**CHAPTER-1**

**INTRODUCTION**

**CHAPTER-1**

INTRODUCTION

For over the long period in the human history, speech has been the most dominant and convenient means of communication between people. Today, speech communication is not only for face-to-face interaction, but also between individuals at any moment, anywhere, via a wide variety of modern technological media, such as wired and wireless telephony, voice mail, satellite communications and the Internet. With the rapid development of communication technologies, a promising speech communication technique for human-to-machine interaction has come into being. Automatic speech recognition (ASR) is the core challenge towards the natural human-to-machine communication technology.

Automatic speech recognition aims to automatically convert a speech waveform into a sequence of words by machine[1]. Currently, there have been a number of successful commercial ASR products. However, many problems still exist in real-world ASR applications. The recognition accuracy of a machine is, in most cases, far from that of a human listener, and its performance would degrade dramatically with small modification of speech signals or speaking environment. Due to the large variation of speech signals, speech recognition inevitably requires complex algorithms to represent this variability. As a result more computation and memory capacity are needed. But for real-world applications, such as embedded ASR systems, only limited resources are available. The goal is focused on the exploitation of various optimization methods for state-of-the-art ASR techniques for DSP-based embedded applications while retaining high recognition accuracy.

**CHAPTER-2**

**HISTORY OF ASR DEVELOPMENT**

**CHAPTER-2**

HISTORY OF ASR DEVELOPMENT

The earliest attempts to implement automatic speech recognition on machine began in the 1950s. The first significant ASR system was built at Bell Labs in 1952 by Davis, Biddulph and Balashek for isolated digit recognition[2]. In this system, only one single speaker can be recognized based on measuring the spectral resonances during the vowel region of each digit. In the 1960s and 1970s, a great deal of fundamental ideas in speech recognition emerged and were published. These new techniques include fast Fourier transform (FFT)[3], cepstral analysis[4], and linear predictive coding (LPC)[5], for spectral estimation and feature extraction, as well as dynamic time warping (DTW)[6] and Hidden Markov Models (HMMs)[7] for pattern matching and acoustic modeling. While isolated word recognition (IWR) was investigated in the 1970s, the problem of connected word recognition was the focus in the 1980s. In addition, the approach of pattern recognition shifted from template-based to statistical modelling methods. In particular, the HMM approach was researched by Bell Labs[8], CMU[9], IBM[10], etc.

Since 1990s, people have moved their interests to the difficult task of Large Vocabulary Continuous Speech Recognition (LVCSR) and indeed achieved a great progress. Speech recognition has been developed from theoretical methods to practical systems. Meanwhile, many well-known research and commercial institutes have established their recognition systems including Via Voice system by IBM, HTK system by the University of Cambridge and Whisper system by Microsoft, etc.

**CHAPTER-3**

**FUNDAMENTAL OF**

**AUTOMATIC SPEECH RECOGNITION**

**CHAPTER-3**

FUNDAMENTAL OF AUTOMATIC SPEECH RECOGNITION

**3.1 How do humans do it?**

An acoustic wave is an oscillation of pressure that travels through a solid, liquid, or gas in a wave pattern. It transmits sound by vibrating organs in the ear that produce the sensation of hearing. Acoustic waves, or sound waves, are defined by three characteristics: wavelength, frequency, and amplitude.

The wavelength is the distance from the top of one wave’s crest to the next. The frequency of a sound wave is the number of waves that pass a point each second. Sound waves with higher frequencies have higher pitches than sound waves with lower frequencies. Amplitude is the measure of energy in a sound wave and affects volume. The greater the amplitude of an acoustic wave, the louder the sound.

An acoustic wave is what makes humans and other animals able to hear. A person’s ear perceives the vibrations of an acoustic wave and interprets it as sound. The outer ear, the visible part, is shaped like a funnel that collects sound waves and sends them into the [ear canal](http://www.wisegeek.com/what-is-the-ear-canal.htm) where they hit the ear drum, which is a tightly stretched piece of skin that vibrates in time with the wave. The ear drum starts a chain reaction and sends the vibration through three little bones in the [middle ear](http://www.wisegeek.com/what-is-the-middle-ear.htm) that amplify sound. Those bones are called the hammer, the anvil, and the stirrup.

