPTER TWO: NON-UNIFORM SAMPLING

2.1. Introduction to Non-Uniform Sampling

When sampling is mentioned in the context of Digital Signal Processing (DSP) , usually it is assumed that the sampling considered is deterministic and uniform. The model of sampling according to which signal samples are considered by time intervals with a constant and known duration is the most popular. Researchers were interested in approximation (of analog signals by discrete time signals) that is as close as possible using uniform sampling. These efforts culminated in Whittaker-Kotel'nikov-Nyquist--Shannon (WKNS) sampling theorem [1]. This theorem ensures that the original band-limited signal can be recovered from the uniform samples provided the sampling rate is at least equal to (or greater than) twice the highest frequency to which the original analog signal is band-limited. If the sampling rate is below the Nyquist rate, reconstruction error, called aliasing occurs. Since a signal is represented by finitely many quantization levels, there is a reconstruction error (of original signal by the samples) due to quantization noise.

As the highest achievable sampling rate depends on the technical perfection of the electronic devices available for implementation of signal analog-to-digital conversions, the sampling theorem evidently determines the boundary limiting the field where signals could be processed in a digital manner. How wide that field is at any given moment apparently depends on the achievable quality of the microelectronic elements being produced at that time.

These considerations and widely accepted conclusions are of course true. However, a significant fact is more often than not overlooked. It is the fact that the conclusions of the sampling theorem in engineering practice (WKNS) are often considered in a simplified form.

The point is that this basic version of the theorem has been derived and actually holds fully only in cases where band-limited signals are sampled equidistantly. In other words, the simplified interpretation of the theorem is valid for the classical DSP approach developed a long time ago.

As use of the uniform sampling approach is still overwhelming, it is often not realized that there could be any other type of digital version of the respective analog signals.
The conditions under which randomized (non-uniform) signal processing might take place differ. This operational mode might be either enforced or chosen willingly. The randomness present at some stage of signal processing might be either an unavoidable reality caused by uncontrollable signal acquisition conditions or it may be introduced deliberately in order to obtain some benefits, like avoiding aliasing. Indeed, sometimes signals are sampled non-uniformly simply because they can be observed only at some unpredictable random time instants. Then it is virtually impossible to control the conditions used to obtain the signal sample values [3].

At a first glance deliberate randomization of signal processing is perceived as the addition of a noise to the signal but that interpretation seems to be absurd. Actually the essence of randomization of sampling and quantization is different. Randomization, in fact, means substitution of some fixed or stationary signal conversion procedures by their variable or nonstationary versions. In the case of sampling randomization, equidistant signal sample taking is transformed into a non-uniform procedure of sample value taking in a nonequidistant way. The distances between sampling instants are then aried in accordance with certain rules [16-17].

A specific non-uniform sampling procedure, based on waveform crossings, is then performed in order to digitize signals as close to each of the distributed signal sources as possible, and to do it in a very simple low-power way [2, 18-19].

Traditionally the negative consequences caused by aliasing are usually accepted as unavoidable. Actually, this is not so. It is possible to avoid overlapping of signal spectral components and to distinguish them without increasing the mean sampling rate.

Let's take an example, assume that a digital data set, representing a signal sample value sequence, is given. As shown graphically in Figure 2.1. Looking at the signal samples and trying to imagine how the signal looks from where they have been taken. It is hard to do that. The digital sample values have to be processed to reconstruct the original signal they belong to. In this particular case,

the indicated sine function 1 (solid line) is found to fit the data. Therefore it should be the signal from which the sample values have been taken. However, if the reconstruction process is continued, it becomes clear that there are other sinusoids at differing frequencies, which also can be drawn exactly through the same sample value points as the first. All these sinusoids (dotted curves) are aliases and overlapping of their sample values is aliasing. The aliasing effect leads to an uncertainty. Indeed, the given sine waves of different frequencies fit equally well all of the indicated frequencies.

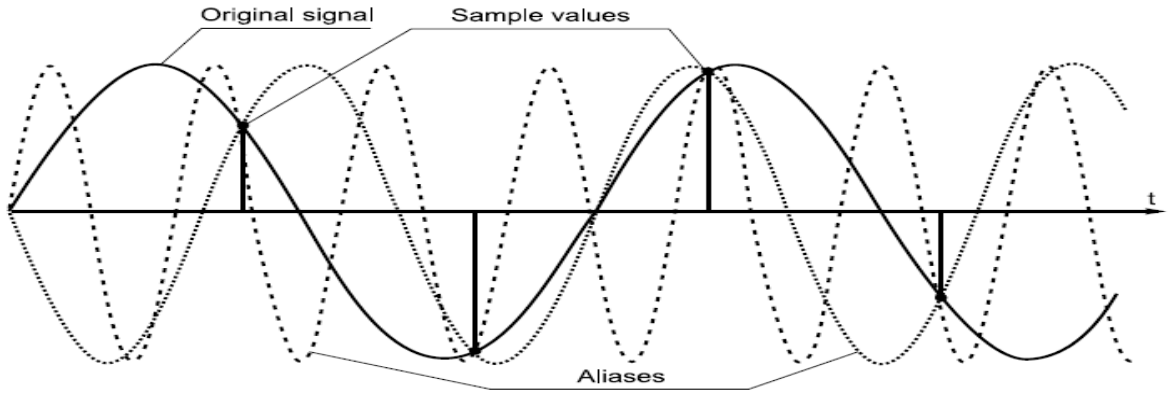


Figure 2.1 Overlapping of a periodically sampled signal component and its possible aliases [2]

To ensure that a digital signal can provide the correct original analog signal, the bandwidth of the signal should not exceed half of the sampling frequency. If all spectral components outside the limited frequency band are filtered off the original signal or more signal sample values are taken within the same time interval, there would be no uncertainty. Either one of these possible actions impose limitations on the bandwidth of the signal, which could be sampled at the given sampling frequency without corruptions due to aliasing. Apparently, if some other way could be found to avoid aliasing, a special application of oriented digital processing of signals would be possible in a much broader frequency range. That would open up a broad area of new beneficial digital signal processing applications.

Trying to vary the time intervals of Figure 2.1 in order to find an alternative approach, the sample values of all indicated sinusoidal curves become different, even at small changes in distances between them. It means that taking signal sample values irregularly disturbs the aliasing phenomenon as shown in Figure 2.2 The signal shown (solid line) is the same one as given in

Figure 2.1. The lower frequency sine function is again sampled and the corresponding data set is obtained.

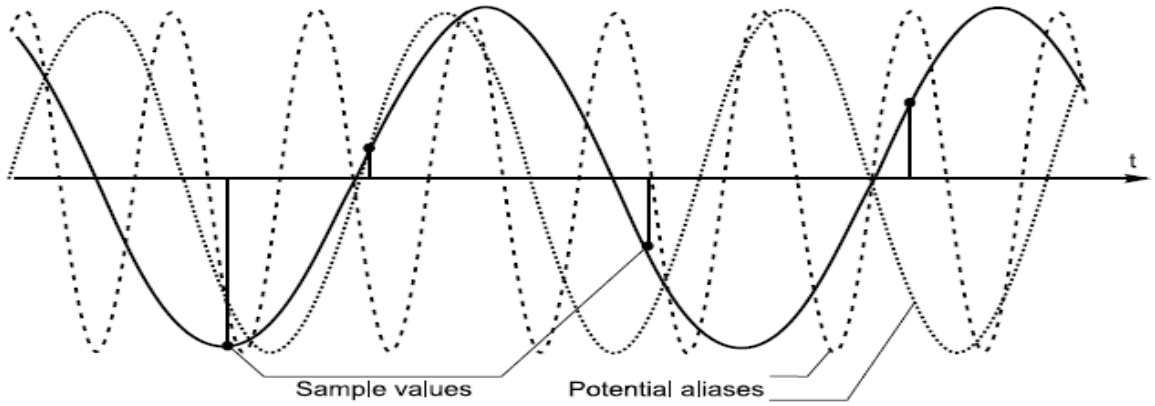


Figure 2.2 Reconstruction of the original signal from non-uniform samples with out aliases[2]

However, the distances between the sampling instants along the time axis now differ. They are irregular. Amazingly, this proves to be very useful. Indeed, as can easily be seen, now only one sine function can be drawn exactly through the points indicating the signal sample values. The sinusoidal curves at other frequencies simply do not fit them. The results of this simple experiment suggest that the digital signals formed by using the non-uniform sampling operation should have features strongly differing from typical features of the digital signals obtained in the cases when signals are sampled uniformly[2].

Studies of this kind of sampling show that non-uniform sampling of sinusoids at different frequencies provides differing data sets [2,4,16,20]. Therefore irregularly or non-uniformly sampled signals have no completely overlapping aliases like those observed at uniform sampling. Because of these non-uniform sampling open up the possibility of distinguishing all spectral components of the signal, even if their frequencies substantially exceed the mean sampling rate. As the result, the other advantages of non-uniform sampling is the expectation of the spectrum of a properly randomly sampled signal coincides with the spectrum of the respective analog signal even if the mean sampling rate is considerably below the upper frequency of the signal spectrum. Thus non-uniformity of sampling for spectral analysis applications seems to be very useful, as the following example indicates.

In the Figure 2.3, Discrete Fourier Transform is used to calculate the spectrum which shows spectrogram results. The spectrograms shown below display the spectrum of a noisy signal containing three sinusoidal components at the indicated frequencies. The Signal to Noise Ratio (SNR) is 10 dB. In periodic sampling ($t_n = nT$) case (Figure 2.3a) there are aliases for signal components. The second spectrogram (Figure 2.3b) was obtained simply by changing the sampling mode from uniform to additive non-uniform. As can be seen, non-uniformity of sampling in this case efficiently improves the spectrogram, allowing to determine true signal components at frequencies exceeding the half of mean sampling rate[2,5].

The increased background noise level in the second spectrogram is mainly caused by the method used to process the non-uniform samples but this considered as disadvantage of the use of non-uniform sampling. To suppress it the more complicated signal processing procedures should be applied for solving of spectral analysis tasks[2,5,21-23]. In our case we used NDFT as discussed in section 2.3.

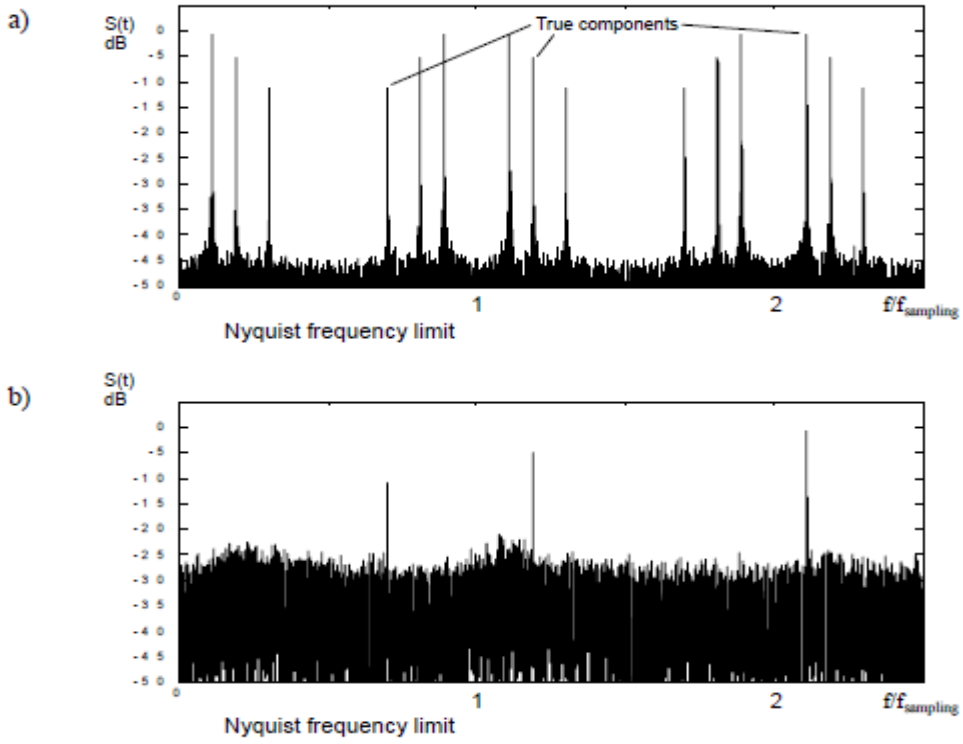


Figure 2.3. Spectrograms of composite noisy signal a)uniform sampling;
b) non-uniform sampling[2,5]

And also it is possible to reconstruct the signal waveform from such a sparse sequence of non-uniform sample values when the signal is ergodic and quasi-stationary. When the parameters do not vary during the time period it is being observed, a reduced number of independent sample values are needed to reconstruct it by estimating all three parameters (amplitude, frequency and phase angle) of all signal components. So non-uniform sampling makes it possible to compress data significantly, so much simpler electronic circuitry is needed to complete the task[3].

In the case of spares non-uniform sampling, the limitations on the lowest sampling rate are imposed by signal parameter variation dynamics rather than by the upper frequency of their spectra. This changes the attitude to establishing the required parameters for used sampling drivers. The signal nonstationarity issue becomes the primary consideration and analysis of the expected signal behavior has to be carried out to determine the requirements for the designs of the sampling driver including the required mean sampling rate [2].

In general, the digital signals formed in the case of non-uniform sampling should contain two times more bits than the digital signals obtained in the case of uniform sampling. The necessity to measure the time instants of sampling and to spend two times more bits for a digital description of a signal is difficult in itself, but this condition is especially worrisome in the light of the additional computation complexity caused by it. However, more detailed consideration of this situation reveals that there is a much better approach to this problem; in fact it is possible to avoid doubling data volumes that have to be processed when sampling is performed non-uniformly. A special approach to realization of non-uniform sampling has to be used for that.

In principle, there are two options: measuring each sampling instant digitally or performing the sampling operation at predetermined time instants.

The first option, measuring digitally the instants exactly when each signal sample has been taken, is a quite demanding engineering task, especially if the required time resolution is taken into account. Indeed, the period, for example, of the 1 GHz signal is 1 nanosecond. To sample such a signal, the smallest time digit obviously has to be equal to a few picoseconds [2].

The second option is much better. To realize it, the required sampling point process has to be generated and the instants when the signal sample values have to be taken need to be memorized. This information is then used both for driving the sampler and for digital processing of the sampled signal.

Therefore, only one digital number per sample value taken appears at the output of the digitizer performing the sampling operation in this case. As it is much easier to realize this second approach in timing non-uniform sampling events, typically it is now almost always used.

According to this approach, the signal sample values are to be taken at exactly predetermined time instants. However, in reality there is always some discrepancy between the dictated and the actual sampling instants. In other words, sampling instants jitter. Apparently this jittering has to be kept within certain margins. The impact of sampling instant jittering depends on the specific signal processing taking place. Not only does the average noise level increase as a result of such jittering but it might also lead to peaks at some spurious frequencies if it is not carefully controlled[2,16].

Generally ADC can be improved or its performance can be increased using non-uniform sampling. Since improving ADC helps to increase the representation of original signal (speech) which in other ways increases the performance of the feature extraction stages, in this thesis we deliberately introduce non-uniform sampling in order to get the advantage of the non-uniform sampling at feature extraction stage. In addition, since the human hearing system has non-uniform spectral sensitivity and the spectrum of a non-uniform sampled signal coincides with the spectrum of the respective analog signal without aliasing as shown in Figure 2.3b might helps to autofocus on most reliable parts of the spectrum in noisy cases, we used non-uniform sampling for feature extraction stage in this thesis.

2.2. Approaches to Non-Uniform Sampling

A sampling process may also be considered as a sequence of events taking place at some time instants t_k. Graphically this process can be depicted as a stream of points or, in other words, as a point process. There are various sampling point processes with significantly differing features.
Mathematical descriptions of sampling and sampled signals $x(t_k)$ of the original signal $x(t)$ are often based on the set of Dirac distributions:

$$\mathrm{b}u(t) = \sum_{k=-\infty}^{\infty} \delta(t - t_k) \tag{2.1}$$

where $\delta\,(t - t_k)$ is the delta function. Then a sampled signal can be given as

$$x(t_k) = x(t)u(t) \tag{2.2}$$

Non-uniform sampling is one of the sampling point processes, which is not most often observed. This non-uniform sampling realized either as deliberately randomized or deliberately pseudo-randomized operation.

Randomized Sampling is sometimes appropriate for applications when it is undesirable to distort signals by filtering off signal components at frequencies exceeding half of the sampling rate. However, the signal sample values at randomized sampling are taken at unknown random time instants. Therefore application of this kind of randomized sampling is limited to the relatively rare cases where information about exact sampling instants is not relevant. Estimation of some signal parameters, including signal power, and measurements of time intervals with a sub nanosecond time resolution might be mentioned as some application examples for such a sampling approach [3]. This type of sampling might be randomized either directly or indirectly.

In direct sampling randomization, signal sample values are taken at random time instants linked to pulses in a specially generated random sampling pulse process.
For indirect sampling randomization, a periodic reference waveform rather than a sequence of pulses formed at random time an instant is used to perform the sampling process. The sampling

process itself is then defined at the signal and the reference waveform crossings. It has been found to be quite useful for temporal and spatial data acquisition from a large quantity of signal sources

Pseudo-randomized sampling is the basic anti-aliasing technique. The indications for its use are the same as for the randomized sampling except that the sampling instants in this case are predetermined with high resolution and precision [2-3].

In general, the approaches to non-uniform sampling are, directly randomized non-uniform sampling, indirectly randomized non-uniform sampling, pseudo-randomized non-uniform sampling, hybrid periodic/non-uniform sampling and hybrid double periodic/non-uniform sampling [2].

Directly randomized non-uniform sampling is one of the oldest approach in which the signal parameter estimation not require sample value timing. It needs only sub nanosecond resolution time interval and used for related parameter digital measurements. It also requires simple technical implementation of sampling and simple design of processing device. And it has limited application range.

Indirectly randomized non-uniform sampling is the other approach which is based on reference function crossings. It reduced complex low-power remote sampler designs and used for data compression and its application limited to data acquisition from relatively low frequency.

Pseudo-randomized non-uniform sampling is used for universal digital alias-free signal processing and has wide range of alias-free versatile digital processing applications but it increase complexity processing. The existing fast algorithm is not applicable for processing it, which is its drawback.

Hybrid uniform/non-uniform sampling approach is based on the hybrid between periodic uniform sampling and non-uniform sampling. It has versatile signal analysis and waveform reconstruction. It reduces the non-uniform sampling drawbacks and widens the dynamic range. It needs digital signal preconditioning for reduced complexity processing and suited better for processing adapted

to sampling irregularities. It increase complexity processing in comparison with the case of periodic sampling and the existed fast algorithms is not applicable for this approaches too.

Hybrid double periodic/non-uniform sampling approach is based on two ADC connected in parallel on which both the sampling process repeated. In this approach signal processing require closely placed sample values taking and it used for high resolution correlation analysis and spectrum analysis.

Although the non-uniform sampling techniques differ, all of them are based on taking signal sample values irregularly in time.

Taking Direct Randomization approaches, the statistics of the signal sample taking timing process is signal independent. There are two types of direct randomization, additive random point process and periodic jittering.

Fluctuation of sampling instants is a fairly common occurrence. It can even be said that it is always present, although more often than not it is insignificant enough to be ignored. How seriously fluctuation affects the precision of signal processing depends on the kind of processing being performed and, of course, the magnitude of the fluctuations. The randomized sampling model used for analysis of this problem is known as periodic sampling with jitter [17].
The sampling instants $\{t_k\}$ in it are given by

$$t_k = k * T + \tau_k \ , k = 0 \ ,1, 2 \ldots ., \qquad (2.3)$$

where T is period, $\{\tau_k\}$ is a family of independent identically distributed random variables with zero mean.
Considering another aspect which different from deliberate randomization of sampling, fluctuations in sampling instants may turn out to be highly undesirable because they may introduce significant bias and random errors. The model briefly described above is worth considering in order finding an answer to the question of how harmful these fluctuations really are under given conditions. The errors caused by such fluctuations depend considerably on the specific digitized signal processing algorithms applied. Some, like algorithms for the estimation of

a number of averaged signal parameters (such as the mean power, the higher moments of signal distributions, etc.), are relatively insensitive, while others are more sensitive [17].

However, this method in fact has a number of substantial disadvantages that prevent its wide application. This are, technical implementations of this sampling scheme should be wideband, even at relatively low mean sampling rates when the intervals between t_k and t_{k+1} very short, sometimes the randomness introduced at sampling cannot be small as a result it may affect the spectrogram as noise and also sometimes the statistical errors resulting from the relatively powerful randomness introduced at sampling are significant.

In the case of additive random sampling, signal samples are taken at instants

$$t_k = t_{k-1} + \tau_k, \quad k = 0,1,2,,\ldots\ldots \tag{2.4}$$

where τ_k is a realization of a random variable. This random sampling scheme, suggested by Shapiro and Silverman [16], was originally based on the assumption that successive sampling intervals $\{\tau_k, \ \tau_{k+1}\}$ were statistically independent and identically distributed. They were characterized first of all by their mean value μ and a standard deviation σ. obviously, the mean sampling rate is equal to $1/\mu$.

Considering the time intervals $[0, t_k] = \tau_1 + \tau_2 + \text{---} + \tau_k$. These random variables are characterized by their respective probability density functions $\{p_k(t)\}$.

Then

$$\begin{aligned} p_1(t) &= p_\tau(t), \\ p_2(t) &= p_1(t) * p_\tau(t), \\ &\ldots\ldots\ldots\ldots\ldots \\ p_k(t) &= p_{k-1}(t) * p_\tau(t), \end{aligned} \tag{2.5}$$

where the asterisk * denotes the composition operation. On the grounds of the central limit theorem in statistics, a very important conclusion is made.

As the random variable $[0, t_k]$ represents the net result of a linear sum of k statistically independent constituent variables $\tau_1, \tau_2, \ldots, \tau_k$, then whatever probability density functions these

constituent variables may have, the probability density of $\tau_1 + \tau_2 + \text{---} + \tau_k = [0, t_k]$ will approach the normal form as k approaches infinity.

Consequently, whenever the additive random sampling scheme is applied, the density functions $p\tau(t)$ may vary from case to case within wide boundaries without worsening sampling conditions, because the sampling point density function $p(t)$ with t increasing will always tend to the constant level $1/\mu$. When sampling is stationary, i.e. when t_0 is a properly distributed random variable, $p(t) = 1/\mu$ for the whole time interval over which a signal is sampled. This means that all of the instantaneous signal values are sampled with equal probability. Taking T_a some time value

$$p(t)_{|t \geq T_a} = \frac{1}{\mu}. \tag{2.6}$$

In general the satisfaction of Eq. (2.6) provides unbiased estimation of signal spectra and unbiased reconstruction of signal waveforms when the reconstructions carried out on the basis of the signal spectral component estimations and the sampling procedure should be performed in such a way that all parts of the signals should be sampled with equal probability which is obviously relevant for most signal processing cases [16]. Uniform sampling with jitter, in general, does not meet these conditions.

The motivation considered so far for randomizing of sampling has been the widening of the frequency range for fully digital signal processing. That surely is a good enough argument in favor of deliberate direct randomizing of sampling as it leads to the irregularities of the sample taking process vital for elimination of aliasing. However, avoidance of aliasing is not the only possible reason why sometimes it makes sense to use this approach. There might also be some other motivational factors, like complexity reduction of systems for data acquisition, transmission and processing and in our case for feature extraction.

In some other cases randomizing of sampling occurs indirectly. This approach to signal digitization has been preferred, for example, in cases of data acquisition from multiple signal sources on a large scale [2]. Threshold- crossing sampling has been used in addition to amplitude sampling in indirect sampling approaches [18-19].

The term 'threshold-crossing sampling' covers many things. Specifically, it encompasses sampling based on zero crossings, constant level crossings, multiple level crossings and input $x(t)$ and time-variant reference signal $r(t)$ crossings. Recovery of the signal sample values from the crossing time instants, straightforward at the zero and constant level crossings, requires some calculations at the time-variant reference function crossings.

The simplest and most popular threshold-sampling variety is based on zero crossings.
Obviously, crossings of zero or any other single level provide information limited with regard to the scale factor, as the signal changes below and above the threshold levels are not reflected in any way at all. The signal sample values are then fixed and only the time instants $\{t_k\}$ when the signal assumes these constant values are detected. Relatively many useful zero-crossing applications have been developed. In particular, this technique serves well in cases where the parameters to be estimated do not depend on the amplitude, e.g. at applications related to signal phase angle and frequency measurements. The direct current component usually has to be extracted from the signal [4, 19].