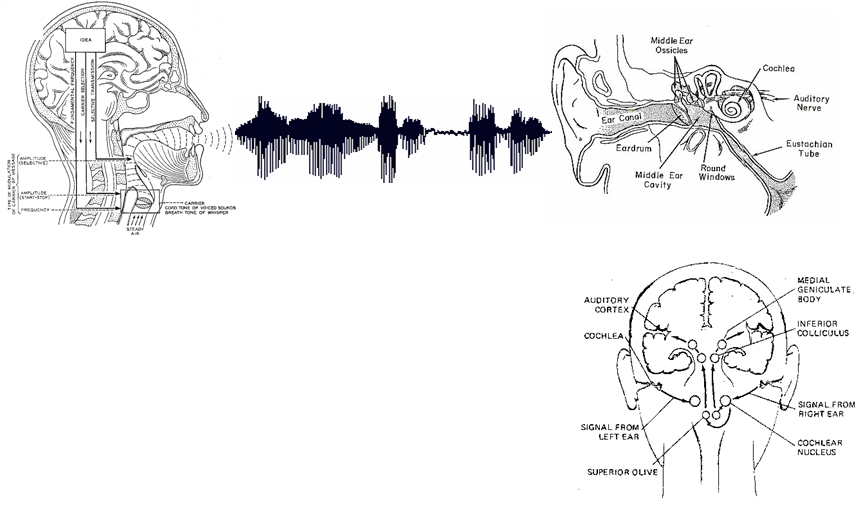
****

Figure 3.1: Voice interpretation by human ear

From here, the vibrations of the sound wave are turned into electrical impulses that the brain can interpret. The stirrup bone presses against the fluid-filled cochlea, or hearing organ, in time to the acoustic wave. The fluid inside the cochlea moves because of the stirrup bone’s pressure, and in turn, moves a hair-cell-lined membrane buried within the fluid.

The hair cells move according to the pattern of the acoustic wave, sending signals to nerve cells that carry their interpretation of the wave to the brain. The brain interprets the acoustic wave as sound and as a result, we hear. The human brain likes patterns, and it is interesting to note that it interprets regular sound wave patterns as pleasant and irregular wave patterns as nothing more than unpleasant noise.

**3.2 How might computers do it?**

Speech recognition is a multileveled pattern recognition task, in which acoustical signals are examined and structured into a hierarchy of subword units (e.g., phonemes), words, phrases, and sentences. Each level may provide additional temporal constraints, e.g., known word pronunciations or legal word sequences, which can compensate for errors or uncertainties at lower levels. This hierarchy of constraints can best be exploited by combining decisions probabilistically at all lower levels, and making discrete decisions only at the highest level.

* Digitization
* Acoustic analysis of the speech signal
* Linguistic interpretation

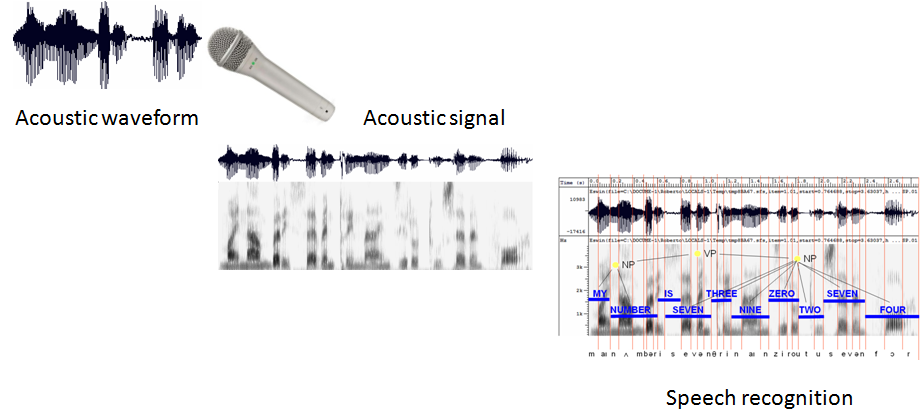


Figure 3.2: Voice interpretation by Computers.

**3.3 Classification of ASR systems**

A speech recognition system can operate in many different conditions such as speaker dependent/independent, isolated/continuous speech recognition, for small/large vocabulary[11].

3.3.1 Speaker-dependent versus independent system

A speaker-dependent system is a system that recognizes a specific speaker's speech while speaker-independent systems can be used by any unspecified speaker.

3.3.2 Isolated versus continuous speech recognition system

In an isolated word recognition system, each word (note that word may be a simple utterance) is assumed to be surrounded by silence or background noise. This means that both sides of a word must have no speech input, making definite word boundaries easy to construct. This kind of recognition is mainly used in applications where only a specific digit or a word needs to be dictated. Connected speech (or more correctly ‘connected utterances’) recognition is similar to isolated word recognition. But it allows several words/digits to be spoken together with minimal pause between them. Longer phrases or utterances are therefore possible to be recognized.

Continuous speech recognition is much more natural and user-friendly. The system is able to recognize a sequence of connected words, which are not separated by pauses, in a sentence. This mode requires much more computation time and memory, and it is more difficult to operate than isolated word recognition mode for the following reasons:

* Speakers' pronunciation is less careful.
* Speaking rate is less constant.
* Word boundaries are not necessarily clear.

Therefore the accuracy of continuous speech recognition is usually low compared with the preceding modes.

3.3.3 Small versus large vocabulary size

It is clear that the smaller the vocabulary size, the higher the recognition accuracy. Usually we classify the difficulty level of implementing a speech recognition system with a score from 1 to 10 according to Table 3.1, where 1 means the simplest system (speaker-dependent, able to recognize isolated words in a small vocabulary (10 words)) and 10 correspond to the most difficult task (speaker-independent continuous speech over a large vocabulary (say, 10,000 words)). State-of-the-art speech recognition systems with acceptable error rates are somewhere between these two extremes[12].

|  |  |  |
| --- | --- | --- |
|  | **ISOLATED WORD** | **CONTINUOUS WORD** |
| **Speaker Dependent** | Small Voc 1  Large Voc 4 | Small Voc 5  Large Voc 7 |
| **Multi-Speaker** | Small Voc 2  Large Voc 4 | Small Voc 6  Large Voc 7 |
| **Speaker Independent** | Small Voc 3  Large Voc 5 | Small Voc 8  Large Voc 10 |

**Table 3.1**: Classification of speech recognition difficulties

**3.4 Automatic speech recognition process**

Fig. 3.3 illustrates the architecture of a typical speech recognition system that employs today's main-stream approach. The acoustic models represent the acoustic properties, phonetic properties, microphone and environmental variability, as well as gender and dialectal differences among speakers. The language models contain the syntax, semantics, and pragmatics knowledge for the intended recognition task. These models can be dynamically modified according to the characteristics of the previously recognized speech. This is referred to as model adaptation.

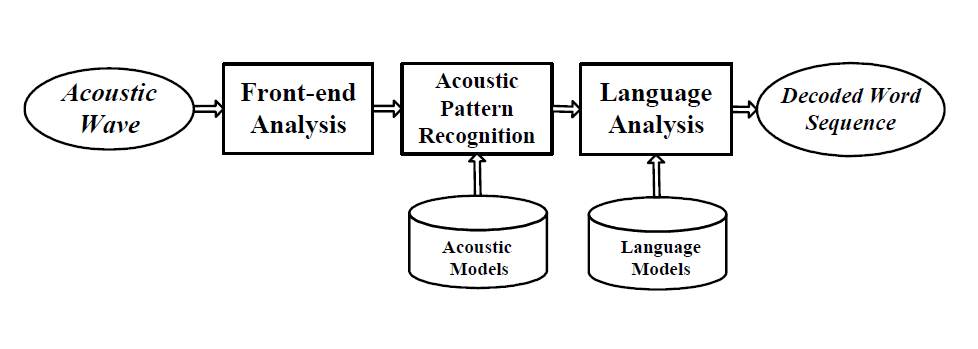


Figure 3.3 General block diagram of an automatic speech recognition system.