Evidently multilevel crossings are more informative. For instance, it is possible to use such an approach when performing signal asynchronous quantizing. Analog-to-digital converters built on this basis are specific as the signal sample values in this case are obtained only at the time instants when the signal crosses one of the quantization levels. Therefore the signal sampling operation is randomized, non-uniform and signal dependent. On the one hand, special techniques are needed for processing digital signals formed in this way, which represents a drawback. On the other hand, this approach has lately received a lot of attention as it has a significant power saving potential, essential for many applications [18].

The main attractiveness of the level-crossing sampling approach is the extreme simplicity of the electronic circuits needed for realization of it. Only one comparator is usually exploited for detecting the signal and threshold-crossing events. The crossings repeatedly happening at time instants $\{t_k\}$ are converted into a non-uniformly spaced one-bit stream. However, the quality of

this kind of sampling depends on the performance of the comparator under given specific conditions [2, 18].

If there is virtually no noise present, reaching the equality $x(t_k) = r(t_k)$ (the reference signal) is fixed precisely as occurring at the correct time instant t_k. Unfortunately, when the signal values approach the threshold, the comparator becomes sensitive to noise, so even relatively weak noise affects the outcome of this type of sampling. If there is an additive noise, the signal and noise mixture crosses the threshold sooner or later than the signal itself. Under the impact of the noise, there might even be some repeated crossings observed as 'ringing' of the comparator. As a result the time instant when the signal crosses the threshold is indicated with some error. How large this is depends on various factors, including the threshold-crossing angle. The error decreases if the signal rise is steeper. If the threshold is fixed at a constant level, the crossing angle totally depends on the signal characteristics and nothing can be done to widen this crossing angle. There are various techniques that can be used to avoid the ringing effect.

Threshold-crossing sampling becomes more dynamic if this process is arranged as signal crossings with a time-variable reference function. Then the threshold changes in time and typically covers the whole dynamic range of the signal being sampled. Although the sampling instants then still depend on the signal, the reference function could be used for controlling, to some extent, the sampling process and that helps to resolve the basic inherent problems of the threshold crossing sampling approach [18-19].

For function-crossing sampling, detection of the time instants $\{t_k\}$ at which a signal $x(t)$ intersects a reference waveform is the basic operation. Therefore, it is crucial to realize how this type of timing information could be used to represent sampled signals. The best way of approaching this problem is to focus on finding exactly how the precise signal sample values $x(t_k)$ could be recovered after these time instants $\{t_k\}$ have been fixed. In the case of a well-defined and stabilized reference function $r(t)$, this task of the signal sample value recovery is actually not very difficult. The signal sample values $x(t_k)$ are evidently equal to the corresponding discrete value of the reference function $r(t_k)$. Therefore, the digital version $x(t_k)$ of the original signal $x(t)$ is obtained as a sequence of the reference signal sample values taken at the crossing instants $\{t_k\}$. It

looks like a typical non-uniform sampled signal represented by a sequence of signal sample values randomly located on the time axis at instants $\{t_k\}$. That confirms the point that the sampling process based on the time-variant threshold crossings could be reduced to the typical non-uniform sampling processes.

The sampling process based on the reference function crossings is basically random and, consequently, the sampling intervals are random continuous-value variables. This means that the sampling irregularities are not *a priori* known, which clearly represents a disadvantage, especially in the cases where adapting signal processing to the sampling irregularities is indicated.
As randomizing of the reference function crossings happens unintentionally, there is no guarantee that the non-uniform output signal, representing the outcome of such a function-crossing sampling, will provide effective suppression of aliases unless special arrangements for controlling the sampling irregularities are made. A particular approach to this problem of achieving an alias-suppression is sinusoidal function (sine-wave crossings) [2, 18-19].

Choosing an appropriate type of reference function is vital for effective implementation of the function-crossing sampling. Although various reference functions might be exploited, sinusoidal functions are often preferable as they are narrowband and can be easily generated, stabilized and used for reconstruction of the input signal [2].

The signal sampling process, based on sine-wave crossings, can be arranged in various ways. Direct application of this sampling concept is illustrated in Figure 2.3. According to this, a comparator is used for comparing the signal with the sine-wave reference function. Binary signal 1 is formed at the output of the comparator marking the parts of the signal waveform where it exceeds the reference sine wave. The rising and falling edges of the comparator output signal indicate the intersections of both waveforms occurring at the time instants t_k, $k = 0, 1, 2, \ldots$, when the equality $x(t_k) - r(t_k) = 0$ takes place. Recovery of the signal sample values $x(t_k)$ is performed by reading the values $r(t_k)$ of the reference waveform corresponding to the sampling instants t_k.

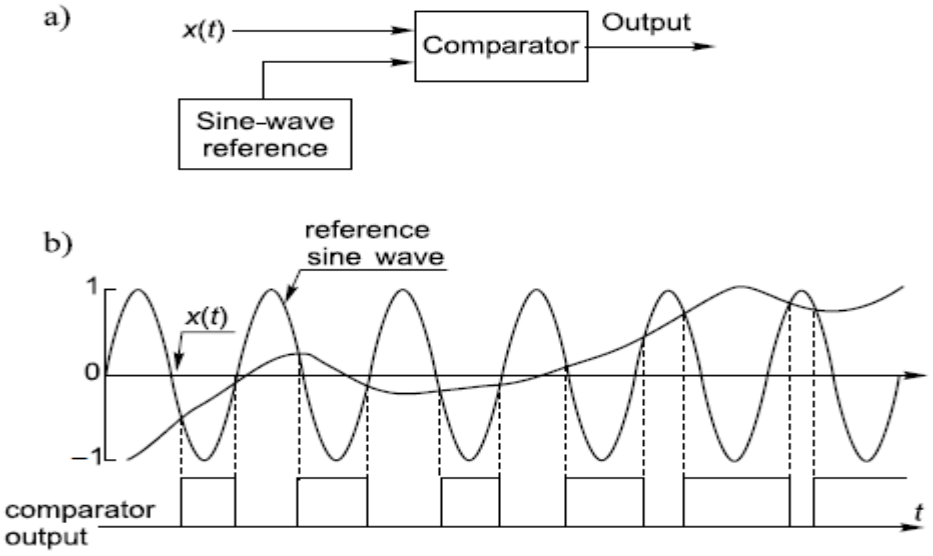


Figure 2.3 Signal sampling based on sine-wave crossings: (a) diagram illustrating realization; (b) time diagram of the involved signal interaction [2]

At the threshold-crossing sampling, the input and reference signal crossings occur randomly in time. Therefore it might seem that this type of indirect sampling randomization suffers from the handicap of not knowing when the signal sample values are taken. That is true, but on the other hand the sample value of the reference function at the crossing instant actually indicates both coordinates of the corresponding signal sample. Indeed, the time instant t_k of taking each signal sample value $x(t_k)$ (and equal reference sine-wave value) functionally depends on the value of this sample. It simply has to be recovered from the corresponding instantaneous value of the reference function $r(t_k) = A*\sin 2\pi*fr*t_k = x(t_k)$, where A is the amplitude of the reference sinusoid. In the case where this type of sampling is realized on the basis of the scheme illustrated in Figure 2.3.

In many cases the power supply for the comparator can be activated only during the time intervals when it is enabled for comparison of the input and reference signals in order to save the power required for the application. And this kind of sampling is used under conditions where the signal is band limited and the frequency of the reference sinusoid exceeds the upper frequency W of the signal spectrum by at least two times. In data acquisition sampling based on sine-wave crossings is better suited for execution of this operation from a distance than the conventional amplitude sampling [2].

Signals sampled on the basis of sine-wave crossings carrying the timing information are relatively insensitive to the impact of surrounding noise. Consequently, they could be transmitted with small distortions over considerably longer distances than the signals reflecting the results of the conventional amplitude sampling. That is obviously crucial for realization of sampling as a remote operation.

It is better if the frequency of the reference sine-wave crossing the signal is relatively high in order to allow more crossing and the steepness of the reference waveform crossing the input signal should be sufficient so that the crossing time instant is fixed more precisely and the impact of noise is less damaging. But the error in crossing time definition increases with this steepness and that translates into more pronounced errors in detection of the corresponding reference signal sample values. Therefore the reference signal frequency has to be chosen with considering the above trade-off. For these reasons, the frequency of the input signals sampled on the basis of sine-wave crossings should be limited so that it does not exceed a certain level. Basically this sampling technique is well suited for handling low-frequency signals. Yet another limitation of a more important nature is related to the fact that the threshold-crossing sampling cannot be regarded as a universal technique for signal digitizing. Actually, this type of sampling is quite specific and its successful usage requires appropriate processing of the digitized signals. The basic difficulty, of course, stems from the unusual non-uniformity of the sampling event spacing, as this leads to the necessity of using special algorithms that take care of all related problems and signal processing in this particular area is not yet sufficiently mature.

As sampling based on sine-wave crossings in general are to be used for digitizing signals at relatively low frequencies, these techniques compete with the traditional periodic sampling positioned rather strongly in this application range [2, 18-19]

Sinusoid-crossing sampling is well suited for executing the sampling operation from a distance and the designs of the microelectronic devices realizing such remote sampling are extremely simple, with all of the consequences that relate to this fact. If this type of remote action sampler is incorporated into a structure of a distributed ADC and is used for building systems for data

acquisition on a large scale, a number of significant advantages could be achieved. Some of them, specifically, are the following: appropriate conditions for sampling signals very close to the signal source; extremely simple designs of the input devices; low power consumption of the data acquisition front devices; one-bit representation of data positioned in time; transmission of data that are relatively insensitive to noise over relatively long distances; fast and robust logic driven sequential activation of the multiple input channels substituting input signal switching (multiplexing) sensitive to errors; a very large number of inputs measured at least in hundreds[2].

2.3. Pre-processing of Non-uniform Sampling

The methods and algorithms initially developed for dealing with non-uniformly sampled signals have proved to be quite useful for resolving essential problems typical for processing signals belonging to the lower frequency range.

Use of non-orthogonal transforms might be mentioned as an example illustrating this. Their first application was reconstruction of non-uniformly sampled signal waveforms. Then it was discovered that they are also good for processing signals at extremely low frequencies. For example, they can be used to remove the negative effect caused by cutting off part of a signal period when, according to the classical definition, the processing should be carried out over a number of integer signal periods (or the periods of their separate components). These transforms are also useful for decomposing signals into several basic parts simultaneously or for extraction of signal components under conditions where the spectra of the components partially overlap [2].

For signals consisting of a number of frequency components, the Fourier Transform (FT) effectively reveals their frequency contents and is generally able to represent the signals with an acceptable resolution divided by equal bandwidth in the frequency domain. The discrete Fourier transform (DFT) is an important tool in digital signal processing.

The N-point DFT of a length-N sequence is given by the frequency samples of the z-transform at N-uniformly spaced points [24]. The proposed non-uniform DFT (NDFT) [25] is the most general form of DFT that can be employed to evaluate the frequency samples at N arbitrary but distinct

points in the z-plane. Since the unitary property is not inherently guaranteed, some fast computation algorithms have been designed by using the approximation algorithm [21-23].

Taking consideration of DFT of uniform sampling case for discrete signal s_n, $s_n = s(t_n)$, $t_n = nT_s$, T_s is time values of a period, n=1 2 …. N

$$S(\omega_m) = \sum_{n=0}^{N-1} s(t_n) e^{-j\omega_m t_n} \tag{2.7}$$

where w= 2 π *f*, $\omega_m = m(2\pi / T)$, m =0, 1, 2,…, N-1 .

Substituting and simplifying the Eq.(2.7) led to the final definition of the DFT

$$S(m) = \sum_{n=0}^{N-1} s_n e^{-j\frac{2\Pi}{N}nm} \tag{2.8}$$

The inverse DFT (IDFT) is :-

$$s_n = \frac{1}{N} \sum_{m=0}^{N-1} S(m) e^{jw_m t_n} \tag{2.9}$$

In general case, the definition of the Non-uniform Discrete Fourier Transform(NDFT) is the same as the one given by Eq. (2.7), taking into consideration that the samples can be taken at irregular intervals both in time (t_n) and/or in frequency (ω_m).

However, in practice, we want to take into consideration a more restricted case, which is the case where the samples are non-uniformly taken in the time domain *t* but uniformly taken in the frequency domain. That is to say that the samples $s(t_n)$ of the non-uniform Fourier transform are taken at multiples of a quantity Δk, which is a fixed quantity in the Fourier domain. The fixed quantity Δk in the uniform case corresponds to 2π/T, T is time of a period. The extension from regular to irregular sampling, therefore, depends on the duration of the signal $s(t_n)$ and not on the fact that the samples time t_n are taken at uniform or non-uniform intervals.

So then the Non-Uniform Discrete Fourier Transform (NDFT) can be expressed as:

$$S(m) = \sum_{n=0}^{N-1} s_n \, e^{-j\Delta k t_n} \tag{2.10}$$

It is shown that the formulation of the NDFT is very similar to the one of the DFT except of the presence of the spatial coordinates t_n instead of the index n. In this case, the NDFT is defined as:

$$S(m) = \sum_{n=0}^{N-1} s_n \, e^{-j\frac{2\Pi}{T} m t_n} \tag{2.11}$$

Shen et al. [15] stated the three types of NDFT and INDFT, and their problems as follows:

$$\alpha_k = \frac{1}{N} \sum_{i=-N/2}^{N/2-1} \widetilde{f_i} \, e^{jw_k x_i} \quad k = 0, 1, 2 \ldots\ldots, N-1, \tag{2.12}$$

$$f_i = \sum_{k=0}^{N-1} \widetilde{\alpha_k} \, e^{-jw_k x_i} \quad i = -\frac{N}{2}, \ldots., \frac{N}{2} - 1, \tag{2.13}$$

where, $f_i, \widetilde{f_i} \in \mathbb{C}$, $\alpha_k, \widetilde{\alpha_k} \in \mathbb{C}$ are directly sampling values in time domain and frequency domain, respectively. $\tilde{f}$,$\tilde{\alpha}$ are kind of weighted f , α according to distribution of x_i ,w_k [21-23].w_k , x_j are sampling points in time domain and frequency domain, respectively. If $w_k=k$, $x_i = 2\pi i / N$ the above equations becomes DFT and IDFT, separately and $f_j = \tilde{f}_j, \alpha_k = \tilde{\alpha}_k$.

The three types of problems according to this sampling method [15] are:-

Type 1: Equal spaced in frequency domain and unequal spaced in time domain

$$\alpha_k = F(f)_k = \frac{1}{N} \sum_{i=-N/2}^{N/2-1} f_i \, e^{jw_k \cdot 2\Pi i / N}, k = 0, \ldots., N-1, \tag{2.14}$$

$$f_i \neq \sum_{k=0}^{N-1} \alpha_k \, e^{-jw_k \cdot 2\Pi i / N}, i = -\frac{N}{2}, \ldots., \frac{N}{2} - 1 \tag{2.15}$$

where $\left\{ \{f_i\}_{i=-N/2}^{\frac{N}{2}-1} \right\}$ are values of equal-spaced sampling points in frequency domain and $\{\alpha_k\}_{k=0}^{N-1}$ are unequally- spaced values in $\{w_k\}_{k=0}^{N-1}$ of time domain.

Type 2: Equal spaced in time domain and unequal spaced in frequency domain,

$$g_i = \mathrm{G}(\beta)_i = \sum_{k=0}^{N-1} \beta_k \, e^{-jkx_i}, i = -\frac{N}{2}, -\frac{N}{2}+1, \ldots\ldots, \frac{N}{2}-1, \tag{2.16}$$

$$\beta_k \neq \frac{1}{N} \sum_{i=-N/2}^{\frac{N}{2}-1} g_i \, e^{jkx_i}, k = 0, \ldots, N-1 \tag{2.17}$$

where $\{\beta_k\}_{k=0}^{N-1}$ are values of equal-spaced sampling points in time domain and $\{g_i\}_{i=-N/2}^{N/2-1}$ are values in unequal-spaced $\{x_i\}_{i-N/2}^{N/2-1}$ of frequency domain.

Type 3: Unequal spaced either in time or frequency domain.

$$\gamma_k \neq \frac{1}{N} \sum_{i=-N/2}^{\frac{N}{2}-1} h_i \, e^{jw_k x_i}, k = 0,1, \ldots., N-1 \tag{2.18}$$

$$h_i \neq \sum_{k=0}^{N-1} \gamma_k \, e^{-jw_k x_i} \;, i = -\frac{N}{2}, -\frac{N}{2}+1, \ldots\ldots, \frac{N}{2}-1, \tag{2.19}$$

where $\{\gamma_k\}_{k=0}^{N-1}$ are values in time domain for unequal-space sampling points $\{w_k\}_{k=0}^{N-1}$ and $\{h_i\}_{i=-N/2}^{N/2-1}$ are values in frequency domain for unequal-spaced $\{X_i\}_{i=-N/2}^{N/2-1}$.

There are five basic problems according to the above types,

(1)Given f, find $\alpha = F(f)$ in equation (2.14)

(2)Given β, find $g = G(\beta)$ in equation (2.16)

(3)Given γ, find h;

(4)Given α, find $f = F^{-1}(\alpha)$ in equation (2.14)

(5)Given g, find $\beta = G^{-1}(g)$ in equation (2.16)

Some works have been done to develop fast algorithm for approximating NDFT (INDFT), in order to take advantage of FFT, which are called NUFFT generally. These typical methods includes such as Gauss interpolation algorithm, least-square error method based on the regular matrix, the min-max approach and other approaches [23,25-26,30-31].

For example, Pottas et al [23] show a fast algorithm for computing an approximation of

$$Y(f_k) = \sum_{m=1}^{M} y_m e^{-i2\pi f_k m/M}, k = 1,2, \dots M, \quad (2.20)$$

at frequencies f_k below the Nyquist rate, where M is the maximum number of samples, y_m is the values of sampling points in time domain , $Y(f_{k)}$ are values in frequency domain. This is done by approximation of the transform as well as zero-padding. An over sampling factor $\alpha > 1$ is introduced and $\hat{Y}(f_k)$ is approximated with

$$\hat{Y}(f_k) = \sum_{m=1}^{\propto M} a_m \emptyset \left(f_k - \frac{m}{\propto M}\right)', \quad (2.21)$$

where $\phi(f)$ is some function with the same period as $Y(f)$. One of the suggested algorithms is summarized in Algorithm 1, and the paper also shows how to use this when the non-uniform grid is instead in time domain. This is also extended to include non-uniform sampling in both time and frequency. It is shown that Algorithm 1 is considerably faster than computing (2.20) directly [23,32]. The characteristics of the computation error are investigated for several choices of the approximation kernel $\phi(f)$.

Algorithm 1:-Nonuniform Fast Fourier Transform (NUFFT)

Given: M, $\propto > 1, f_k, y_m, and\ \phi$(f)

1. Precompute

$$\phi\left(f_k - \frac{m}{\propto M}\right) \qquad m = 1, \dots, \propto M$$

$$c_k = \int \phi(f)\, e^{i2\pi f(k-1)} df, \qquad k = 1, \dots, M,$$

2. Form

$$\widehat{a_k} = \begin{cases} y_k/c_k & ,k=1,\dots\dots,M \\ 0 & k=M+1,\dots,\propto M \end{cases}$$

3. Use the FFT to get

$$a_m = \frac{1}{\propto M} \sum_{k=1}^{\propto M} \widehat{a_k}\, e^{-\frac{i2\pi km}{\alpha M}}, \qquad m = 1, \dots \dots \alpha M,$$

4. Finally

$$\hat{Y}(f_k) = \sum_{m=1}^{\propto M} a_m \phi(f_k - \frac{m}{\propto M}), \quad k = 1, \dots \dots, M$$

is the approximation of the transfer .

However, all the methods exist are approximating numerical methods. For problems (4) and (5), the sampling values can't be used directly; instead, either a weigh or a scale factor is needed, which could not be given accurately. So there is need for fast algorithm for these sampling methods for accurate representation.

One of the processes done in signal processing is calculation of the spectrum of the sampled signal.

The spectra of randomly sampled (non-uniform) signal [2,28-29,33] can be calculated as follow:-

$$S_x(f_n) = \lim_{N\to\infty} \frac{2}{N} \sum_{k=1}^{\widehat{N}} Nx(t_k)\exp\,(-2\pi f_n t_k), \qquad (2.22)$$

where $\widehat{N}$ is the random number of signal samples taken during the time interval $\Theta = N\mu$. $x(t_k)$ in this case is defined by Eq. (2.2).

Noting that the probability density function $\varphi_k(t)$ of the function $\delta(t - t_k)$ is also the same for time intervals in the range $[0, t_k]$. If the additive random sampling, if properly performed, is characterized by the equation

$$\sum_{k=1}^{\infty} \varphi_k(t) = \frac{1}{\mu} = constant \qquad (2.23)$$

where $1/\mu$ denotes the mean sampling rate. If this condition holds, then it follows from Eq.(2.25) that the expectation of the estimated spectra is:-

$$E\left[\lim_{\theta\to\infty} \hat{S}_s(f_n)\right] = \lim_{\theta\to\infty} \frac{2}{\theta} \int_0^{\theta} \sum_{k=1}^{\infty} x(t)\,\exp(-j2\pi f_n t)\,\varphi_k(t)\,dt \qquad (2.24)$$

$$= \lim_{\theta\to\infty} \frac{2}{N} \int_0^{\theta} x(t)\exp\,(-2\pi f_n\, t) \sum_{k=1}^{\infty} \varphi_k\,(t)dt = S_x(f_n). \qquad (2.25)$$

where $\theta = E[t_N]$

If randomized sampling satisfies the condition (2.23), then the expectation of the estimated spectra of randomly sampled quasi-stationary signals coincides with the spectra of the respective original signals [2,28--29,33].

The randomly sampled signals, under specific conditions, might have unique spectral characteristics.

Firstly, the spectra of the randomly sampled signals are not uniform in the frequency domain. Secondly, the spectra of the original signals digitized in this way might well be broadband, with their upper frequencies considerably exceeding the Nyquist limit. Consequently, such randomization of sampling makes it possible to perform spectral analysis of high-frequency signals without distortions of their spectra that are usually observed under the same conditions when the sampling operation is performed periodically [2,28-29,33]. However, these spectral characteristics have been obtained on the assumption that the signals are stationary and that they are observed and digitized infinitely. Nevertheless, the fact that randomization of sampling might lead to elimination of spectral distortions is significant.

2.4. Application of Non-Uniform Sampling

Attempts to eliminate the harmful impact of aliasing have led to the development of advanced digital technologies for signal processing, specifically to the development of an innovative technology called 'Digital Alias-Free Signal Processing', or DASP [2]. DASP is basically based on non-uniform sampling.

This strengthens the competitiveness of digital techniques considerably. The successful use of special digitizing techniques for the elimination of aliasing has been important in showing the significance of digitizing in the whole process of signal digital processing. Many other benefits could be obtained similarly by focusing on digitizing and matching it to the needs of signal processing, as suggested by DASP.

Sometimes it is beneficial to use Non-Uniform sampling techniques for massive data acquisition rather than for the elimination of aliasing. While such massive data acquisition systems might be used widely, the applications related to data acquisition from multiple sources of biomedical signals are especially well suited [25].

And also in many real-time applications, sample values and time stamps are delivered in pairs, where sampling times are non-uniform. Frequency analysis using non-uniform data occurs in

various real life problems and embedded systems, such as vibrational analysis in cars and control of packet network queue lengths [34].

The non-uniform sampling concept also used in Geophysics and Radar application [21].

Some types of inverse problems, arising in medical and other imaging techniques can be presented as a problem of signal reconstruction from its non-uniform frequency domain samples. This method is used in 2D and 3D problems of tomographic reconstruction [36].

In general nowadays, non-uniform sampling has a wider application. Mostly at high frequency range it has remarkable application. In this thesis, we try to use this method in order to see its effect at low frequency on Speech Recognition application.

CHAPTER THREE: FEATURE EXTRACTION

3.1. Introduction

Feature Extraction can be understand us a step used to transform the input data ,which is too large to be processed by an algorithm and also suspected to be notoriously redundant (much data, but not much information), into a reduced representation set of features (also named features vector)[37].

Feature extraction is a crucial part of a high-performance speech recognition system. It has been a research topic since the beginning attempts to automatically recognize speech. It is impossible to construct a state-of-the-art recognizer using low quality features, no matter how well the recognizer performs. On the other hand, if speech features were able to capture all necessary speech characteristics such that each speech unit can be correctly classified, a sophisticated statistical model would be unnecessary for the recognizer. In reality, speech is not pronounced very clearly and the environment may not be quiet. Also some distortions may be introduced during the process [15]. Linear Predictive Coding derived Cepstral Coefficients (LPC-CC), Reflection Coefficients (RC), Mel-frequency Cepstral Coefficients (MFCC) are the most common feature extraction method[13].

Among all popular speech parameters, the most functional and efficient ones extract spectral information (in the frequency domain) from speech, because a more concise and easier analysis of speech can be performed spectrally rather than temporarily (in the time domain) [38]. Although speech signals demonstrate a range of inter-speaker variations for the same utterance in the time domain, this utterance still exhibits consistency in the frequency domain, to some extent. For this reason, spectral analysis is preferred over temporal analysis to discriminate between phonemes and extract speaker-independent features from speech signal [10].

According to speech production model, voiced speech is composed of a convolved combination of the excitation sequence, with the vocal system impulse response [9], as depicted in Figure 3.1.

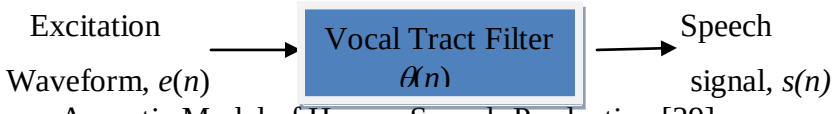


Figure 3.1 Linear Acoustic Model of Human Speech-Production [39]

In this model, the speech signal is expressed as the convolution of an excitation signal $e(n)$ with the vocal tract response $\theta(n)$. The excitation sequence is either a quasiperiodic vocal cord pulse in the case of producing voiced speech or just random noise at the vocal tract constriction, which generates unvoiced speech [40].

Cepstral deconvolution methods helps to decouple the vocal tract response from the excitation response. The decomposition of a speech signal $s(n)$ into the excitation sequence $e(n)$ and the vocal tract function $\theta(n)$ can be described as follows [39]:

$$s(n) = e(n) * \theta(n) \tag{3.1}$$

where the operator , "$*$" represents the convolution operation.

Cepstral analysis has been extensively used for feature extraction in speech recognition. Perhaps, the most popular derivation of cepstral analysis combines the cepstrum with a nonlinear frequency-warping, known as Mel-scale conversion [9]. The resulting coefficients are called Mel-frequency Cepstral Coefficients (MFCC).

3.2 Pre-processing of Speech Signal

Pre-processing of speech signal is necessary for accurate or high performance feature extraction. Determining the onset and termination of speech boundaries is necessary for the incoming word to be as free of "nonspeech" regions as possible to avoid such regions from causing mismatch.

The problem of detecting endpoints would seem to be relatively trivial, but, in fact, it has been found to be very difficult in practice, except in cases of very high signal to ("background") noise ratios. Some of the principal causes of endpoint detection failures are weak fricatives (/f/, /T/,/h/) or voiced fricatives that become unvoiced at the end ("has"), weak plosives at either end *(/p/, /t/, /k/)*, nasals at the end ("gone"), and trailing vowels at the end ("zoo")[9].

The onset and termination of speech boundaries can be determined by combining zero-crossing rate and short-term energy to form appropriate spatial or temporal features. These temporal features can be extracted simply from the sample values of speech signal without transforming the signal into the frequency domain. Short–term energy has been used to distinguish between voiced sounds and unvoiced sounds or silence [9-10]. The absolute short-term energy was used as the

short time energy due to its simple implementation and efficiency. The short-time absolute energy can be computed as follows:

$$E_s = \sum_{n=m-N+1}^{m} |s(n)w(m-n)|, \tag{3.2}$$

where *s(n)* is speech samples, *w(m)* refers to the window with the *N*-length frame duration ending at *n=m*.

In addition to short-time energy zero-crossing rate of speech have been extensively used to detect the endpoints of an utterance. The number of zero-crossings, which is also a useful temporal feature in speech analysis, refers to the number of times speech samples change sign in a given frame. The rate at which zero-crossings occur is a simple measure of the frequency content of a narrowband signal. The short-term zero-crossing measure for the *N*-length frame interval ending at $n = m$, is

$$ZCR = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|sgn[s(n)] - sgn[s(n-1)]}{2} w(m-n), \tag{3.3}$$

The zero-crossing rate is a useful parameter for estimating whether speech is voiced or unvoiced. Voiced speech has most of its energy collected in the lower frequencies, whereas most energy of the unvoiced speech is found in the higher frequencies. Since speech is wideband, the zero-crossing rate corresponds to the average frequency of the primary energy concentration in the signal.

Since high frequencies imply high zero-crossing rates and low frequencies imply low zero-crossing rates, high and low zero-crossing rate correspond to unvoiced and voiced speech, respectively[9-10].

3.3. Mel-Frequency Cepstral Coefficients (MFCCs)

Currently, Mel-frequency Cepstral Coefficients (MFCCs) are regarded as the de facto standard acoustic features for ASR [15]. Most high-performance automatic speech recognizers employ MFCC or a close variation and also it is used in this thesis for comparison purpose to the proposed method. Davis et al. [41] introduced the mel-scale critical-band filtering with a simple set of 20 triangular bandpass filters. They compared several parametric representations (mel-frequency

cepstrum, a linear frequency cepstrum, a linear prediction cepstrum, a linear prediction spectrum, or a set of reflection coefficients) of the acoustic signal with regard to word recognition performance in a syllable-oriented continuous speech recognition system and their primary conclusion was that MFCC parameters outperform other parameter types in speech recognition applications with a compact set of coefficients capturing the information relevant for recognition. The basic conclusion of this paper is still considered valid today although there have been many improvements in speech signal representations since the early 1980s. Figure 3.4 shows the block diagram for the MFCC feature extraction step by step.

The details of the block diagram are described below based on some modification on Voice Toolbox [42] for MFCCs feature extraction implementation in MATLAB 7.0.

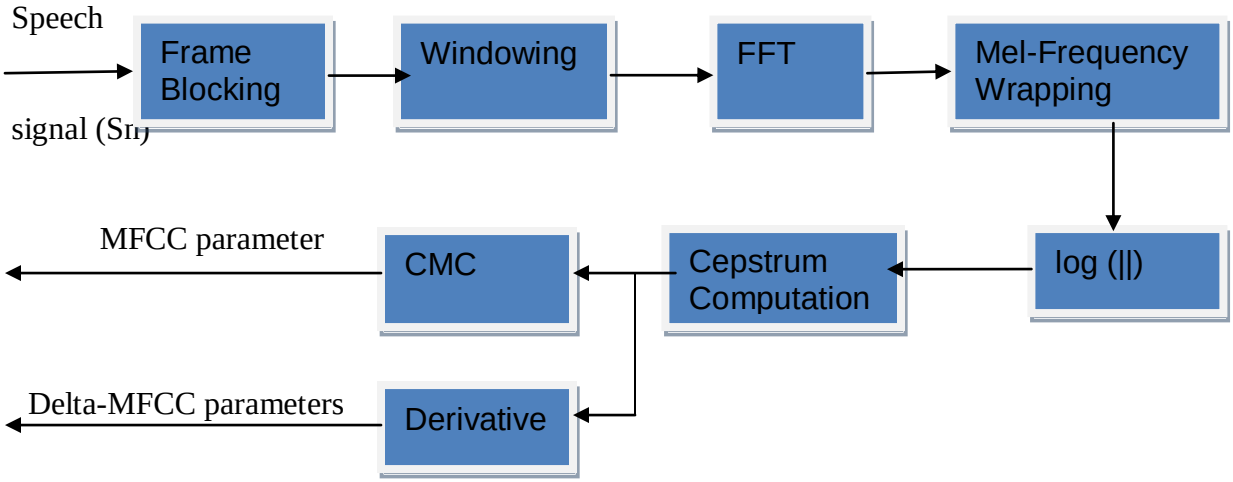


Figure 3.2 Block Diagram for the MFCC Feature Extraction [42].

Frame Blocking: Speech signal shows stationary characteristics in a sufficiently short period of time interval (or called quasi-stationary). For this reason, speech signals are processed in short time intervals. It is divided into frames with sizes generally between 20 and 100 milliseconds [43]. Each frame overlaps its previous frame by a predefined size which helps to smooth the transition from frame to frame.

In the thesis, the cropped speech data is blocked into frames of N = 400 samples (corresponding to 25ms) with 40% overlapping to better capture temporal changes from frame to frame.

Windowing: Windowing all frames is done in order to eliminate discontinuities and spectral distortion at the edges of the frames. Typically, a Hamming window is a good choice in speech

analysis [11,15, 43] and it is used to decrease the signal to zero at the beginning and end of each frame. Thus, a Hamming window of the form was selected:

$$w(n) = \begin{cases} 0.54-0.46cos\left(\frac{2\pi n}{N-1}\right), & n=0,1,\ldots..,N-1. \\ 0 & otherwise \end{cases} \quad (3.4)$$

The window length is chosen as a trade-off between frequency resolution and temporal resolution [6]. It is known that a short duration window allows for the detection of the amplitude decay of formants with better temporal resolution. However, using a longer window allows for better spectral resolution. Although a longer window is useful for capturing fine spectral variations, it is also in contrast to the requirement for the quasi-stationary analysis of signal segments. As a result, a window length equal to the frame length (N = 400) was selected in this thesis.

Fast Fourier Transform (FFT): the Fast Fourier Transform is a fast implementation of the Discrete Fourier Transform (DFT) which converts *N*-samples of frames into frequency spectrum. Recall that the convolution operation in time corresponds to a multiplication in the frequency domain. Thus, Equation (3.2) becomes

$$S(\omega) = E(\omega) \cdot \theta(\omega) \quad (3.5)$$

Note that the complex speech spectrum $S(\omega)$ is composed of a quickly varying part, excitation spectrum $E(\omega)$ (which corresponds to high frequency components) and a slowly-varying part, vocal tract response $\theta(\omega)$ (which corresponds to low frequency components)

Mel-frequency wrapping: The human ear perceives the frequencies non-linearly. Researches show that the scaling is linear up to 1 kHz and logarithmic above that. The Mel-Scale (Melody Scale) filter bank which characterizes the human ear perceiveness of frequency is used as a band pass filtering for this stage of identification. The signals for each frame is passed through Mel-Scale band pass filter to mimic the human ear. Simply, it maps an acoustic frequency to a perceptual frequency scale as follows [39, 43]:

$$F_{Mel} = 2595 * \log_{10}\left(1 + \frac{f(Hz)}{700}\right) \quad (3.6)$$

$$L = int(3\log * sampling\ frequency) \quad (3.7)$$

The mel-scale filter bank implementation used in this thesis includes L= 29 triangular filters non-uniformly spaced along the frequency axis, as shown in Figure 3.3.

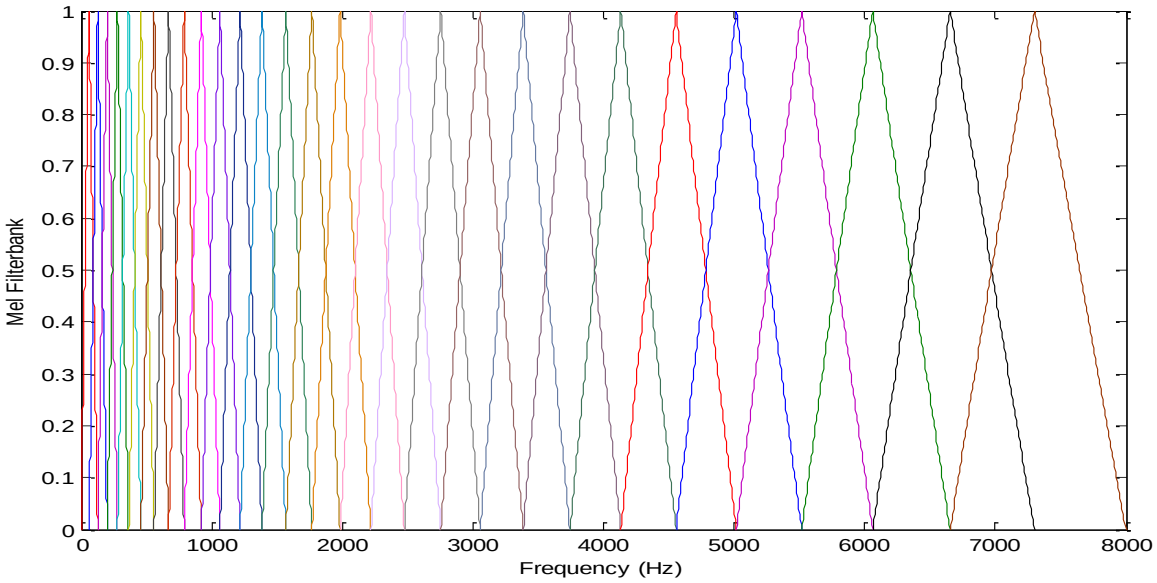


Figure 3.3 Mel-scale Filter Band

Log Energy Computation: This step applies a logarithm transformation to the absolute magnitude of the coefficients obtained after mel-scale conversion. The absolute magnitude operation discards the phase information while the logarithm operation performs a dynamic compression, making feature extraction less sensitive to speaker dependent variations [6].

Considering that the speech signal is real-valued, the logarithm of Eq. (3.5) on both sides leads to

$$\log(|S(\omega)|) = \log(|E(\omega) \cdot \theta(\omega)|) = \log(|E(\omega)|) + \log(|\theta(\omega)|) \quad (3.8)$$

Mel-frequency Cepstrum Computation: Mel-frequency cepstral coefficients are computed by applying the inverse FFT to the logarithm of the magnitude of the filter bank outputs. The inverse DFT reduces to a Discrete Cosine Transform (DCT) operation as the log magnitude spectral of the coefficients are real and symmetric [6]. Moreover, the DCT has the advantage of producing highly uncorrelated features [11, 40, 43]. The resulting mel-frequency cepstral coefficients(C) are computed as follows:

$$c(k) = w(k) \sum_{n=1}^{N} \log(|s_k(n)|) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), k = 1, \dots\dots, L, \quad (3.9)$$

Where

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, & k=1 \\ \sqrt{\frac{2}{N}}, & 2 \le k \le L. \end{cases} \quad (3.10)$$

N = 400 (window length), L = 29 (Number of filters in the mel-scale filter bank),

and $s_k(n)$ represents the output of the k^{th} filter in the filter band. (3.11)

The zero-order MFCC coefficient c_o represents the average log energy of the frame. This coefficient is usually discarded from the feature space because absolute power measures are not reliable in speech recognition [6, 10, 39].

Figure 3.3 plots the real cepstrum computed in the first frame of voiced word 'zero'

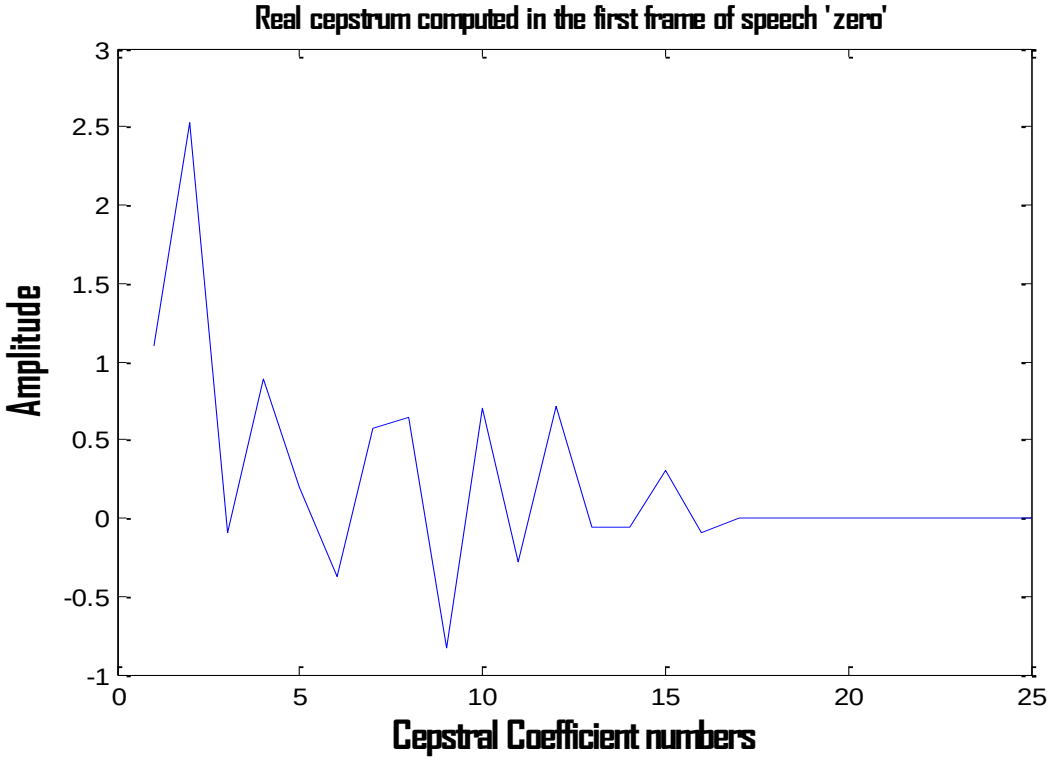


Figure 3.4 Real Cepstrum computed in the first frame of speech 'zero'.

Figure 3.4 illustrates the fact that the dominant contents of the cepstra are located near the origin, and that a small number of cepstrum coefficients can be used to provide enough spectral information about the word 'zero' and the first 8-15 coefficients carry most of the spectral information of the word 'zero' in the first frame. Typically, 8-14 cepstral coefficients and their "derivatives" are used for speech recognition in systems that employ cepstral techniques [8-9].

Therefore, the zero-order coefficient was excluded and the remaining first 13 MFCC coefficients kept for a compact representation in this thesis.

Cepstral Mean Correction (CMC): It is inevitable that a speech signal is subject to some spectral distortions during recording and pre-processing prior to recognition due to microphone offsets, environmental effects (like reverberation) and/or transmission channel [6]. The cepstral mean correction was applied to compensate for such distortion by subtracting the cepstral mean of a frame from the cepstral coefficients for additional robustness in recognition.

Delta-MFCC Computation: Feature vectors consisting of only MFCC parameters appear to provide smooth estimates of the local spectrum. However, these features do not contain information regarding the speech signal dynamic evolution, which also carries relevant information in speech recognition [6, 10]. Therefore, improvements in recognition performance can be obtained by taking into account the dynamic characteristics of the MFCC features [40].

Delta(dynamic)-cepstral features were proposed (in a different form) in [46] to add dynamic information to the static cepstral features. They also improve recognition accuracy by adding a characterization of temporal dependencies to the frames, which are nominally assumed to be statistically independent of one another. However, researchers have also argued that the basic differencing operation is too sensitive to random inter-frame variations and should be replaced by a smoother estimate of the local time-derivative [9-10, 40].

Consequently, most speech recognition systems today incorporate delta features as an add-on to the static parameters such as MFCCs [7]. For these reasons, 13 delta-MFCC parameters were included in this study's set of speech features in addition to the regular MFCC parameters.

The delta-MFCC coefficients are computed using the following regression formula [42]:

$$d_k = \frac{\sum_{\propto=1}^{M} \propto (c_{k+\propto} - c_{k-\propto})}{2\sum_{\propto=1}^{M} \propto^2}, \tag{3.12}$$

where M=4 is the window size, C is the static c*epstra*l coefficients, d is *delta cepstral* coefficients. One potential problem with the use of delta MFCC parameters is that they are not in the same range of amplitude with the static MFCC parameters, which results in poor vector quantization. Through experiments, it was found that the delta parameters approach the range of static parameters when they are weighted by a factor of six [38, 41,44-45].

It is shown in Figure 3.5 a typical example of the resulting 13 MFCC and 13 weighted delta-MFCC parameters extracted for the spoken word "zero."

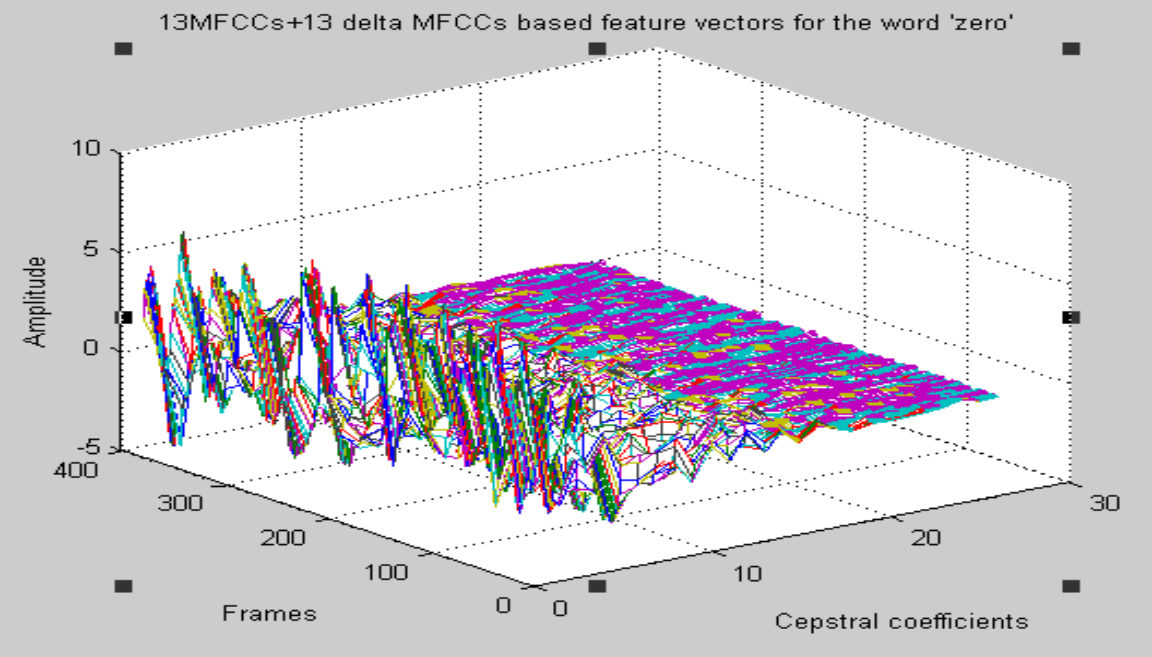


Figure 3.5 MFCC and Delta MFCC Parameters Extracted from spoken word 'zero'

Figure 3.6 shows the cepstrogram for the same 26 parameters.

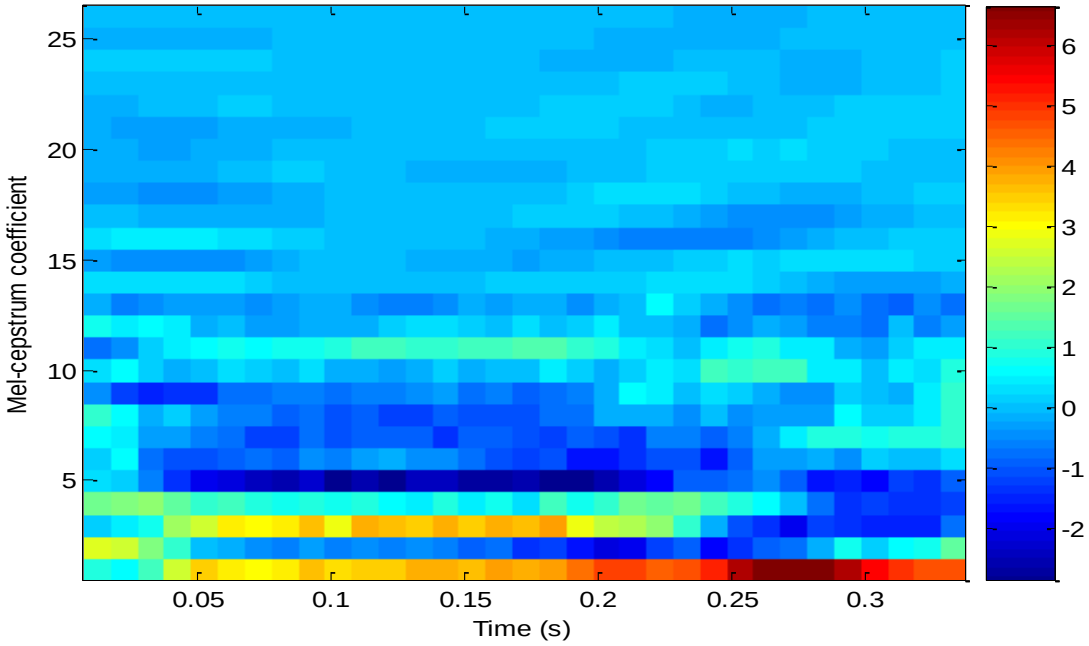


Figure 3.6 Cepstrogram of speech 'zero' based on MFCCs.

CHAPTER FOUR: Non-Uniform Mel-Frequency Cepstral Coefficients (NU-MFCCs), Vector Quantization and Gaussian Mixture Models (GMM)

4.1. Non-Uniform Mel-Frequency Cepstral Coefficients (NU-MFCCs)

Mel Frequency Cepstral coefficients (MFCCs) has been widely accepted and used for speech recognition system as feature extraction techniques but its' performances decrease under noisy condition. Non-Uniform Mel Frequency Cepstral Coefficients (NU-MFCCs) is proposed here by combining the concept of non-uniform sampling and MFCCs in order to form strong feature extraction techniques which allows to increase the performance of the speech recognition system under noisy condition.