3.4.1 Front-end analysis

Front-end analysis, also referred to as acoustic analysis or feature extraction, is the first step in an automatic speech recognition system. This process aims to extract acoustic features from the speech waveform. The output of front-end analysis is a compact, efficient set of parameters that represent the acoustic properties observed from input speech signals, for subsequent utilization by acoustic modelling.

There are three major types of front-end processing techniques:

* Linear predictive coding. (LPC)[13]
* Mel-frequency cepstral coefficients. (MFCC)[14]
* Perceptual linear prediction. (PLP)[15]

3.4.2 Acoustic pattern recognition

Acoustic pattern recognition aims at measuring the similarity between an input speech and a reference pattern or a model (obtained during training) and accordingly determines a reference or a model, which best matches the input speech, as an output. One approach of acoustic pattern matching is called the dynamic time warping (DTW)[16]. DTW is a method which measures the distance between each input frame and each reference frame using the dynamic programming algorithm to find the best warping of the pattern, and determines the best match by minimizing the distance between the input frame and the reference frame.

Another approach is based on statistical models. The most successful one is hidden Markov models (HMMs)[17][18][19], which characterize speech signals using a pre-trained “hidden" Markov chain. In the training stage, one or more HMMs corresponding to speech sounds of the same class (e.g., phones, words, phrases) are designed and optimized to represent the statistical features of that class. In the recognition stage, probabilistic measures are taken to calculate how much an unknown input speech matches the given set of HMMs. There are many different kinds of HMM structures, among which the discrete HMM (DHMM), continuous-density HMM (CDHMM) are the most popular and successful.

3.4.3 Language analysis

In recent years language analysis is becoming more and more important in speech recognition, especially for large vocabulary continuous speech recognition (LVCSR) tasks. The speech decoding process needs to invoke knowledge of pronunciation, lexicon, syntax, and pragmatics in order to produce a satisfactory text sequence for further interpretation. In particular, the probabilistic N-gram language models, a specific form of probabilistic FSG language models, has been widely used due to its ease of implementation and coherence with the structure of an HMM.

**3.5 Performance measurements of ASR**

The performance of an ASR system mainly consists of three major parts: recognition accuracy, complexity and robustness.

3.5.1 Recognition accuracy

Recognition accuracy is the most important and straightforward measure of speech recognition performance. Empirically, the collected speech data are partitioned into training set and test set. The training set, which usually contains most of the available data, is utilized for parameter estimation of the acoustic models. The remaining data form the test set, which is used to measure the ASR performance over signals unavailable during training. The recognition accuracy of training set and test set is evaluated by the training-set word error rate and test set word error rate, respectively, where the latter is a critical design target for most ASR systems. However, as the recognition accuracy for new data is not available in real implementations, the design objective is usually converted to minimizing both the training-set word error rate and the performance difference between the training set and the test set.

3.5.2 Complexity

Complexity is another issue that needs to be considered in most commercial ASR systems, especially when hardware cost is critical to system success. In general, the complexity of an ASR system refers to its computational complexity and model complexity. Computational complexity concerns the cost of execution time in each module of the system. For most practical implementations where the ASR operation is required to be finished in real time, computational complexity should definitely be well considered. Model complexity is usually measured by the number of distinct model parameters. There is a tradeoff between the model complexity and recognition accuracy. A reduced model complexity can provide the obvious benefits of savings in memory and computation while the recognition accuracy will drop down at the same time.