In a NU-MFCCs processing, the first step is to record the speech data by over sampling it. Then the non-uniform time spaced samples will be collected with their specific time sampling instant. Consequently the framing is done based on the quasi-stationarity properties of the speech signal within certain time range. Windowing the non-uniform speech sample will follow in order to prevent swamp of relevant energy from the speech samples. After windowing, NDFT is used to find the power spectrum of each frame. Here we perform filter bank processing to the power spectrum, which uses Mel scale. Discrete cosine transformation is applied after converting the power spectrum to log domain in order to compute NU-MFCC coefficients. In order to include the dynamic properties of speech we used delta NU-MFCCs.

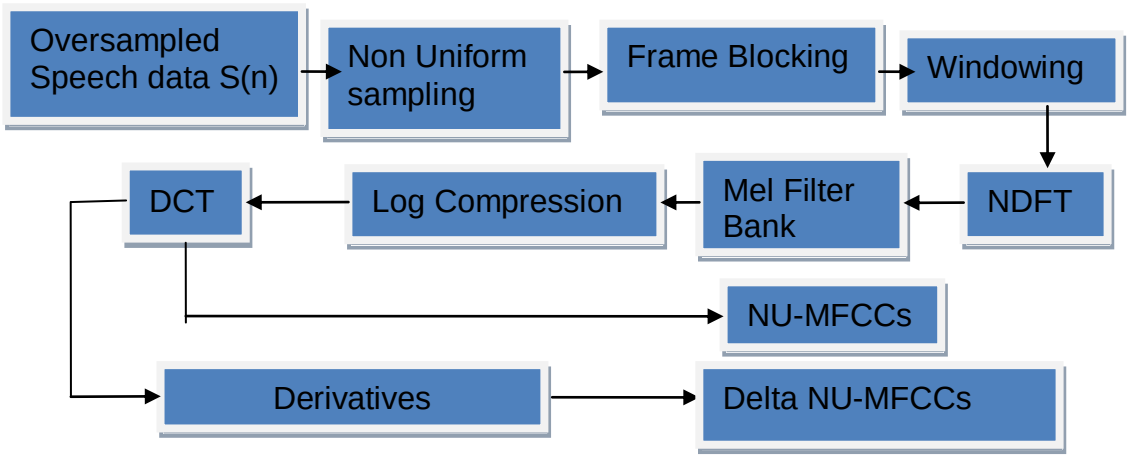


Figure 4.1 NU-MFCCs Processing

4.1.1 Pre-processing

The Nyquist theorem states that a signal must be sampled at least twice as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content will *alias* at a frequency inside the spectrum of interest (passband)[4].

$$f_{nyquest} > 2 \cdot f_{signal} \tag{4.4}$$

where f_{signal} is the highest frequency of interest in the input signal. Sampling frequencies above $f_{nyquist}$ are called 'oversampling'.

Since oversampling helps to get more data points, to avoid aliasing, improves resolution and reduce noise, it is used as a part of the NU-MFCCs method. This is done to replace some of the advantages which can be gated by building Analog to Digital Converter using non-uniform sampling.

Threshold crossing based sampling is used in order to get the non-uniform spaced time sampled data values from the oversampled speech data. Since sine-wave crossing has remarkable usage under low frequency [2, 18, 19], it is used as means of reference signal.

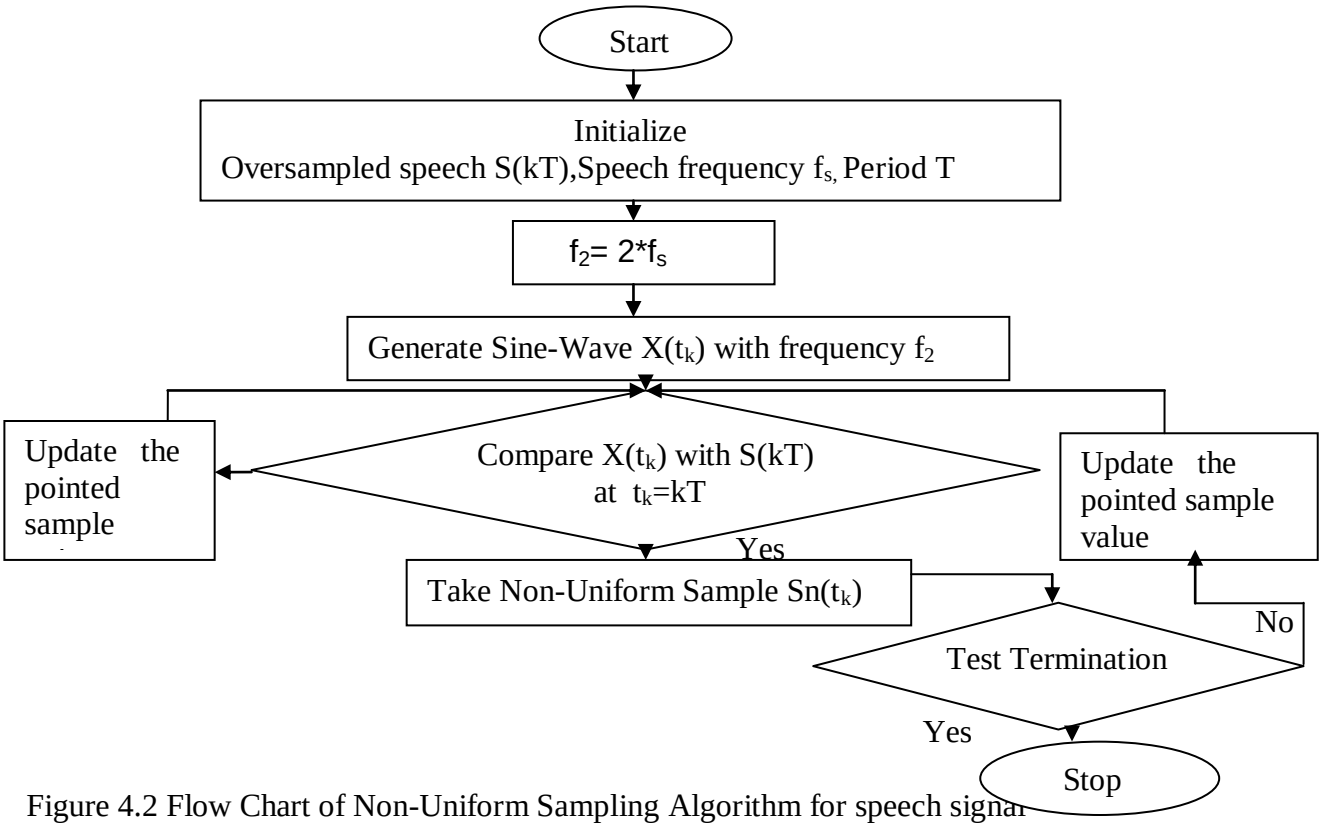


Figure 4.2 Flow Chart of Non-Uniform Sampling Algorithm for speech signal

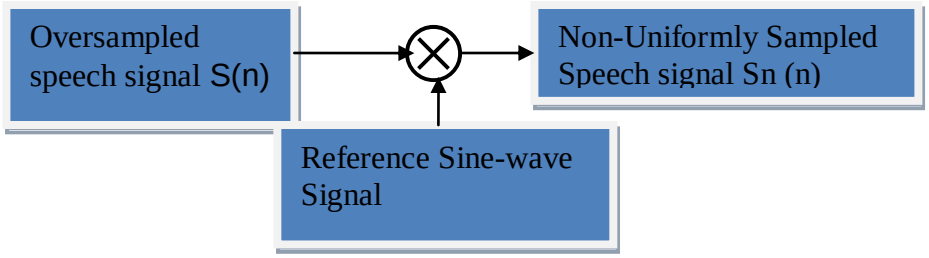


Figure 4.3 Block diagram of Non-Uniform Sampling process

The generated sine-wave signal frequency is need to be greater than the frequency of the original signal in order to allow more crossing and to collects more samples [2,19]. The matlab program of the Figure 4.2 can be found at Appendix A.1 for further understanding.

Taking for example for a spoken word 'one', considering the sampling frequency of the sine-wave be twice of the speech signal, and recording the speech signal at 44.1kHz sampling frequency ,and following the non-uniform sampling process gives the result shown in Figure 4.4.

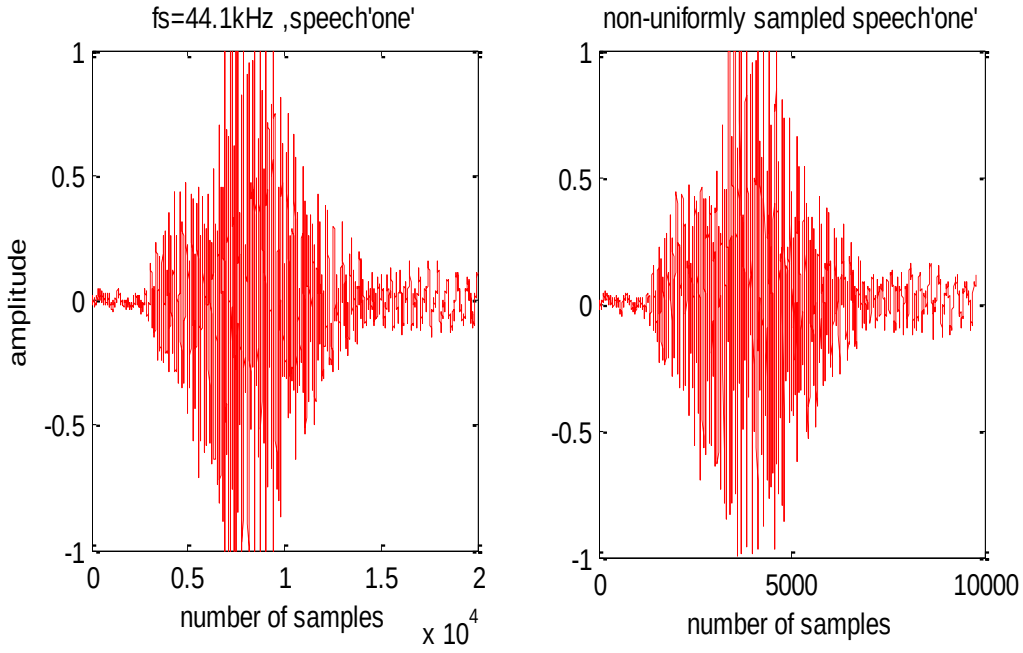


Figure 4.4 Oversampled uniform speech signal and sine-wave referenced non-uniformly sampled signal

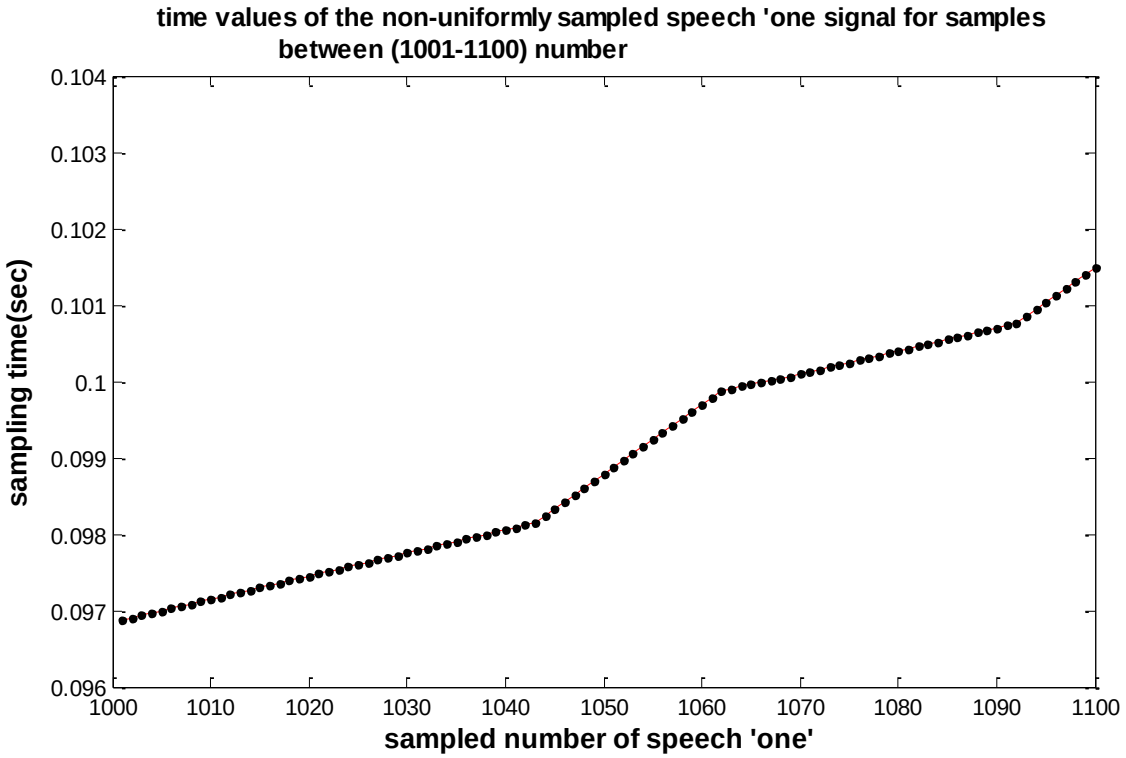


Figure 4.5 sampling time instants of the non-uniform sampled speech ’one’ signal

As shown in Figure 4.4 the sine-wave crossing based non-uniformly sampled signal has less number of samples .The numbers of samples found using 44.1 kHz oversampling frequency of the speech signal and sine-wave having twice the frequency of the speech recorded signals is more or less comparable to number of samples found at sampling frequency of 16 kHz of the uniform case and sometimes to 20kHz. Since most of the speech recognition is done at 16kHz sampling rate and in order to have comparable number of samples, we choose 44.1kHz oversampling frequency for recording the speech signal and non-uniformly sampled it by sine-wave having twice the frequency of the recorded speech signals for NU-MFCCs method.

Figure 4.5 show the non-uniformity between the time instants of the sample values which is gated by the non-uniform sampling process.

4.1.2 Spectral Analysis

If the signal is assumed to be stationary then spectral evaluation of the signal can be reliable. Windowing the speech signal is done since speech characteristics do not change much in a short time period. Hence, window size and shape affect the consequent procedures to find feature

coefficients. The length of the window determines the frequency resolution of the signal. Increasing resolution is equal to using longer window, but in this case we may violate the assumption of stationary for signals. Typically 10 – 25 ms windows are used in speech signal processing [11]. We use 25 ms Hamming window because side lobes of this window is lower than the other windows which avoids leakage, (i.e. does not swap relevant energy from distant frequencies) as discussed in chapter three.

In the beginning of framing, the non-uniform spaced time instants values of the speech signal is recorded with the non-uniform samples value. This helps to find the samples within the 25 ms time interval of each frame by comparing the time instant of the sample values with the time interval of each frame allocated to have. If the time instants value is not found in that time interval, the time instant of the sample value will be compared with another frame time interval and the samples value will be stored in the frame at which the time instant of the sample is found. Figure 4.6 shows the overall step in a flow chart. The Matlab implementation of the framing algorithm can be found in Appendix A.2.

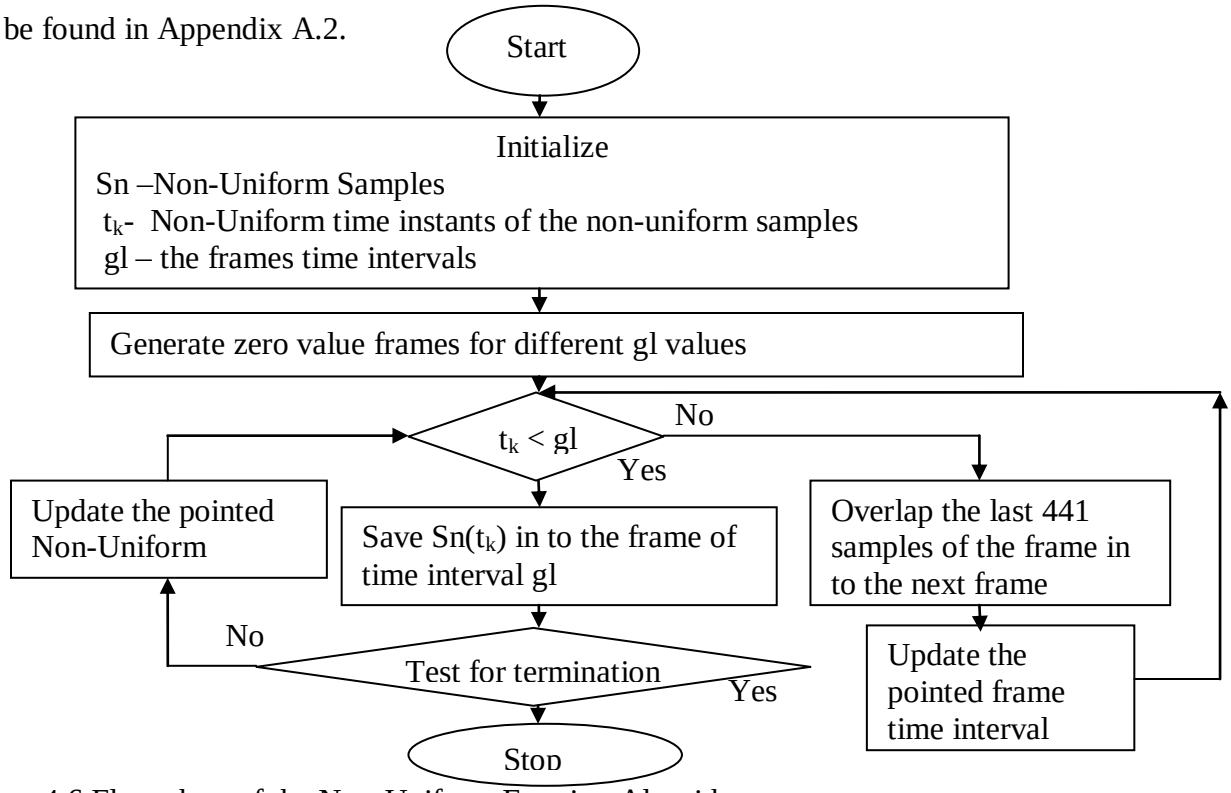


Figure 4.6 Flow chart of the Non-Uniform Framing Algorithm

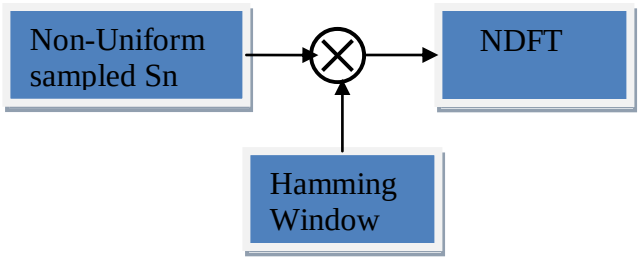


Figure 4.7 Block Diagram of the windowing process of the non-uniform sample values

As discussed in chapter two, accurate fast algorithm for NDFT does not exist. As a result we choose the direct algorithm of NDFT [4, 22] for processing the non-uniform samples in order not to lose spectral information which comes by the approximation used in the fast algorithms.

NDFT is used to convert speech frame to its frequency domain representation. The short-term power spectrum is found. Here phase information is discarded because it does not carry useful information from the hearing perspective.

4.1.3 Filter Bank Processing

Similar to chapter three here also, triangular filters are used to emphasize some of the frequency contents in power spectrum of the peech like ear does which is also called perceptual weighting. More filter banks process the spectrum below 1 kHz since the speech signal contains most of its useful information such as first formant in lower frequencies [9]. And also mel-scaling is used here in order to mimic the non-uniform spectral sensitivity of human ear.

In order to make the comparison between MFCC and NU-MFCC under normal circumstance, we choose 29 number of filter in the filter banks which is the number of filter used for sampling rate of 16 kHz under uniform sampling case.

4.1.4 Computation of Cepstral Coefficients

The logarithm of the coefficients is found after passing the spectrum through the filter banks. Human ear also smoothes the spectrum and use logarithmic scale. Logarithmic processing is useful in emphasizing the formants for noisy speech [11].

Since the frequency component of the sample uniformly spaced, we used IDFT on the logarithm of the filter bank output. As discussed in chapter three here also, we used DCT instead of direct IDFT which leads to using diagonal covariance matrices instead of full covariance matrices while modeling the feature coefficients by linear combinations of Gaussian functions [9, 11]. Therefore complexity and computational cost can be reduced. This is especially useful for large vocabulary speech recognition systems. Since DCT gathers most of the information in the signal to its lower order coefficients, by discarding the higher order coefficients significant reduction in computational cost can be achieved. Typically the number of static coefficients for recognition ranges between 8 and 13 [11,13, 44]. Since increasing the number of coefficient able to smoothen the spectra, we choose 13 numbers of static coefficients.

In order to combine dynamic properties of speech, first and second order differences of these coefficients may be used which are called delta and delta-delta coefficients.
We choose 13 delta coefficients in order to combine the speech dynamic properties since it is found enough to improve the recognition performance [46, 47]. Mathematically the above steps can be express as below.

Taking a discrete oversampled speech signal S[n]=*e*[n] $\otimes$ *θ*[n] which is the convolution of excitation signal with vocal tract response and non-uniformly sampling the discrete oversampled speech signal *S*[*n*] leads to non-uniform samples *Sn*[t_k].

NDFT of the *Sn*[t_k] gives:-

$$S_p(l) = \sum_{k=0}^{N-1} Sn[t_k]\, e^{-j2\pi x_l t_k} \;, l = 0 \ldots . N-1 \tag{4.5}$$

where x_l *is* $frequency\ values, t_k$*is time values ,N is number of samples*
Taking log compression Eq. (4.5) changes to Eq. (4.7)

$$\hat{S}_p(l) = \log\{S_P(l)\} \tag{4.6}$$

$$\hat{S}_p(l) = \log\left|S_p(l)\right| + jarg\left|S_p(l)\right| . \tag{4.7}$$

Then the Cepstral(*C*) can be computed as [56]

$$C_p[n] = \frac{1}{N}\sum_{k=0}^{N-1} \log\ |S_p[k]|\, e^{j\frac{2\pi}{N}kn}, n = 0,1, \ldots \ldots, N-1\,. \qquad (4.8)$$

Finally using NU-MFCCs method we computed the feature vectors for speech 'zero' as show in Figure 4.8 and the Cepstrogram as shown in Figure 4.9.

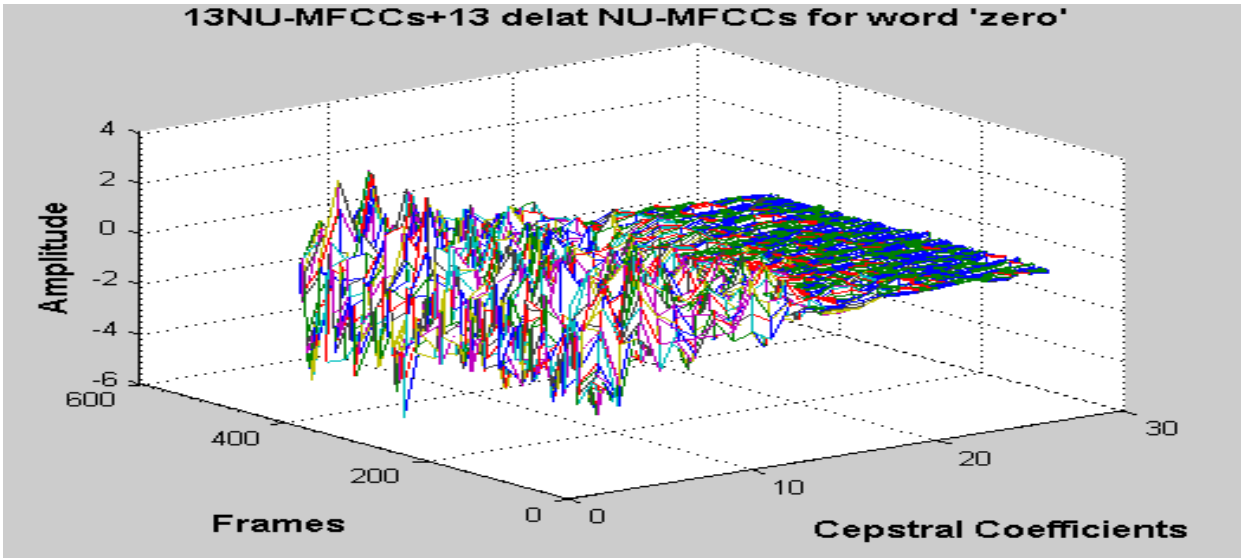


Figure 4.8 NU-MFCCs based Feature Vectors for speech 'zero'

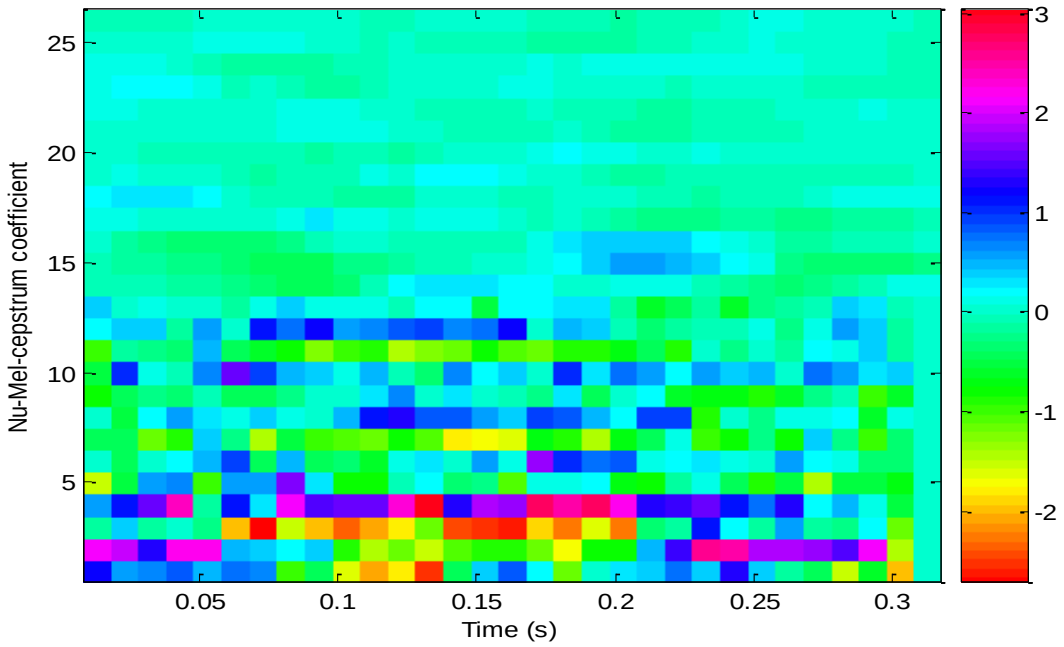


Figure 4.9 Cepstrogram of speech 'zero' based on NU-MFCCs.

4.2 Vector Quantization (VQ)

Ideally, storing as much data obtained from feature extraction techniques is advised to ensure a high degree of accuracy, but realistically this cannot be achieved. The number of feature vectors would be so large that storing and accessing this information using current technology would be unfeasible and impractical.

In this thesis, we use the idea of vector quantization results as part of Gaussian mixture models (GMM). The fundamental problem with the GMM in spite of having very high recognition rates is its computational complexity, efficiency and its adaptability for the low cost systems. The GMM uses the expectation maximization (EM) algorithm to train the speech and speaker models which actually contributes to the complexity of system. S. Memon et al. suggest that if we use the vector quantization techniques instead of EM algorithm then it not only reduces its computational complexity but also adds to its adaptability for low-cost systems [48].

Vector Quantization (VQ) is a quantization technique used to compress the information and manipulate the data such in a way to maintain the most prominent characteristics. It is a kind of pattern recognition that performed by clustering algorithm by unsupervised fashion in which information about the class membership of the training vectors is not provided [9]. In this technique the speech features vectors are clustered around some centroid locations. The resulting cluster centers constitute a codebook. Then, the codebook can be used to index a new feature vector by finding the cluster center (centroid) that is closest in some norm to the new vector [10].

VQ is used in many applications such as in data compression (i.e. image and voice compression). For this particular application, the regions correspond to specific spoken sounds, and the distortion caused by the quantization is a measure of how well the utterance matches the speech. Suppose that $x=[x^{(1)}\, x^{(2)}\, x^{(3)}\, x^{(L)}]$ is an L-dimensional column vector whose components $\{x^{(i)}, 1 \leq i \leq L\}$ are real-valued, continuous-amplitude random variables. In vector quantization, the vector x is mapped to another real-valued, discrete-amplitude, L-dimensional column vector, y. Thus, the quantization function is of the form

$$y = Q(x) \tag{4.9}$$

Typically, y takes on one of K distinct values such that $Y = \{y_j^i, 1 \le j \le K\}$,where y_j represents a vector of the form $y_j=[y_j^1, y_j^2, .., y_j^L]^T$. The set Y is referred to as the *codebook*, and each y_j is a *code vector*. To generate such a codebook, the L-dimensional space S_L of the speech feature vector x is partitioned into K mutually exclusive regions or clusters such that $\{C_j, 1 \le j \le K\}$ and a code vector y_j is associated with each cluster C_j . The vector quantizer then assigns the code vector y_j to the feature vector x_i if the feature vector x_i is in the cluster C_j , i.e.,

$$Q(x) = y_j, if\ x \epsilon\ C_j \tag{4.10}$$

For example each codebook is generated as follows: Given a set of training feature vectors (a 4-point feature vector for each frames of the utterance) which characterize the speech, find a partitioning of the feature vector space.

Each region contains a cluster of vectors, which represent the same basic sound. This region is represented by the *centroid vector*, which is the vector, which causes the minimum distortion when vectors in the region are mapped to it as shown in Figure 4.10 [49].

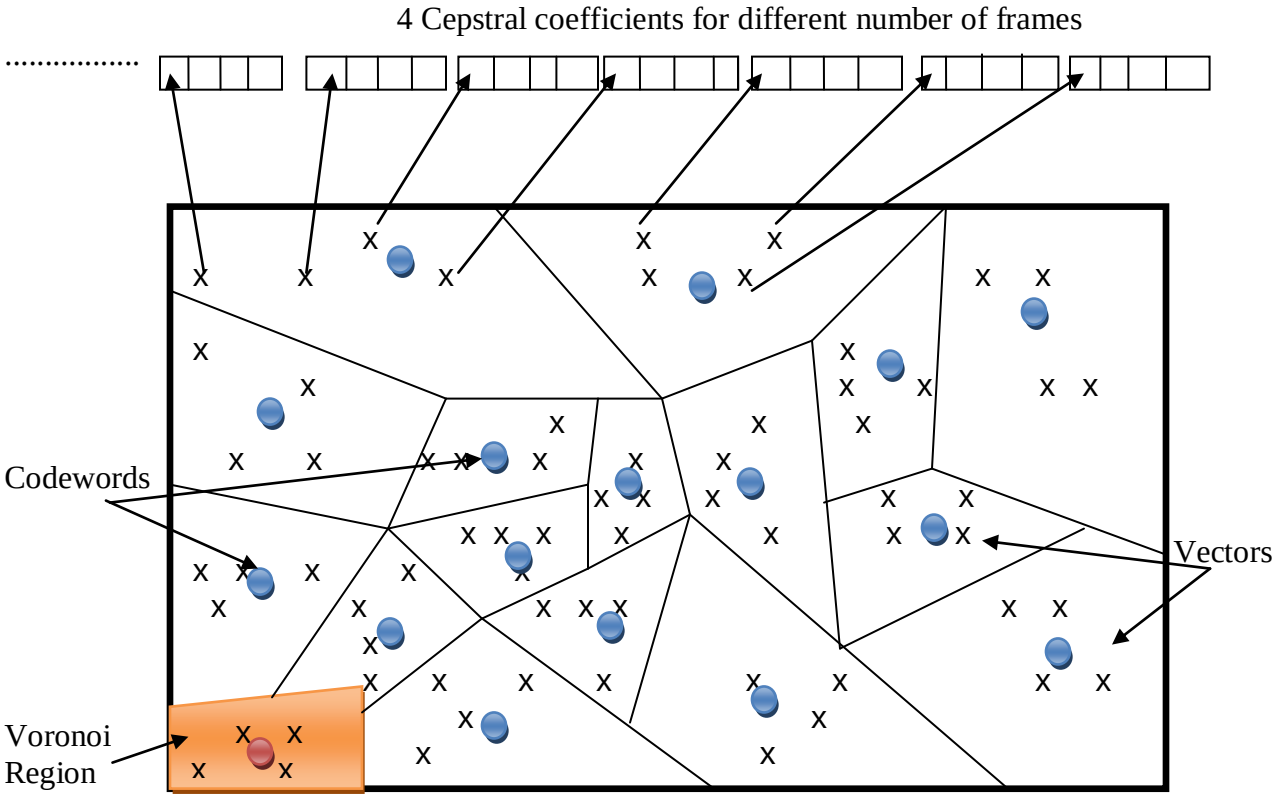


Figure 4.10 Two dimensional Voronoi diagram

When x is quantized as y_j, a quantization error results and a distortion measure (or a distance measure) $d(x, y_j)$ can be defined between feature vector x and the code vector (or codeword) y_j.

The squared Euclidian distance is one of the most popular distortion measures in speech recognition applications. The quantized code vector is selected to be the closest in Euclidean distance from the input speech feature vector [10]. The *Euclidean distance* is defined by:

$$d(x, y_j) = \sqrt{\sum_{i=1}^{L} \left(x^{(i)} - y_j{}^{(i)}\right)^2}, \tag{4.11}$$

where $x(i)$ is the i^{th} component of the input speech feature vector, and y_j^i is the i^{th} component of the codeword y_j.

There are two basic classes of clustering algorithms [9]. In *dynamic clustering*, a fixed number of clusters (classes) is used. At each iteration, feature vectors are reassigned according to certain rules until a stable partitioning of the vectors is achieved. In *hierarchical clustering*, each feature vector is initially a separate cluster, and then at each step of the algorithm, the two most similar clusters (according to some similarity criteria) are merged until the desired number of clusters is achieved. There are a variety of clustering algorithms, but we focus on only one example of an iterative approach which is widely used in speech processing for a number of tasks. The *Linde-Buzo-Gray* (LBG) *algorithm* was chosen to execute the vector quantization step for this thesis. The LBG algorithm is a finite sequence of steps in which, at every step, a new quantizer, with a total distortion less or equal to the previous one, is produced. The LBG algorithm and a slight variation are detailed in Algorithm2.

In this thesis using LBG algorithm, we clustered the feature vectors into 128 codebook size and also we calculated the mean and variance of each cluster which is used as initial mean and initial variance in Gaussian Mixture Modeling (GMM).

Algorithm 2 : The generalized Lloyd or Linde-Buzo-Gray (LBG) algorithm[28].

Initialization: Choose an arbitrary set of K code vectors, say $\bar{X}_k$,k = 1,2, ... ,K.

Recursion:

1. For each feature vector, x, in the training set, "quantize" x into code $\bar{X}_{k^*}$, where

$$k^* = \underset{k}{\operatorname{argmin}}\, d(x, \bar{x}_k) \tag{4.12}$$

Here *d(·, .)* represents some distortion measure in the feature space.

2. Compute the total distortion that has occurred as a result of this quantization,

$$D = \sum d[x, Q(x)], \tag{4.13}$$

where the sum is taken over all vectors *x* in the training set, and Q(x) indicates the code to which x is assigned in the current iteration. (This is an estimate of ε {d[x, Q(x)]}.) If *D* is sufficiently small STOP.

3. For each *k*, compute the centroid of all vectors *x* such that $\overline{X}_k$ = Q(x) during the present iteration. Let this new set of centroids comprise the new codebook, and return to Step 1.

Termination: Iterations in step 1 through 3 repeat until a specific termination criterion is met mostly reaching a maximum iteration number or having a sufficiently small squared Euclidean distance. Specifically, the goal is to terminate the iteration when the difference between the updated code vectors and the previous code vectors is sufficiently small.

4.3. Gaussian Mixture Models (GMM)

The probability density functions of many random processes, such as speech, are non- Gaussian. A non-Gaussian probability density function may be approximated by a weighted sum (i.e. a mixture) of a number of Gaussian densities of appropriate mean vectors and covariance matrices.

A mixture Gaussian density with M(in our case 128 codebook size) components is defined as [ ]

$$f_x(x) = \sum_{i=1}^{M} w_i \, N_i(x, \mu_i, \sum_i) \tag{4.14}$$

where $N_i(x, \mu_i, \sum_i)$ is a multivariate Gaussian density with mean vector μ_i and covariance matrix $\sum_i$ and w_i are the mixing coefficients. Consider *x* a D-dimensional discrete-valued data vector (i.e. measurement or features), each component density is a D-variant Gaussian function of the form $N_i(x, \mu_i, \sum_i)$ is given by[62]

$$N_i(x, \mu_i, \sum_i) = p(x/i) = \frac{1}{(2\pi)^{D/2} \, |\sum_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)' \sum_i^{-1} (x-\mu_i)\right\} \tag{4.15}$$

$$in\ our\ case \quad \mu = [26\ ,128]\ matrix\ , \qquad \mu_i = \begin{bmatrix} \mu m1 \\ . \\ \mu m26 \end{bmatrix} \tag{4.16}$$

$$\sum = [26,26,128]\ matrix, \quad \sum_i = \begin{bmatrix} m1_1 & \cdots & m26_1 \\ \vdots & \ddots & \vdots \\ m26_{26} & \cdots & m26_{26} \end{bmatrix} \tag{4.17}$$

The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = \{w_i, \mu_i\ , \sum_i\} \qquad i = 1, \ldots\ldots, M. \tag{4.18}$$

There are several variants on the GMM shown in Eq. (3.17). The covariance matrix , $\sum_i$, can be full rank or constrained to be diagonal. Additionally, parameters can be shared, or tied, among the Gaussian components, such as having a common covariance matrix for all components, The choice of model configuration (number of components, full or diagonal covariance matrices, and parameter tying) is often determined by the amount of data available for estimating the GMM parameters and how the GMM is used in a particular application [51].

It is also important to note that because the component Gaussians are acting together to model the overall feature densities, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modeling the correlations between feature vector elements. The effect of using a set of M full covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians. Because all the features used to train the GMM are unlabeled, the acoustic classes are hidden in that the class of an observation is unknown. A GMM can also be viewed as a single-state Hidden Markov Model (HMM) with a Gaussian mixture observation density, or an ergodic Gaussian observation HMM with fixed, equal transition probabilities. Assuming independent feature vectors, the observation density of feature vectors drawn from these hidden acoustic classes is a Gaussian mixture.

Given training vectors and a GMM configuration, we can estimate the parameters of the GMM, λ, which in some sense best matches the distribution of the training feature vectors. There are several techniques available for estimating the parameters of a GMM [52-53]. By far the most popular and well-established method is maximum likelihood (ML) estimation.

The aim of ML estimation is to find the model parameters which maximize the likelihood of the GMM given the training data. For a sequence of T training vectors $X = \{x_1,.., x_T\}$, the GMM likelihood, assuming independence between the vectors1, can be written as,

$$p(X \mid \lambda) = \prod_{t=1}^{T} p(x_t \mid \lambda) \tag{4.19}$$

Unfortunately, this expression is a non-linear function of the parameters λ and direct maximization is not possible. However, ML parameter estimates can be obtained iteratively using a special case of the expectation-maximization (EM) algorithm or other methods[48,53].

In general, there are an infinite number of different M-mixture Gaussian densities that can be used to 'tile up' a signal space. Hence the modeling of a signal space with a M-mixture probability density function space can be regarded as a many-to-one mapping, and the EM or the other methods can be applied for the estimation of the parameters of the Gaussian probability density function models[48,51].

Given some input feature vector *x* and a particular mixture *i* that has covariance matrix $\sum_i$ and mean μ_i, It can be able to tell how much a particular data point *x* is represented by a particular mixture[50]. This value is given by

$$p(i/x) = \frac{p(x/i)}{\sum_{j=1}^{M} p(x/j)p(j)} \tag{4.20}$$

where the values of *p(j)* are all set to 1/M, where M is the number of mixtures. In words this means that each mixture has equal probability, i.e. there is no biasing.

The basic idea of the EM algorithm is, beginning with an initial model λ, to estimate a new model $\bar{\lambda}$, such that $p(X|\lambda) \geq p(X|\bar{\lambda})$. The new model then becomes the initial model for the next

iteration and the process is repeated until some convergence threshold is reached. The initial model is typically derived by using some form of binary VQ estimation. On each EM iteration, the following re-estimation formulas are used which guarantee a monotonic increase in the model's likelihood value [51],

Mixture Weights

$$\hat{p}(i) = \frac{1}{T}\sum_{t=1}^{T} p(i|X_t\ ) \tag{4.21}$$

Mean

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \frac{\sum_{t=1}^{T} p\ (i|X_t)x_t}{\sum_{t=1}^{T} p(i|X_t)} \tag{4.22}$$

Variances (diagonal variance)

$$\hat{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p\ (i|X_t)x_t^2}{\sum_{t=1}^{T} p\ (i|X_t\ )} - \hat{\mu}_i^2 \tag{4.23}$$

where $\sigma_i^{\ 2}$, x_t, and μ_i refer to arbitrary elements of the vectors $\boldsymbol{\sigma}_{\mathbf{i}}^{\ \mathbf{2}}$, $\mathbf{X_t}$, and $\boldsymbol{\mu}_{\mathbf{i}}$, respectively.

In general the means and covariances are estimated from the vectors in each cluster. After the estimation, the feature vectors can be reclustered using component densities (likelihoods) from the estimated mixture model and then model parameters are recalculated. This process is iterated until model parameters converge.

CHAPTER FIVE: SYSTEM IMPLEMENTATION AND RESULTS ANALYSIS

Using of the word error rate (WER) as a common measure has served to advance speech recognition research in recent times [54]. The word error rate, commonly used to evaluate ASR systems, is derived from the Levenshtein distance, or edit distance. The edit distance between two strings is the minimum number (or weighted sum) of insertions, deletions and substitutions required to transform one string into the other [55].

The WER is the edit distance between a reference word sequence and its automatic transcription, normalized by the length of the reference word sequence. This normalization is applied to allow comparison between different systems on different tasks, as the magnitude of the edit distance depends on the string length.

Defining N_r as the total words in the reference transcription, Na as the total words in the automatic transcription, S as the number of substituted words in the automatic transcription, D as the number of words from the reference deleted in the automatic transcription, I as the number of words inserted in the automatic transcription not appearing in the reference, and H as the number of correctly recognized words. The word error rate is defined as [55]:-

$$WER = \frac{S + D + I}{N_r}. \tag{5.1}$$

While this measure is most commonly used as an error rate, it is also often quoted as the word recognition rate,

$$WRR = 1 - WER = \frac{H - I}{N_r}, \tag{5.2}$$

Noting that $H = Nr - (S + D)$. Sometimes, particularly for isolated word recognition systems, the word correct rate is used as a performance measure for speech recognition [54]. It doesn't consider insertion errors, and is defined as

$$WCR = \frac{H}{N_r} \tag{5.3}$$

As a result we choose *WCR* as a performance measure for the build speech recognition system.

5.1- Overview of the System Implementation

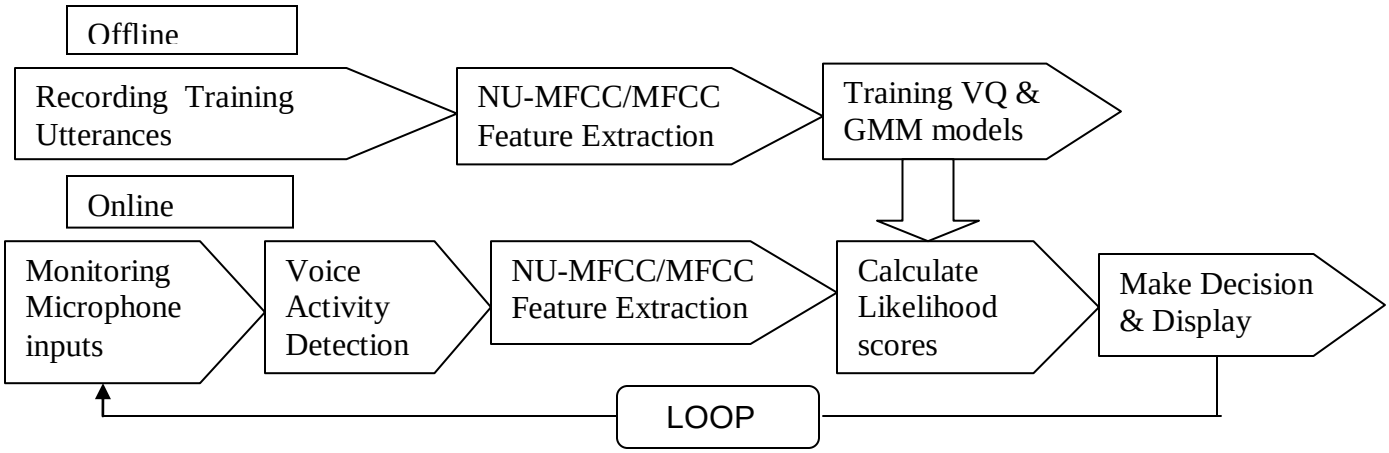
The main goal of the study was to recognize a set of eleven spoken words, in which the acoustic signals were collected under normal external surrounding and noisy condition using non-uniform sampling based isolated word recognizer. The small vocabulary consists of eleven words: {Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten} and the speech database was generated using 10 adult subjects (2 female, 8 male) who uttered each word 4 times for sampling rate of 16KHz and 4 time for sampling rate of 44.1 kHz under normal external surrounding for the training purpose and also uttered 5 time for each sampling rate for testing purpose.
A GMM recognizer was chosen because the vocabulary under consideration is small and consists of short words. And also it need little computational time and is simple to implement.
Both MFCCs and NU-MFCCs based Isolated Word Automatic Speech Recognition has been built in order to allow the comparison to be performed under normal circumstance.

Two important software programs were used during the development of the recognition system. For the data collection stage which mainly consists of voice recording and voice signal editing, Sound Forge (version 10.0) audio editing software of Sonic Foundry was used. This program has extensive audio editing and spectrum analysis capabilities which made the experimental training stage easier. The second program was MATLAB (version 7.0) of Math Works, where the system was actually developed and tested. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Along with MATLAB, two MATLAB toolboxes, provided under GNU General Public License, were used.

1. VOICEBOX is a toolbox that includes many useful functions for speech processing. Specifically for MFCC calculations, MELCEPST function of this library was used [42].
2. H2M is MATLAB code library which contains several functions for implementing GMM and VQ [56].

The built Isolated Word Speech Recognition Systems includes the following five phases:-



Figures 5.1: Isolated Word Speech Recognition System build for both MFCC and NU-MFCC based feature extraction method

1-Monitoring Microphone Input (Recording step*)*: The audio recording takes place at normal external surrounding (office environment). An isolated utterance of 880 were collected from 10 adult subject who uttered each word 4 times for training purpose from the eleventh word at 16 kHz and 44.1 kHz sampling rate separately. In addition, 1100 isolated utterances were collected from the same 10 adult subjects who uttered each word 5 times within different sampling rate(16kHz and 44.1kHz), which were recorded for testing purpose under normal external surrounding. A Toshiba L-300 laptop is used in which the ADC digitized the speech recordings with both sampling rate at 16-bit quantization.

2-Front end processing step (Voice Activity Detection): An accurate extraction of the speech utterances from the audio was difficult so a careful implementation was done. First, the mean of the digital signal was normalized to zero, so as to eliminate any bias introduced by the microphone. Secondly, a search scheme based on short-time energy and zero-crossing rate was used to identify beginning and end of each utterance, as discussed earlier in Chapter three.

3-Feature Extraction:-This step is performed in order to obtain the spectral and temporal characteristics of the speech waveform in a compact representation. MFCCs based feature extraction is used for comparison purpose to the proposed NU-MFCCs based feature extraction.

In the MFCCs case , a frame size of 400 samples(corresponding to 25 ms) with an overlap of 160 samples (corresponding to 40% overlapping) from frame to frame and a Hamming window applied to each frame and also a filter bank of 29 triangular filters was used. The first 13 MFCC (static parameters) and 13 delta-MFCC (dynamic parameters) parameters were extracted from

each frame of the speech signal, and the mean of the static MFCC parameters was normalized to zero and the delta-MFCC parameters were scaled to the range of the static MFCC parameters by multiplying them by a factor of six.

And also similarly for NU-MFCCs case, since it is not know the number of samples found in 25 ms time interval of the frame, a maximum frame size of 1103 samples (corresponding to 25 ms at 44.1 kHz under uniform sampling rate) with an overlap of 441 samples (corresponding to 10ms overlapping time) from frame to frame and a Hamming window applied to each frame, but the other parameters used for extraction of the feature is similar to the MFCCs case.