3.5.3 Robustness

While accuracy is crucial to speech recognition performance, robustness is also of great importance to an ASR system. At present, most ASR systems are trained upon a set of speech samples collected under some intended conditions. They would perform well if the operating conditions match the intended conditions. Unfortunately, this assumption is often violated after system deployment, as variation in operating conditions is virtually inevitable. Important aspects of operating conditions include the level of background noise, channel noise and distortion, speaker difference, speaking style and syntactic deviation, spontaneity of speech, etc. In practice, the deviation of these conditions from those assumed during the design phase may result in substantial degradation in performance. This has been the cause of increasing concerns about ASR robustness in recent years[20], and it has in fact become a critical performance index of almost all speech recognition systems.

**3.6 Motivation and goal of this work**

After nearly sixty years of research, speech recognition technology has reached a relatively high level. However, most state-of-the-art ASR systems run on desktop with powerful microprocessors, ample memory and an ever-present power supply. In these years, with the rapid evolvement of hardware and software technologies, ASR has become more and more expedient as an alternative human-to-machine interface that is needed for the following application areas:

* Stand-alone consumer devices such as wrist watch, toys and hands-free mobile phone in car where people are unable to use other interfaces or big input platforms like keyboards are not available.
* Single purpose command and control system such as voice dialing for cellular, home, and office phones where multi-function computers (PCs) are redundant.

There is a great request for the implementation of automatic speech recognition for these consumer applications.

As a matter of fact, the development has been impeded by a number of issues: computational costs, robustness in adverse environments, etc. One solution for the first issue is to use embedded systems which aim to provide low-power and high speed characteristics. Among various embedded systems, Digital Signal Processor (DSP) is one of the most popular embedded platforms on which computationally intensive algorithms can be implemented. It has the advantage of providing good flexibility and relatively short application development cycle. Constrained by the current hardware technologies, reliable recognition of large-vocabulary fluent speech is still not state of art, but reliable and highly accurate isolated/connected word recognition (IWR/CWR) with small limited vocabulary (tens of words) systems should be achievable. However, the algorithms of IWR/CWR are developed using many floating point and complex functions such as integer divisions, logarithms, cosines and conditional loops. Low cost embedded systems generally can not fast process these functions because they don't have specific hardware structures to handle them. Optimizations of the recognition algorithms for low-cost hardware are necessary. It is important to implement them with no floating-point and minimal complex functions to reduce computational complexity. In fact, there lies a trade-off that algorithm optimization not only makes the implementation run faster, it degrades the recognition accuracy as well. One goal of this research is to deeply understand the ASR algorithms and investigate the possible algorithmic optimizations to improve the ASR accuracy, while controlling both the model complexity and computational complexity with acceptable performance degradation. In recent years, there are some researches on code optimizations for speech recognition on low cost fixed-point embedded systems (DSP or others)[21][22]. They did have outlined some optimization techniques for front-end and/or pattern recognition parts. But unfortunately, they did not give a full presentation of the optimization techniques for the whole recognition system and furthermore there is still no such a framework that generally summarizes the optimizations and the relationship between the optimizations and the recognition performance.

Another goal of this research is intensively setting up a framework which could help the system designers who want to realize real-time ASR on DSP estimate the maximal recognition performance they could achieve within their limited hardware recourses.

In a real-world environment, the background noise and when the speaker speaks and stops are both unknown. We need to explore some practical techniques to make the speech recognition system more adaptive to the environment for practical implementations.

**CHAPTER-4**

**PROPOSED WORK USING SIGNAL PROCESSING**

**TECHNIQUES FOR**

**FRONT-END**

**CHAPTER-4**

PROPOSED WORK USING SIGNAL

PROCESSING TECHNIQUES FOR FRONT-END

Fig. 4.1 shows a block diagram of the speech recognition system used in our research. A speech recognition system can be roughly divided into two parts, namely front-end and pattern recognition.