4. Vector Quantization (VQ) and GMM step: - In this thesis, we apply a vector quantization technique such as LBG algorithm for the training of GMM models as a replacement of EM algorithm. Feature vectors were generated from the training set and used to generate a codebook of length equal to 128 and LBG algorithms used as vector quantization techniques for both MFCC and NU-MFCC cases, as discussed in Chapter three. GMM is used as parametric model in both cases because of its simplicity and need little computational time.

5. Recognition step: - Recognition involves mapping the given input in the form of features to one or the known sounds. This may involve use of various knowledge models for the precise identification and ambiguity removal. Endpoint detection, feature extraction were performed for each word in the testing set. Once model parameters are determined from the training set, the recognizer is designed to decide that unlabelled data is most likely to have come from a specific model $\boldsymbol{\lambda}_i$ by selecting the model which maximizes the probability that the data was generated by the given model, i.e.,

$$\lambda_s = \arg\max_i \left[p(\lambda_i|X) = \frac{p(\lambda_i)(p(X|\lambda_i)}{p(X)} \right] \quad for\ i = 1,2,\ldots.11 \qquad (5.4)$$

where λ_s is the identity of the word, X is the observation vector , $\lambda_i$is the i^{th} vocabulary word and $p(\lambda_i)$ is a given set of prior probabilities for a given set of probabilities. The most probable spoken word depends only on the likelihood $p\left(\frac{X}{\lambda_i}\right)$, $p(X)$ is the probability of the recorded signal so after recording it becomes same for all recordings, The problem is then to find the maximum value of the product of $p(\lambda_i) with\ p(\frac{X}{\lambda_i})$ and, i.e., the recognizer identifies the model, which is most-likely, given the observation sequence.

5.2- Experimental Results and Discussions

Several models were considered for MFCC based speech recognition by varying the frame length, the number of MFCC coefficients, and implementations with and without the delta-MFCC coefficients. The suited simulation parameters found for MFCC based speech recognition under normal external surrounding is shown in Table 5.1:-

Table 5.1: Simulation parameters used for MFCC based Speech Recognition.

Parameters	Values
Database	Isolated 11 words(Zero-Ten)
Speakers	2 Female, 8 Male
Sampling Frequency	16 KHz
Frame length	400
Increments	160
Speech features	13-MFCCs and 13-delta MFCCs
Mel-scale filter bank	29 triangular filter
Delta-MFCC scaling factor	6
Codebook Size	128

5.2.1 Normal External Surrounding Condition

The normal external surrounding condition can be understand as office environment or as when SNR $\geq$ 40 dB. The experiment conducted for different number of testing utterance for each word under normal external surrounding. In the case of MFCC based feature extraction taking 50 utterances for each word, the following recognition rate matrix is found.

Table 5.2: Recognition rate matrix for 50 testing utterance for each word under normal external surrounding and their WCR performance in percentage

words	zero	one	two	three	four	five	six	seven	eight	nine	ten	Acc(%)
zero	45	0	0	1	0	0	0	0	0	1	3	90
one	1	46	0	0	1	2	0	0	0	0	0	92
two	4	0	42	3	1	0	0	0	0	0	0	84
three	0	0	0	47	0	0	0	0	0	0	3	94
four	0	0	0	0	50	0	0	0	0	0	0	100
five	0	0	0	0	0	50	0	0	0	0	0	100

six	0	0	0	1	0	0	49	0	0	0	0	98
seven	0	0	0	0	0	0	0	50	0	0	0	100
eight	0	0	0	6	0	0	0	0	38	3	3	76
nine	0	0	0	0	0	0	0	0	0	46	4	92
ten	0	0	0	1	0	0	0	0	4	0	45	90
Average WCR performance of the speech Recognition system at 550 utterance : 92.36 %												

Table 5.3: Sound Lexicon of Digits [32]

Word	Sounds	ARPABET
Zero	/ z I r o/	Z-IH-R-OW
One	/w ʌ n/	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 ɛ v ə n/	S-EH-V-AX-N
Eight	/e^y t/	EY-T
Nine	/n a^y n/	N-AY-N
Ten	/t ə n/	T-AX-N

As shown in a Table 5.2 the word ‘three’ is confused with most words. We think that this is due to the similarity of many parts of digit 3 to noise such as phoneme / θ / which exist in the digit as shown in Table 5.3 [57]. End point detection also has some effect for miss recognition of one word with another. For example, unvoiced consonants such as stop sound‘t’ as in the word ‘eight’ have considerably less energy and are usually focused on higher frequency whereas the voiced vowels such as ‘u’ in the word ‘two’ have most of the energy located in the lower portion of the spectrum as a main difference. But the reasons why one word miss recognize with another word mostly need different research. We did not give much emphasize on this problems since our aim is to see how the recognizer perform based on NU-MFCCs method as compared to MFCCs method.

Table 5.4 Words that were picked in case of miss-recognition of the test words

Words	Mostly Confused With words
zero	three, nine ,ten
one	zero ,four ,five
two	zero, four ,three
three	ten
four	-
five	-
six	three
seven	-
eight	three , nine ,ten
nine	ten
ten	eight ,three

It is shown in Figure 5.1 the recognition performance of the speech recognition based on MFCC increases as the number of utterance increased. As a result increasing the utterances helps to know the accurate performance of the recognition system.

Taking different number of testing utterance for each word, the following WCR performance is found.

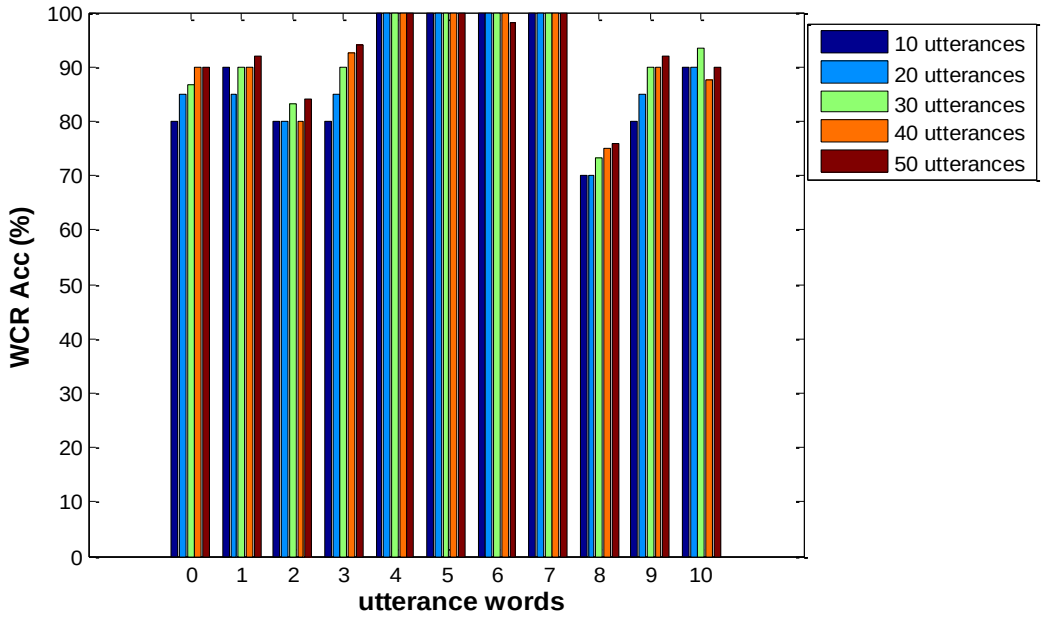


Figure 5.1 WCR accuracy of MFCC based feature extraction for different number of testing utterance for each word

Similarly for NU-MFCC based feature extraction, the experiment is conducted for different number of utterance for each word.

Table 5.5: Simulation parameters of NU-MFCCs based Speech Recognition

Parameters	**Values**
Database	Isolated 11 words(Zero- Ten)
Speakers	2 Female ,8 Male
Sampling Frequency	44.1 kHz
Frame Length	1103 samples
Increments	441 samples
Speech Features	13- NUMFCCs and 13- delta NUMFCCs
Mel-scale Filter Bank	29 Triangular Filter
Delta-NUMFCC Scaling Factor	6
Codebook Size	128

Taking 50 utterances for each word, the following confusion matrix is found

Table 5.6: Confusion Matrix for 50 testing utterance for each word under normal external surrounding and their WCR performance in percentage

words	zero	one	two	three	four	five	six	seven	eight	nine	ten	Acc(%)
zero	47	0	0	0	0	0	0	0	0	2	1	94
one	0	49	0	1	0	0	0	0	0	0	0	98
two	0	2	47	1	0	0	0	0	0	0	0	94
three	2	0	0	38	0	0	6	4	0	0	0	76
four	0	0	0	0	50	0	0	0	0	0	0	100
five	0	0	0	0	0	50	0	0	0	0	0	100
six	0	0	0	0	1	0	48	0	1	0	0	96
seven	0	0	0	0	0	1	0	49	0	0	0	98
eight	0	0	0	6	0	0	4	0	40	0	0	80
nine	0	0	0	0	0	0	0	1	0	49	0	98
ten	0	0	0	5	0	2	0	3	0	0	40	80
Average WCR performance of the speech Recognition system at 550 utterances: 92.18 %												

Table 5.7: Words that were picked in case of miss-recognition of the test words

Words	Mostly confused words with
zero	nine ,ten
one	three
two	one ,three

three	zero, six seven
four	-
five	-
six	four, eight
seven	five
eight	three , six ,ten
nine	seven
ten	three ,five ,seven

Taking different number of utterance for each word the following results has been found in the case of NU-MFCCs based speech recognition under normal external surrounding,

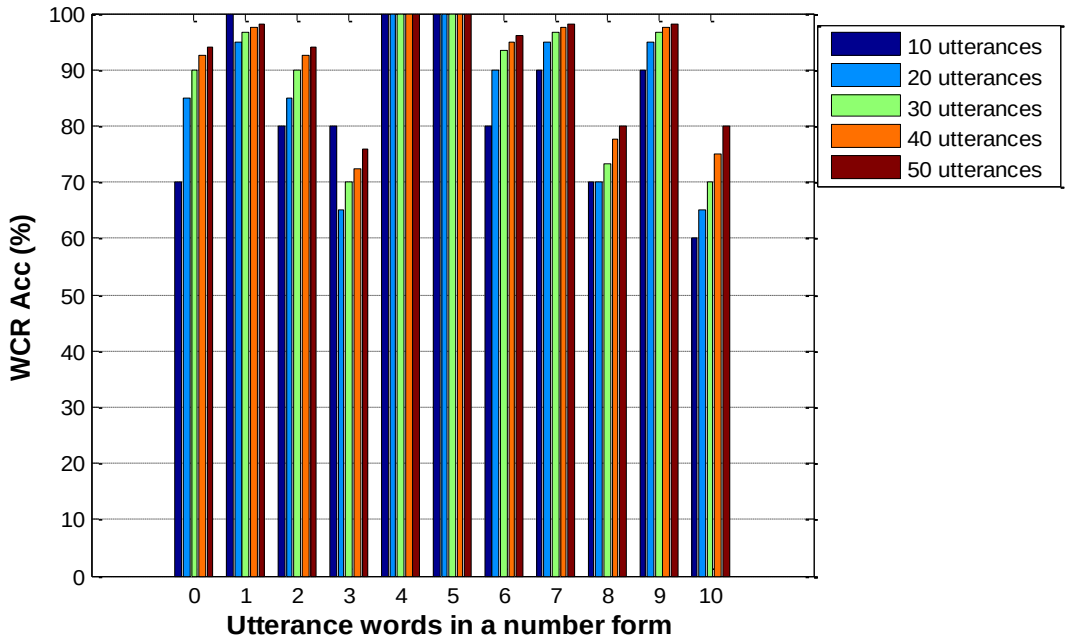


Figure 5.2 WCR accuracy of NU- MFCC based feature extraction for different number of testing utterance for each word under normal external surrounding

In general, comparing the performances of the MFCC and NU-MFCC based feature extraction for different number of utterances under normal external surrounding, the MFCC based feature extraction have better performance under a smaller number of utterance but when the number utterances increased the NU-MFCC based feature extraction have comparable result with the MFCC as shown in Figure 5.3.

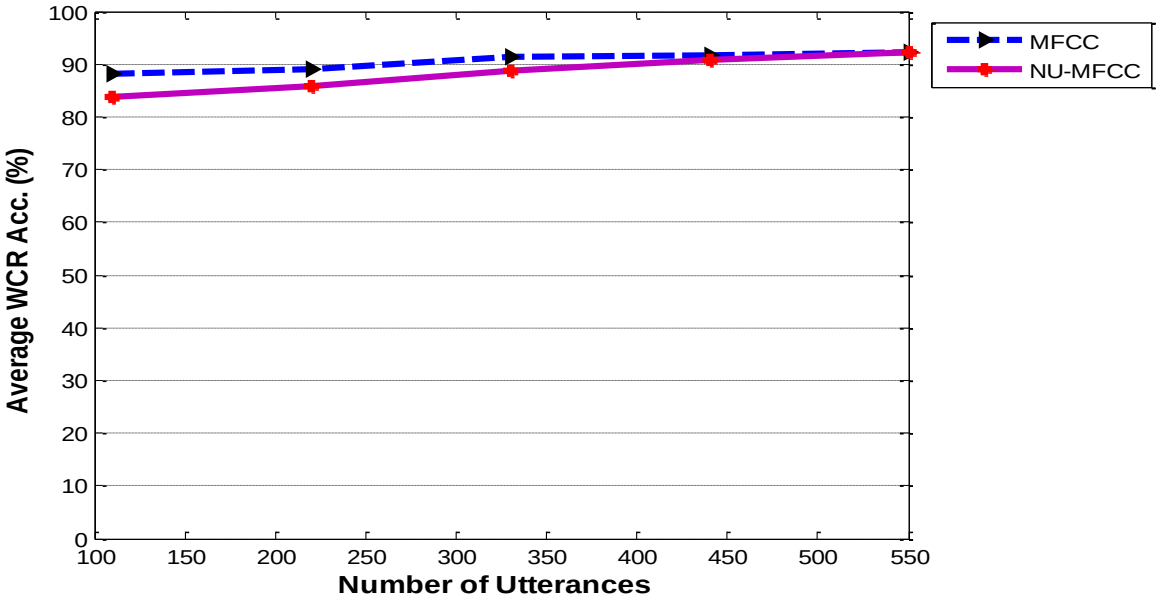


Figure 5.3 Average performance comparisons between MFCC and NU-MFCC based feature extraction for different number of utterances under normal external condition.

In order to consider the speech recognition system under noisy condition, we choose AWGN, Pink and Volvo noise in the experiment.

5.2.2 Additive White Gaussian Noise Condition

As shown in Figure 5.3 the performance of MFCC and NU-MFCC based feature extraction is more likely similar at 550 number of utterance. As a result we choose to performance the comparison between MFCC and NU-MFCC based feature extraction at 550 number of utterance in the AWGN condition. And also since the performances of the word 'Four' and 'Five' are 100 % as shown in Figure 5.1 and 5.2, we found these two words as a good candidate to compare the performances of the NU-MFCCs and MFCCs method under different conditions.

Comparison of the performance of MFCC and NU-MFCC based feature extraction is shown in Figure 5.4 taking 50 utterances for speech 'four' under AWGN condition .

As shown in a Figure 5.4, for SNR >35 dB the WCR accuracy of the word 'four' is equal to the condition under normal external surrounding and also we can see the performance of the NU-MFCC is much better than MFCC between 0-30 dB.

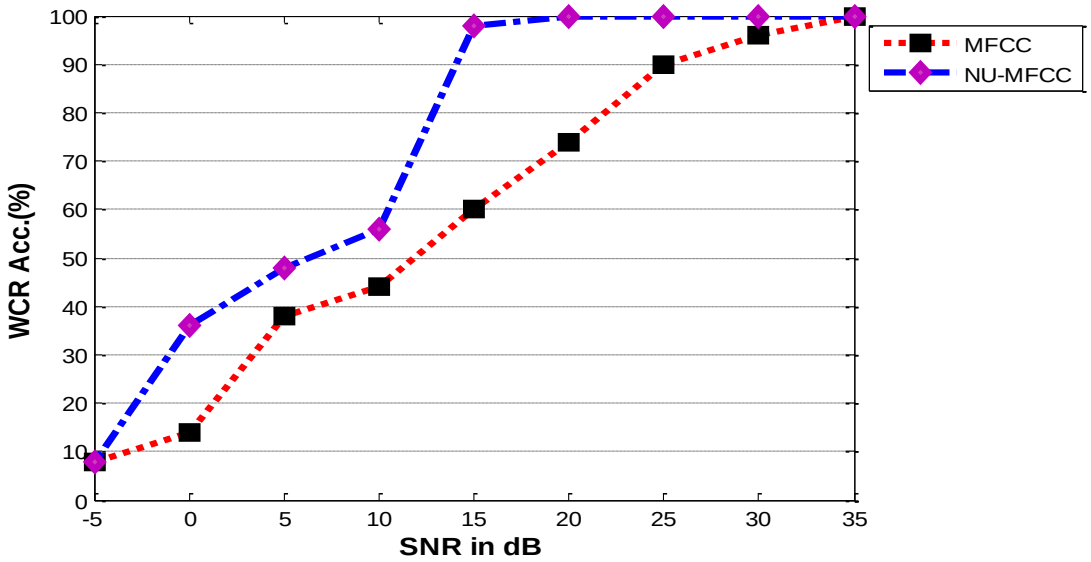


Figure 5.4 Performance Comparison between MFCCs and NU-MFCCs based feature extraction of word 'four' for a system trained with normal external surrounding speech at AWGN.

We can see in Figure 4 and 5, a big (30%-40%) performance increase is found between 10dB and 15dB AWGN under NU-MFCCs case. Similar to Figure 5.4, we can see also in Figure 5.5 the performance of NU-MFCC is much better than MFCC for speech 'five' at SNR between 0-25 dB for a system trained with normal external surrounding speech.

In the performance of the speech recognition, the speech training condition also has an effect. If the system is trained at clean environment and tested at noisy condition the performance of the speech recognition system will decrease as compared to the system trained and tested at noisy condition. In order to include this situation, we trained the system with a mixture of normal external surrounding speech and AWGN at 10 dB SNR in addition to the training done at normal external surrounding. 10 dB SNR AWGN is chosen because most of the AWGN are seen to affect speech recognition between -5 to 25 SNR in dB. So we found taking 10 dB SNR the right choice in order to consider the two extreme ends.

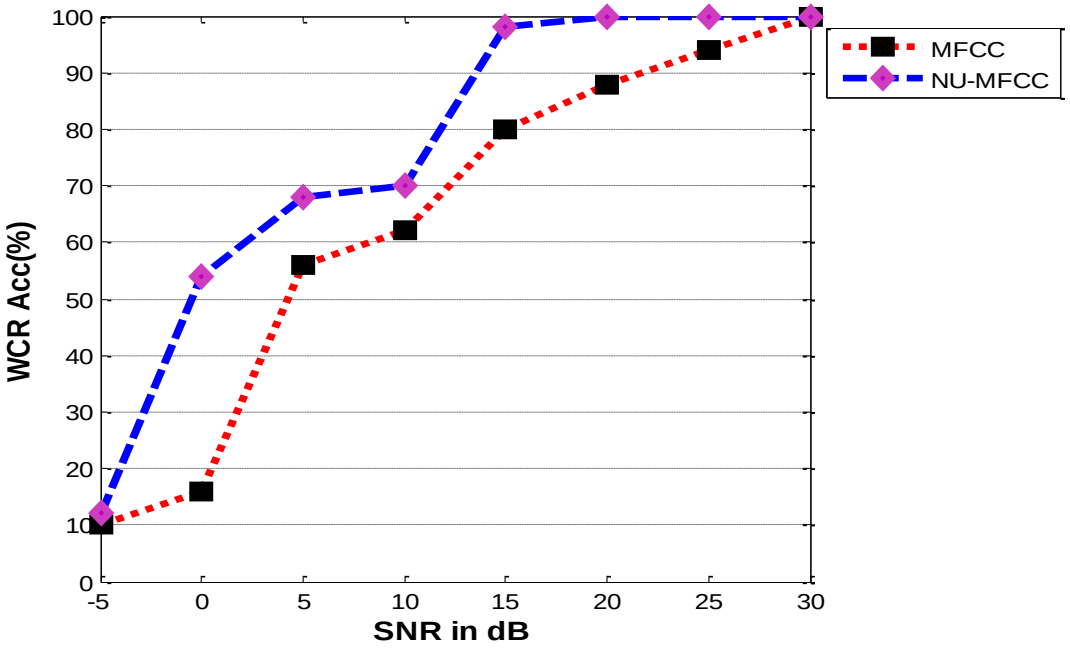


Figure 5.5 Performance Comparison between MFCCs and NU-MFCCs based feature extraction of speech 'five' for a system trained with normal external surrounding speech at AWGN.

We can see in Figure 5.6, the speech 'four' performance based on NU-MFCC is much better than MFCC for SNR between -5 and 20 dB for a system trained with a mixture of normal external surrounding speech and AWGN.

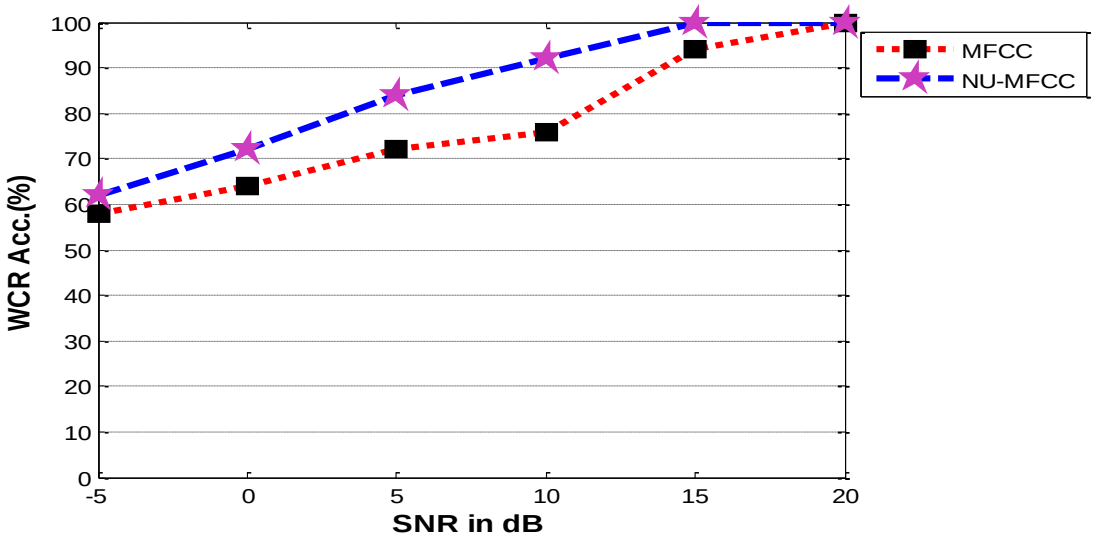


Figure 5.6 Performance comparison between MFCCs and NU-MFCCs methods for a system trained with mixture of normal external surrounding speech and AWGN for the word 'four'.

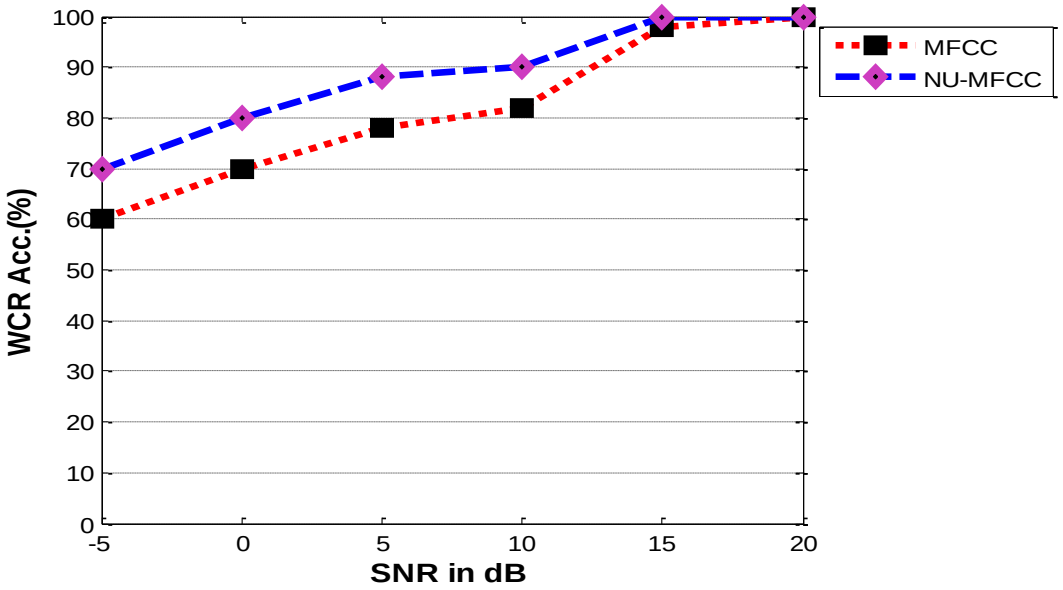


Figure 5.7 Performance Comparison between MFCCs and NU-MFCCs based feature extraction for a system trained with mixture of normal external surrounding and AWGN for the word 'five'.

From Figure 5.7, we can see that the performance of NU-MFCCs is much better than MFCCs at SNR between -5 and 15 dB for a system trained with a mixture of normal external surrounding speech and AWGN taking speech 'five'.

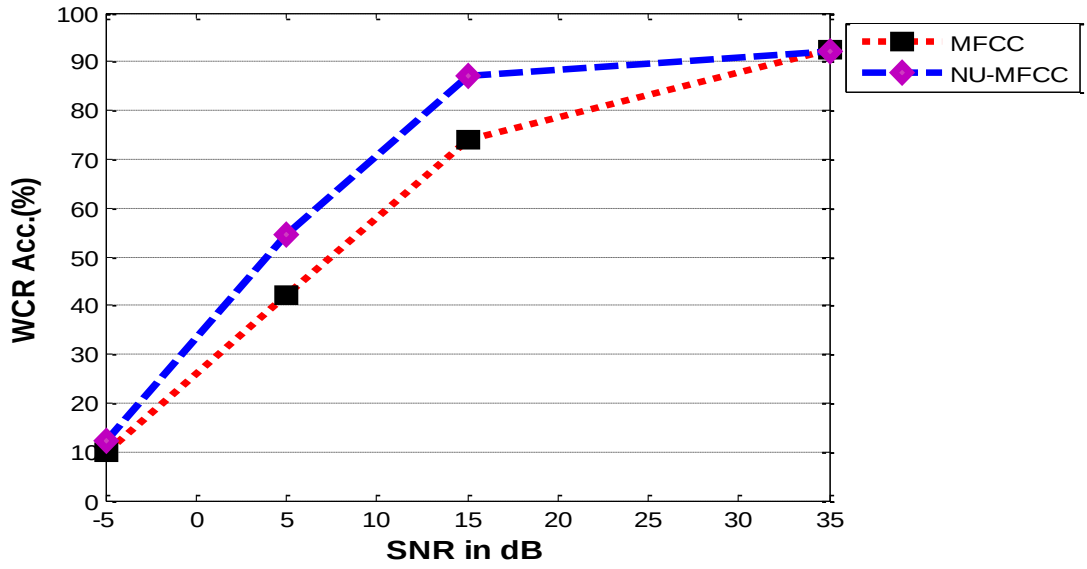


Figure 5.8 Average performance comparisons between MFCC and NU-MFCC based feature extraction for a system trained with a normal external surrounding speech at AWGN condition.

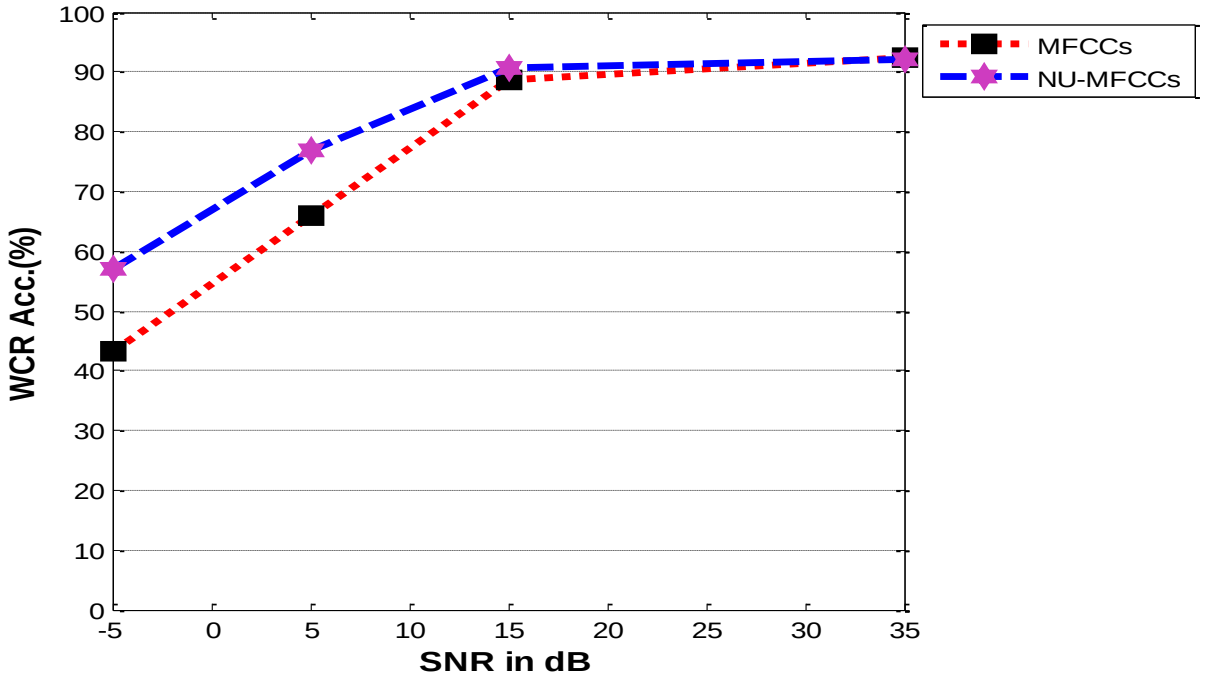


Figure 5.9 Average performance comparisons between MFCC and NU-MFCC based feature extraction for a system trained with mixture of normal external surrounding speech and AWGN.

Taking -5, 5 and 15 dB SNR and calculating the average performance of the eleven words with MFCC based speech recognition for a system trained at normal external surrounding speech in AWGN condition; we can find 42% accuracy. And also for NU-MFCC based speech recognition, we can find 51.27%.

Similarly for a system trained with a mixture of normal external surrounding speech and AWGN at 10 dB SNR, we can find for MFCC based speech recognition 65.87% accuracy and for NU-MFCC based speech recognition 74.84% accuracy for the eleven words used.

In general, unlike in a clean speech we can see in Figure 5.8 and Figure 5.9 the performance of NU-MFCC based speech recognition is much better than MFCC based speech recognition for a system trained with normal external surrounding speech and also for a system trained with mixture of normal external surrounding speech and AWGN.

5.2.3. Additive Pink Noise Condition

Considering pink noise as additive noise to the speech ‘four’ and ‘five’, the following results has been found.

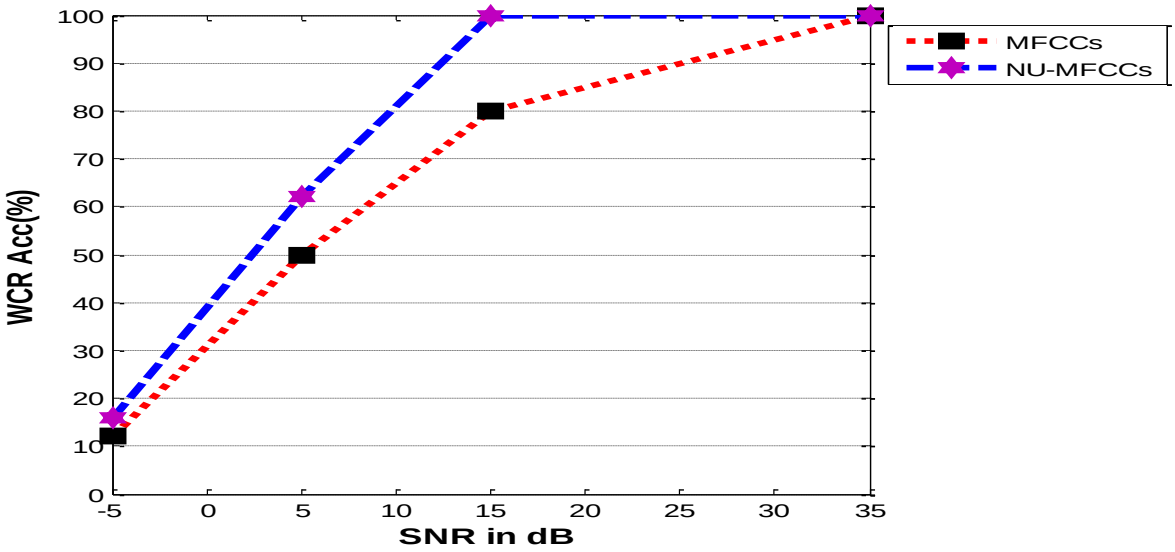


Figure 5.10 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with normal external surrounding speech for the word 'four' at additive pink noise condition.

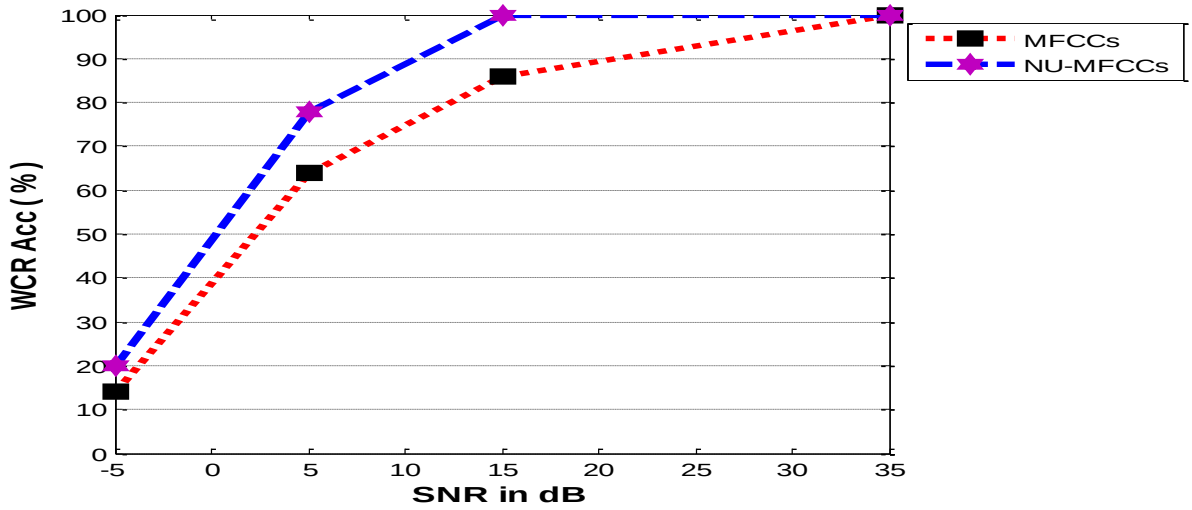


Figure 5.11 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with normal external surrounding speech for the word 'five' at additive pink noise condition.

It is shown in Figure 5.10 and Figure 5.11 that the WCR accuracy of the NU-MFCCs method is better than the MFCCs method for the word 'four' and 'five' under additive pink noise condition. But as compared to AWGN condition(below 10%) at -5 db SNR the performance of the speech recognition increased under additive pink noise condition(10-20%) for a system trained at normal external surrounding speech.

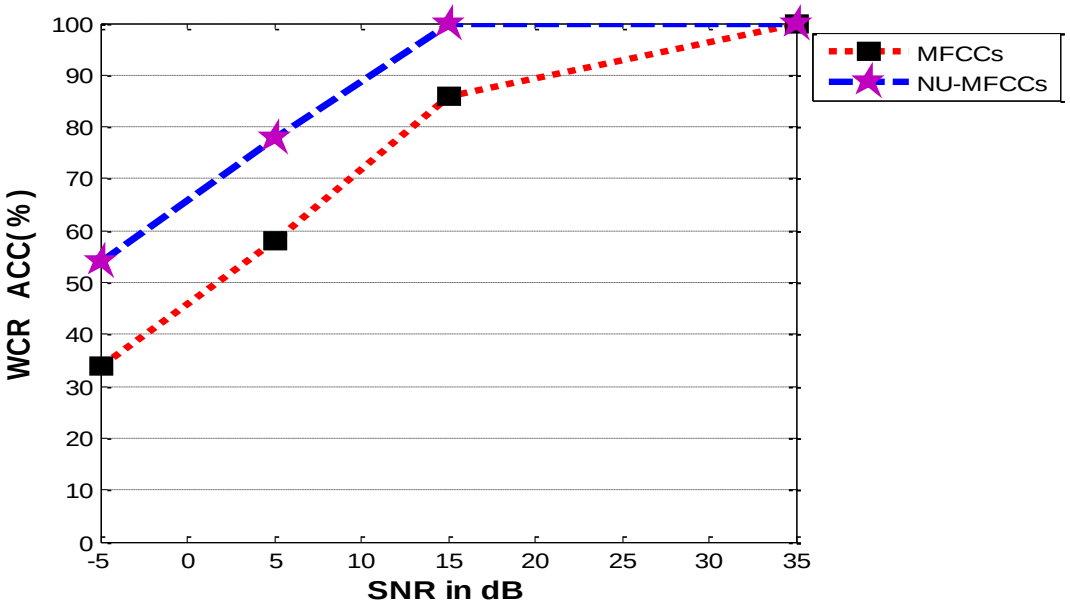


Figure 5.12 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with a mixture of normal external surrounding speech and AWGN for the word 'four' at additive pink noise.

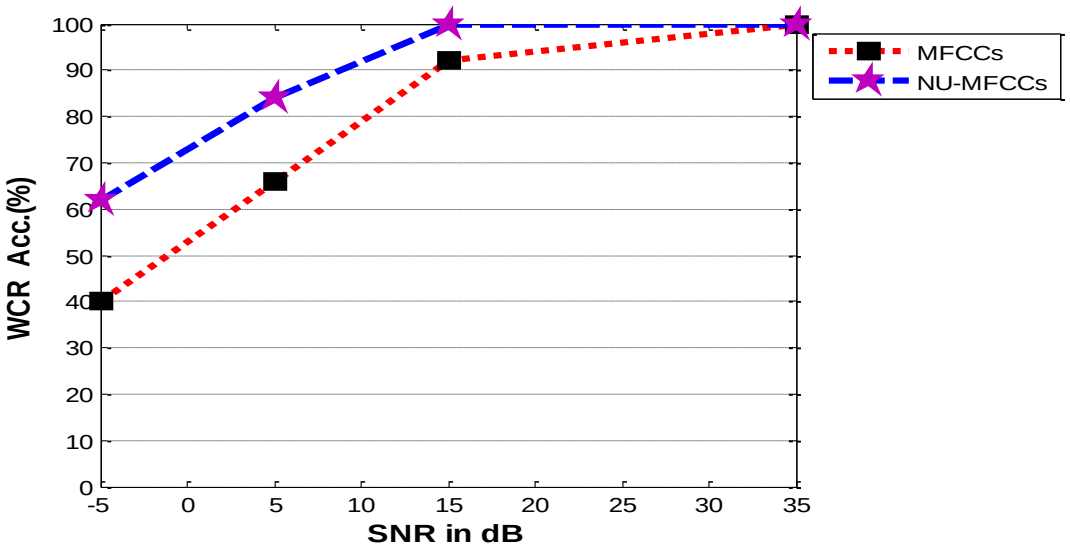


Figure 5.13 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with a mixture of normal external surrounding speech and AWGN for the word 'five' at additive pink noise condition.

Generally, similar to the above cases the performance of NU-MFCCs method perform better than MFCC method for a system trained with a mixture of normal external surrounding speech and

AWGN speech at additive pink noise condition as it is shown in Figure 5.12 and Figure 5.13 for the word 'four' and 'five'.

5.2.4. Additive Volvo Noise Condition

Considering Volvo noise as additive noise to the speech 'four' and 'five' similar to the above cases, the following results has been found.

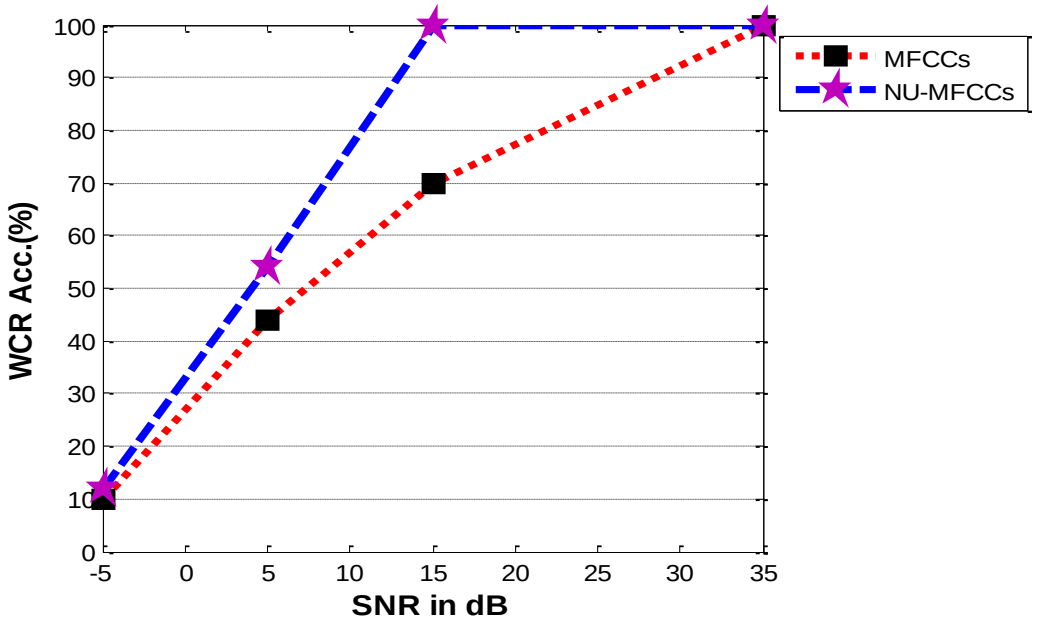


Figure 5.14 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with normal external surrounding speech for the word 'four' at additive Volvo noise case.

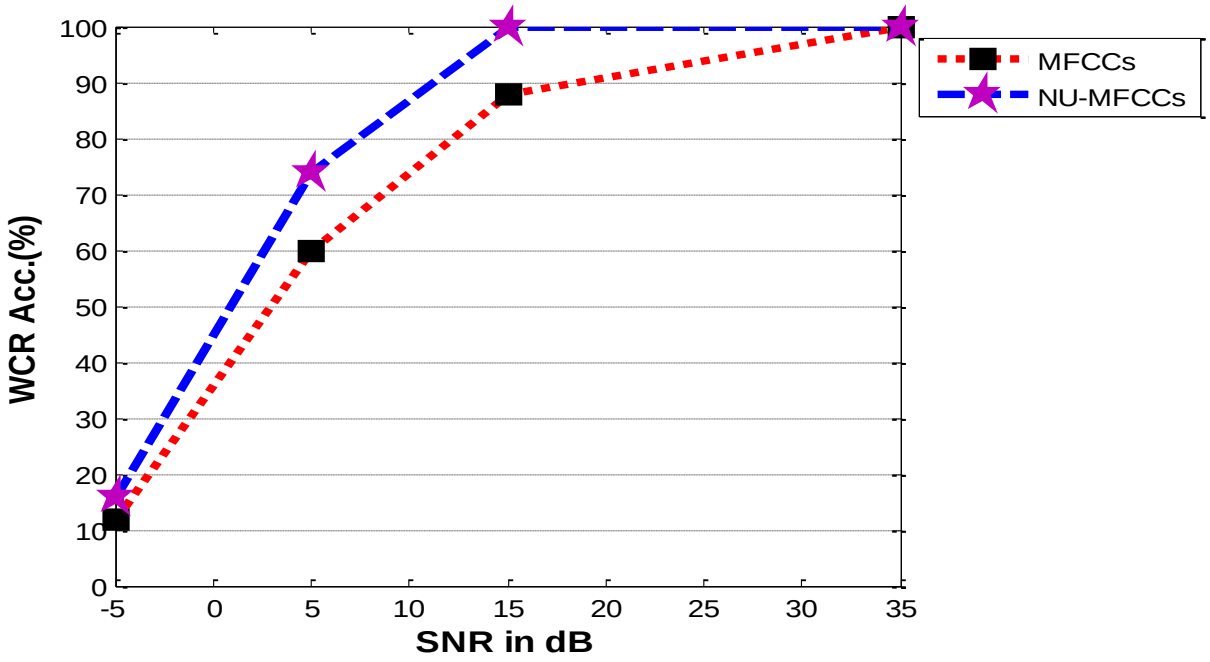


Figure 5.15 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with normal external surrounding speech for the word 'five' at additive Volvo noise case.

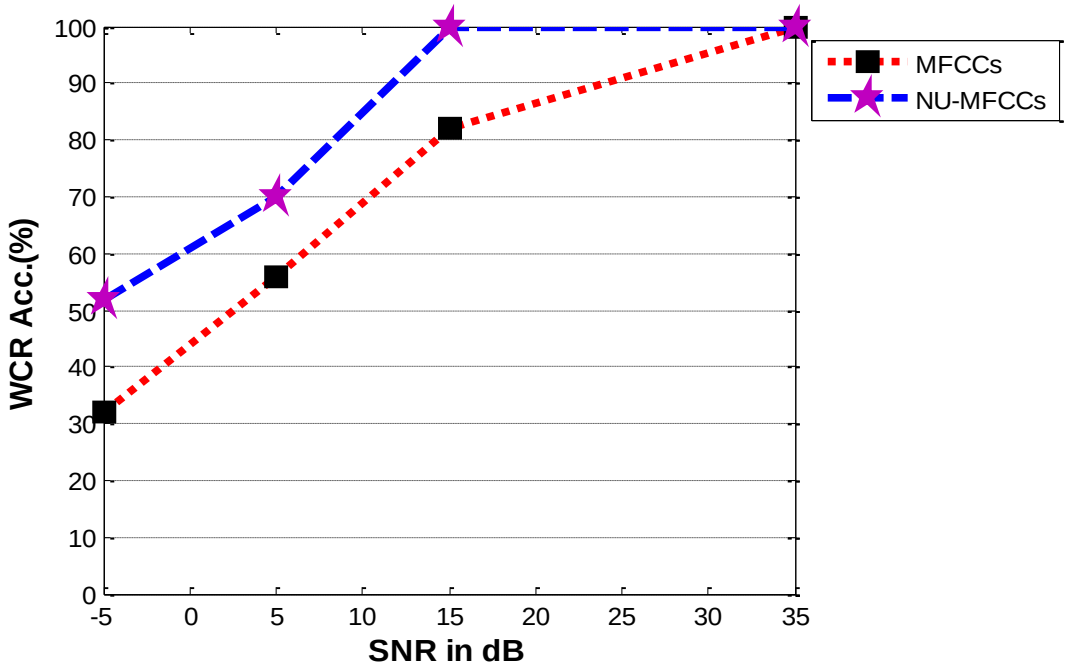


Figure 5.16 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with a mixture of normal external surrounding speech and AWGN for the word 'four' at additive Volvo noise case.

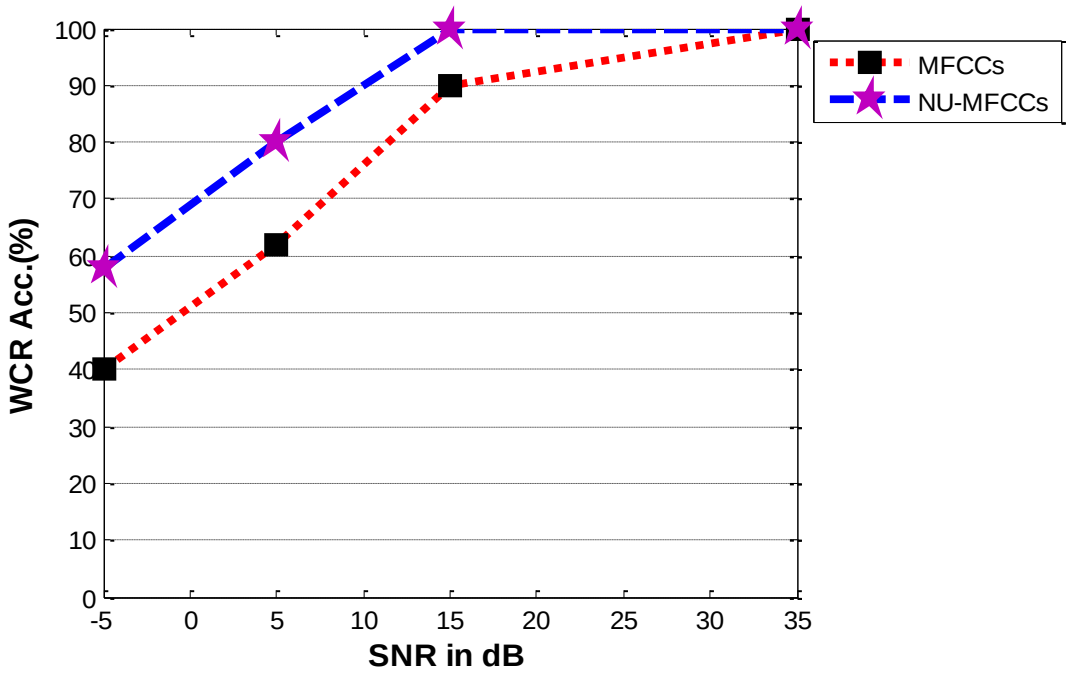


Figure 5.17 Performance Comparison between MFCCs and NU-MFCCs methods for a system trained with a mixture of normal external surrounding speech and AWGN for the word 'five' at additive Volvo noise case.

In general as shown in the above figures (5.14-5.17), the NU-MFCCs methods performance is much better than the MFCCs methods in additive Volvo noise conditions under both training conditions.

And also we can see from the results that AWGN is most damaging as compared to additive pink and Volvo noise for speech 'four' and 'five'.

CHAPTER SIX: CONCLUSION AND RECOMENDATIONS

1. Conclusion

This work presented in this thesis represents an attempt to develop a non-uniform sampling based feature extraction for small vocabulary speech recognition system. We performed many experiment for MFCCs and NU-MFCCs based feature extraction method or acoustic representations to show the better performer in case of normal external surrounding and various signal to noise ratio.

Results show that the recognition rate is better in normal external surrounding speech when we use MFCC for a small number of utterances but when the number of utterance increased the recognition performance of both MFCC and NU-MFCC becomes comparable. For example, at 550 utterances for each MFCCs and NU-MFCCs method, we got the average recognition performance be 92.36 % and 92.18 % respectively under the normal external surrounding speech(>35 dB). In noisy condition, the proposed NU-MFCCs feature extraction method gives satisfied results as compared to MFCC. For instance at additive white Gaussian noise(AWGN) condition(between -5dB and 35dB), the average recognition performance of MFCCs and NU-MFCCs is 42 % and 51.27% respectively for a system trained with a normal external surrounding(>35 dB) and, for a system trained with a mixture of normal external surrounding speech and 10 dB additive white Gaussian noise(AWGN), we got 65.87% and 74.84% respectively.

In addition in MFCCs case, we got an average performance of 47.33%, 41.33% respectively for a system trained at normal external surrounding condition and tested at additive Pink and Volvo noise conditions for the word ‘four’ ,and also 59.33%, 55.33% respectively in NU-MFCCs case.

.

Therefore the proposed acoustic representation or feature extraction method is better for a noisy environment. Even if the proposed method did not show remarkable improvement in real noise environment, if the parameters are optimized, much better recognition rate would be found.

6.2. Recommendation for Future Works

As shown in the results, there is still a need for recognition rate improvements. The main areas for further (future) works are:-

- First, the current database may be expanded to include additional speakers and the vocabulary size may be increased to include possible voice commands considering the development of a speech recognition system. In addition, a "clean" database needs to be collected inside a noise-controlled room to more formally investigate the impact of noise on the system.
- Second, the database may include the Amharic language corpus and the performance can be compared again or with different feature extraction method.
- Third, speech features like Linear Prediction Cepstral Coefficients (LPCC), Perceptual Linear Predictive (PLP) and RASTA have been used in various researches as another means of feature extractions. As a result making the performance comparison between NU-MFCCs methods and the others methods helps to know which methods perform better in which area.
- Fourth, performance improvement may be achieved by implementing different model structures like using Hidden Markov Model (HMM) which is now a day's common method used as a classifies in speech recognition system.
- Fifth, in this thesis we investigated the non-uniform time samples effect on the feature extraction by taking the random samples form the oversampled signals of uniform ADC. But building the non-uniform sampling based ADC helps to see the direct effect of the non-uniform sampling on the speech recognition system or on some specific applications.
- Sixth, mathematical analysis of the non-uniform samples of the features can be done in brief.
- Finally, by applying the non-uniform sampling condition on both time and frequency domain, we may test the effect of sampling on speech recognition system. And also using non-uniform sampling method is now common in signal processing at high frequency application area but still there is a lack of investigation on the use of non-uniform sampling for different application at low frequency. As a result of this, we recommend much work to be done at this area.

APPENDIX A. MATLAB PROGRAMING OF NU-MFCCs BASED FEATURE EXTRACTION

In this appendix we includes the MATLAB programs used only for programming the NU-MFCCs based feature extraction for speech recognition. Some of the MATLAB codes originated from those available at [42,56] has been modified to fit to non-uniform sampling case and we programmed the other necessary code for processing the non-uniform samples. The following source code is programmed by us. The function called in the programmed can be found in reference [42, 56] only by doing some modification like by inputting the non-uniform time values in the function and alike.

1. Non-Uniform Sampling

```
function [x1,t1]=Nonuniform(K,f1)
%programmed by Bisrat Girma
%Adivisor Dr.Eneyew Adugna
% nov-19-2010
%K,oversampled speech signal
%f1,Oversampling frequency
%fs=signal frequency,assuming 8kHz for speech recognition
%f2= frequency of the generated sine-wave ,2*fs

tt=0:(1/(8*f1)):1;
f2=16000;
ttt=int8(sin(2*pi*(f2)*tt));
t2=1;
t3=2;
tt1(1)=0;
i=0;
for k=1:length(K)
   if ttt(1+(i*2))<=K(k)
      tt1(t3)=(k*(1/f1));
      x11(t2)=K(k);
t2=t2+1;
      t3=t3+1;
   end
  i=i+1;
end
for k=1:t2-1
```

```
   x1(k)=x11(k);
   t1(k)=tt1(k);
end
```

---

2. Non-Uniform Framing of the Non-Uniform Samples

```
function [f1,f2]=nonenframet1(x,t,len,s)
%programmed by Bisrat Girma
%Advisor Dr.Eneyew Adugna
% nov-20-2010
%s,oversampled speech data
%x,non-uniformly sampled data
%t,non-uniform time instants of the non-uniform sampled data
%e,quasi-stationary time of the speech signal
if (nargin<4) z=2000;
else z=length(s);
end
e=25*10^-3;
%y=t(z);
nf=ceil(z/662);
if (nargin < 3)
  len= 1103;
end
d=length(x);
f1=zeros(nf,len);
f2=zeros(nf,len);
i=1;m=1;
L=0;
g=0;
for a=1:nf
   %b=1;
   for c=1:441
      f1(a,c)=f1(a-L,c+662);
```

```
      f2(a,c)=f2(a-L,c+662);
   end
   b=c+1;
   for k=m:d
      l=t(k);
      if l<=e+g
         f1(a,b)=x(k);
         f2(a,b)=t(k);
         b=b+1;
         i=k+1;
      end
   end
   g=g+15*10^-3;
   L=1;
   m=i;
end
```

---

3. Non-Uniform Discrete Fourier Transfer

```
function b=ndft(x,t,N)
%programmed by Bisrat Girma
%Advisor Dr.Eneyew Adugna
%nov-24-2010
%x,the non-uniform samples frame
%t,the non-uniform time spaced value frame
%N,length of NDFT
M=length(t);
[c d]=size(x);
f_hat=x;
f=zeros(N,c);
j=1;
for k=0:N-1
   x(j)=(8000*k)/N;
   j=j+1;
```

```
end
m=1;
for z=1:c
for j=1:N
    for k=1:M
     f(j,z)=f(j,z)+f_hat(m,k).*exp(-2*pi*i*x(j)*t(m,k));
        end;
end
m=m+1;
end
b=f;
```

---

4. Non-Uniform Mel Frequency Cepstral Coefficient(NU-MFCC) Computation

```
function c=numelcepst(s,fs,t,w,nc,p,n,inc,fl,fh)
%programmed by Bisrat Girma
%Advisor Dr.Eneyew Adugna
%s,non-uniformly sampled speech signal
%fs,sampling frequency
%t,non-uniform time time instants
%nc  number of cepstral coefficients excluding 0'th coefficient (default 12)
 %   n   length of frame in samples (default power of 2 < (0.03*fs))
 %   p   number of filters in filterbank (default: floor(3*log(fs)) = approx 2.1 per ocatave)
 %   inc  frame increment (default n/2)
 %    fl  low end of the lowest filter as a fraction of fs (default = 0)
 %    fh  high end of highest filter as a fraction of fs (default = 0.5)
 %    w   window('M'   Hamming window in time domain (default))
 %              '0'  include 0'th order cepstral coefficient
 %              'd'    include delta coefficients (dc/dt)
 %              'D'    include delta-delta coefficients (d^2c/dt^2)
 %              'p'    filters act in the power domain
 %              'a'    filters act in the absolute magnitude domain (default)
 %              '0'  include 0'th order cepstral coefficient
 %              'e'  include log energy
 % Outputs:  c     mel cepstrum output: one frame per row. Log energy, if requested, is the
 %               first element of each row followed by the delta and then the delta-delta
```

```
%               coefficients.
 if nargin<2 fs=44100; end
if nargin<3 t=1/fs;end
if nargin<4 w='M'; end
if nargin<5 nc=12; end
if nargin<6 p=floor(3*log(fs)); end
if nargin<7 n=pow2(floor(log2(0.03*fs))); end
if nargin<10
   fh=0.5;
   if nargin<9
     fl=0;
     if nargin<8
        inc=floor(n/2);
     end
   end
end
 if length(w)==0
   w='M';
end
if any(w=='M')
  [z1,z2]=nonenframecutt1(s,t,1103);
  [rrr ccc]= size(z1);
    for tt=1:rrr
       zz1(tt,:)=z1(tt,:).*hamming(ccc)';
    end
else
[z1,z2]=nonenframecutt1(s,t,1103);
  [rrr ccc]= size(z1);
    for tt=1:rrr
       zz1(tt,:)=z1(tt,:).*hamming(ccc)';
    end
end
f=nudftmatrixcut(zz1,z2,1103);
[m,a,b]=melbankm(p,n,fs,fl,fh,w);
pw=f(a:b,:).*conj(f(a:b,:));
pth=max(pw(:))*1E-20;
if any(w=='p')
```

```
   y=log(max(m*pw,pth));
else
   ath=sqrt(pth);
   y=log(max(m*abs(f(a:b,:)),ath));
end
c=rdct(y).';
nf=size(c,1);
nc=nc+1;
if p>nc
   c(:,nc+1:end)=[];
elseif p<nc
   c=[c zeros(nf,nc-p)];
end
if ~any(w=='0')
   c(:,1)=[];
   nc=nc-1;
end
if any(w=='e')
   c=[log(sum(pw)).' c];
   nc=nc+1;
end
 % calculate derivative
 if any(w=='D')
  vf=(4:-1:-4)/60;
  af=(1:-1:-1)/2;
  ww=ones(5,1);
  cx=[c(ww,:); c; c(nf*ww,:)];
  vx=reshape(filter(vf,1,cx(:)),nf+10,nc);
  vx(1:8,:)=[];
  ax=reshape(filter(af,1,vx(:)),nf+2,nc);
  ax(1:2,:)=[];
  vx([1 nf+2],:)=[];
  if any(w=='d')
     c=[c vx ax];
  else
     c=[c ax];
  end
```

```
elseif any(w=='d')
  vf=(4:-1:-4)/60;
  ww=ones(4,1);
  cx=[c(ww,:); c; c(nf*ww,:)];
  vx=reshape(filter(vf,1,cx(:)),nf+8,nc);
  vx(1:8,:)=[];
  c=[c vx];
end
if nargout<1
  [nf,nc]=size(c);
  t=((0:nf-1)*inc+(n-1)/2)/fs;
  ci=(1:nc)-any(w=='0')-any(w=='e');
  imh = imagesc(t,ci,c.');
   axis('xy');
   xlabel('Time (s)');
   ylabel('Nu-Mel-cepstrum coefficient');
   map = (0:63)'/63;
   colormap([map map map]);
   colorbar;
 end
```

APPENDIX B: EXPERIMENTAL RESULTS IN TABLE

SNR in dB		-5	0	5	10	15	20	25	30
Recognition results of MFCC	four	8%	14%	38%	44%	60%	74%	90%	96%
	five	10%	16%	56%	62%	80%	88%	94%	100%
Recognition results of NU-MFCC	four	8%	36%	48%	56%	98%	100%	100%	100%
	five	12%	54%	68%	70%	98%	100%	100%	100%

Table B.1 Recognition results of MFCC and NU-MFCC based feature extraction for the speech 'four' and 'five' for a system trained with normal external surrounding

SNR in dB		-5	0	5	10	15	20	25	30
Recognition results of MFCC	four	58%	64%	72%	76%	94%	100%	100%	100%
	five	60%	70%	78%	82%	98%	100%	100%	100%
Recognition results of NU-MFCC	four	62%	72%	84%	92%	100%	100%	100%	100%
	five	70%	80%	88%	90%	100%	100%	100%	100%

Table B.2 Recognition results of MFCC and NU-MFCC based feature extraction for the speech 'four' and 'five' for a system trained with a mixture of normal external surrounding and white noise.

SNR in dB	-5	5	15	Normal external surrounding(>35)
MFCC Performance in WCR Acc. (%)	9.82%	42.18%	74%	92.36%
NU-MFCC Performance in WCR Acc (%)	12.18%	54.54%	87.09%	92.18 %

Table B.3 Over all recognition results of MFCC and NU-MFCC based feature extraction for a system trained with a normal external surrounding speech.

SNR in dB	-5	5	15	Normal external surrounding (>35)
MFCC performance in WCR Acc.(%)	43.27	65.82	88.54	92.36
NU-MFCC performance in WCR Acc(%)	56.91	76.91	90.73	92.18

Table B.4 Over all recognition results of MFCC and NU-MFCC based feature extraction for a system trained with a mixture of normal external surrounding speech and additive white noise.

SNR in dB		-5	5	15	Normal(>35)
MFCC	Four	12	50	80	100
	Five	14	64	86	100
NU-MFCC	Four	16	62	100	100
	Five	20	78	100	100

Table B.5 Recognition results of MFCC and NU-MFCC based feature extraction for the speech ‘four’ and ‘five’ for a system trained with normal external surrounding speech, using pink noise as additive noise

SNR in dB		-5	5	15	Normal(>35)
MFCC	Four	34	58	86	100
	Five	40	66	92	100
NU-MFCC	Four	54	78	100	100
	Five	62	84	100	100

Table B.6 Recognition results of MFCC and NU-MFCC based feature extraction for the speech ‘four’ and ‘five’ for a system trained with mixture normal external surrounding speech, using pink noise as additive noise

SNR in dB		-5	5	15	Normal(>35)
MFCC	Four	10	44	70	100
	Five	12	60	88	100
NU-MFCC	Four	12	54	100	100
	Five	16	74	100	100

Table B.7 Recognition results of MFCC and NU-MFCC based feature extraction for the speech 'four' and 'five' for a system trained with normal external surrounding speech, using Volvo noise as additive noise

SNR in dB		-5	5	15	Normal(>35)
MFCC	Four	32	56	82	100
	Five	40	62	90	100
NU-MFCC	Four	52	70	100	100
	Five	58	80	100	100

Table B.8 Recognition results of MFCC and NU-MFCC based feature extraction for the speech 'four' and 'five' for a system trained with a mixture of normal external surrounding and additive white Gaussian noise , using Volvo noise as additive noise.

REFERENCES

[1] A.V. Oppenheim, L.R. Rabiner and R.W Schafer, "*Digital Processing of Speech Signal* " , Prentice Hall, New Jersey 1978.

[2] I.Bilinskis, "*Digital Alias-free Signal Processing*", John Wiley and Sons Ltd, 2007.

[3] I.Bilinskis and A.Mikelsons, "*Randomized Signal Processing",* Prentice-Hall International (UK) Ltd, 1992.

[4] F.Marvasti, "*A Unified Approach to Zero-Crossings and Nonuniform Sampling of Single and Multidimensional Signals and Systems*", Princeton, NJ: NONUNIFORM Publication, 1987.

[5] EURODASP, "NONUniform Sampling:AN1", September 2001 *www.edi.lv/media/uploads/UserFiles/dasp-web/.../**SAMPLING**.PDF*

[6] C.Becchetti and L.P. Ricotti, "*Speech Recognition"*, John Wiley & Sons Ltd., 1999.

[7] B. Gold and N. Morgan, "*Speech and Audio Signal Processing"*, John Wiley & Sons,2000

[8] L. R. Rabiner and B-H. Juang, "*Fundamentals of Speech Recognition",* Prentice Hall,1993.

[9] J.Deller, J.Proakis and J.Hansen "Discrete-Time Processing of Speech Signals", *Wiley Interscience*, 2000

[10] R.S. Kurcan, "Isolated Word Recognition from IN-EAR Microphone Data using Hidden Markov Models(HMM)," Master Thesis, Naval Postgraduate School, California,March 2006

[11] Alpay Koc,"Acoustic Feature Analysis for Robust Speech Recognition," Master Thesis, Bogazici University, 2002

[12] Q.Zhu and A.Alwan, "The Effect of Additive Noise on Speech Amplitude Spectra : A Quantitative Analysis*," IEEE Signal Processing Letters*, vol 9, no.9 , September 2002

[13] U. Shrawankar and V.Thakare "Feature Extraction For Speech Recognition System In Noisy Environment :A study*" ICCEA,* vol.1,No.pp.358 -361, Mar.2010.

[14] F.Soong, "Non-Uniform Sampling based IWASR", *The Journal of the Acoustical Society of America*, vol. 72, issue S1, p. S31, 1982.

[15] Montri Karnjanadecha, "Signal Modeling with Non-Uniform Time Sampling of Features for Automatic Speech Recognition", Ph.D dissertation, Old Dominion University,2000.

[16] H.S.Shapiro and R.A.Silverman, "Alias-free sampling of random noise", *SIAM J. Appl. Math.*, vol.8,no.2, pp.245–248,1960.

[17] W.M.Brown, "Sampling with random jitter", *SIAM J. Appl. Math.*, vol.11, no., pp.460–473, 1963.

[18] M.Lim and C.Saloma, "Direct signal recovery from threshold crossings", *Physical Review E*, vol.58, no.5, 1998.

[19] Y.Y.Zeevi, A.Gravriely, and S.Shamai-Shitz, "Image representation by zero and sine-wave crossings", *J. Opt. Soc. Am. A*, vol.4, pp.2045–2060, 1987.

[20] I.Bilinskis and A.Mikelsons, "Application of randomized or irregular sampling as an anti-aliasing technique", *Signal Processing Theories and Application*, vol. 5, no., Elsevier Science Publishers, pp. 505–8, Amsterdam,1990.

[21] A. J. W. Duijndam and M. A. Schonewille, "Nonuniform fast Fourier transform", *Geophysics*, vol.64,no.,pp.539-551,1999.

[22] S. Bagchi and S. K. Mitra, "*Nonuniform Discrete Fourier Transform and its Signal Processing Applications*", Norwell, MA: Kluwer, 1999.

[23] D. Potts, G. Steidl, and M. Tasche, "Fast Fourier transforms for nonequispaced data: A tutorial," In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, pp. 247-270, Boston, Birkhäuser, 2001.

[24] J. O. S. Smith III, "*Mathematics of the discrete Fourier transform (DFT")*, W3K Pub,2003.

[25] A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data", *SIAM J. Sci. Stat. Comput.*, vol. 14, no., pp. 1368-1393, 1993.

[26] Y.Shen,C.Zhu, L.Liu, Q.Wang,J.Jin andY.Lin; , "Explicit solution for nonuniform discrete fourier transform," *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE* , vol., no., pp.1505-1509, 3-6 May 2010.

[27] T. Bowles and J.E. Erickson, "An examination of frequency indexes used in the non-uniform DFT," *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, vol., no., pp.77- 80, 23-25 May,2010.

[28] A.Tarczynsk, "Spectrum Estimation of Nonuniformly Sampled Signals," *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol.2, no., pp. 795-798 ,2002.

[29] V. E. Neagoe, "Spectral Estimation Using Nonuniform Sampling in the Time and Frequency Domains", *Proc. of the Latvian Signal Processing International Conference (LSPIC-90)*, Riga, April 1990.

[30] N.Nguyen and Q.H.Liu, "The regular Fourier matrices and nonuniform fast Fourier transforms", *SIAM J. Sci. ComPut.*, vol. 21,no.1, pp. 283-293, 1999.

[31] J.A.Fessler and B.P.Sutton, "Nonuniform fast Fourier transforms using minmax interpolation", *IEEE Transactions on Signal Processing*, vol. 51,no.2, pp. 560-574, 2003.

[32] Frida Eng, "Non-Uniform Sampling in Statistical Signal Processing", Ph.D dissertations, Linköping Studies in Science and Technology, Linköping, Sweden ,2007.

[33] E.Masry, "Spectral estimation of continuous-time processes: performance comparison between periodic and Poisson sampling schemes", *IEEE Trans. Autom. Control*, AC-vol.23,no.4, pp.679–685, 1978.

[34] F.Gunnarsson, F.Gustafsso and, F.Gunnarsson, "Frequency Analysis Using Non-uniform Sampling with Application to Active Queue Management", Linkoping University, SE-581 83 Linkoping, Sweden, 2003.

[35] A. J. W. Duijndam and M. A. Schonewille, "Nonuniform fast Fourier transform", *Geophysics,* vol.64,no.,pp.539-551,1999.

[36] M.Bronstein and A.Bronstein, "THE NON-UNIFORM FFT AND SOME OF ITS APPLICATIONS", *ISBI 2002 conference.*

[37] Wikipedia, "Feature Extraction" http://en.wikipedia.org/wiki/Feature_Extraction

[38] R. Vergin, "An algorithm for robust signal modeling in speech recognition," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Vol. 2, pp. 969-972, May 1998.

[39] J. W. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE,* vol. 81, no. 9, pp. 1215-1247, September 1993.

[40] L. Deng and D. O'Shaughnessy, "*Speech Processing A Dynamic and Optimization- Oriented Approach",* Marcel Dekker, New York, 2003.

[41] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357-366, 1980.

[42] M.Brookes, "VOICEBOX:a MATLAB toolbox for speech processing" www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2003

[43] Volkan Tunali,"A Speaker Dependent ,Large Vocabulary , Isolated Word Speech Recognition System for Turkish," Master Thesis, Marmara University, iSTANBUL, 2005.

[44] C.Lee, D.Hyun, E.Choi, J.Go, and C.Lee "Optimizing Feature Extraction for Speech Recognition*", IEEE Transactions on speech and audio processing,* vol.11, no.1, Jan.2003.

[45] H.Hermansky and N. Morgan, "RASTA Processing of Speech", *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, October 1994.

[46] S.Frui "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum", *IEEE Trans. ASSP*., vol.34, no.1., pp. 52-59, 1986.

[47] T.Applebaum and B.Hanson, "Regression Features for Recognition of Speech in Noise", *Proc. ICASSP*., S14.26, Toronto, 1991.

[48] S.Memon and M.Lech,"Using Information Theoretic Vector Quantization for GMM Based Speaker Verification," *EUSIPCO 2008*, Lausanne, Switzerland, August 2008.

[49] "Text Independent Speech Recognition," http://users.ece.utexas.edu/cvbeums/courses/ee382c/projects/fall99/phan-soong/litsurvey.pdf

[50] Douglas Reynolds, "Gaussian Mixture Models," www.ll.mit.edu/mission/.../0802_Reynolds_Biometrics-Gmm.pdf

[51] Andrew I. Hanna,"Gaussian Mixture Models - Algorithm and Matlab Code," www.sagoforest.com/sagoalephpapers/GaussianMixtureModels.pdf ,November 2006.

[52] D.A.Reynolds and R.C.Rose, "Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models" *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol.3,no.1, pp. 72–83,1995.

[53] A.Dempster, N.Laird and D.Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*,vol. 39,no.1, pp.1–38,1977.

[54] I.McCowan, D.Moore, J.Dines, D.Gatica-Perez, M.Flynn, P. Wellner and H.Bourlard "On the Use of Information Retrieval Measures for Speech Recognition Evaluation" *IDIAP(RR 04-73)*,March 2005.

[55] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics - Doklady 10*, vol. 10, pp. 707-710, 1966. (CNRS-URA 820), Paris, France, 2001.

[56] O.Cappé, "H2M: A set of MATLAB/OCTAVE functions for the EM estimation of mixtures and hidden Markov models", ENST dpt. TSI / LTCI

[57] Y.A.Alotaibi, M.Alghamdi and F.Alotaiby,"Speech Recognition System of Arabic Digits based on a Telephony Arabic Corpus," King Saud University , Riyadh, Saudi Arabia,

[58] Wikipedia ,"Nonuniform Sampling" http://en.wikipedia.org/wiki/Non_Uniform_Sampling