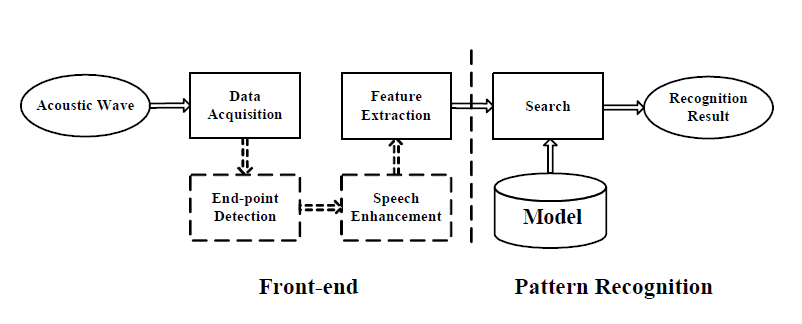


Figure 4.1: Block diagram of ASR system

Typically, the front-end building block includes two modules, namely data acquisition and feature extraction. As an optional choice, the end-point detection and speech enhancement module can be inserted to make the speech signal more adaptive and robust to the noise.

The data acquisition module usually contains a microphone and a codec from which digitized speech data are generated. For DSP applications, speech data is usually represented as a sequence of 16-bit long signed integers. So this module mainly depends on the hardware system.

**4.1 Basic feature extraction principles**

Fig. 4.2 is the detailed block diagram of the feature extraction processing. Feature extraction is done on short-time basis. The speech signal is divided into overlapped fixed-length frames. From each frame, a set of frequency-domain or cepstral-domain parameters are derived to form the so-called feature vector. In the following subsections, some basic principles and analysis techniques used in the feature extraction module will be carried out.

4.1.1 Pre-emphasis

The digitized speech signal, *s*(*n*), derived from the data acquisition module passes through a pre-emphasis process, which performs spectral flattening with a first-order FIR filter[8]:

In this consideration, the output of the pre-emphasis network, (n)

where  *α* is the pre-emphasis parameter (a most common value for α is about 0.95).

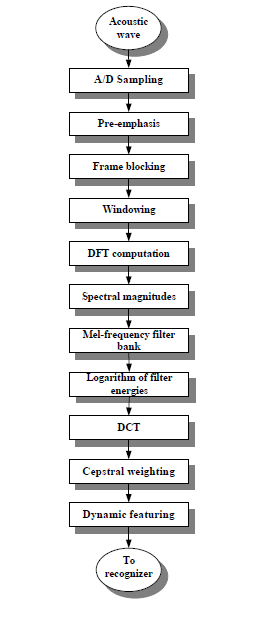


Figure 4.2: Block diagram of MFCC front-end analysis.

4.1.2 Frame blocking and windowing

In this step the pre-emphasized speech signal, , is segmented into frames, which are spaced 20-30 msec apart, with 10-15 msec overlaps for short-time spectral analysis. Each frame is then multiplied by a fixed length window[*h* (*n*)] for n=0 to No-1, where *N*o is the length of a frame.

Window functions are signals that are concentrated in time, often of limited duration *N*o. While window functions such as triangular, Kaiser, Barlett, and prolate spheroidal occasionally appear in digital speech processing systems, Hamming and Hanning are the most widely used to taper the signals to quite small values (nearly zeros) at the beginning and end of each frame for minimizing the signal discontinuities at the edge of each frame[23].

In this, Hamming window, which is defined in Eq. 4.3, is applied.

The output speech signal of Hamming windowing can be described as

4.1.3 Discrete Fourier Transform (DFT) computation

After windowing the speech signal, Discrete Fourier Transform (DFT) is used to transfer these time-domain samples into frequency-domain ones. There is a family of fast algorithms to compute the DFT, which are called Fast Fourier Transforms (FFT). Direct computation of the DFT from Eq. 4.5 requires N\*N operations, assuming that the trigonometric functions have been pre-computed. Meanwhile, the FFT algorithm only requires on the order of *N* log2 *N* operations, so it is widely used for speech processing to transfer speech data from time domain to frequency domain.

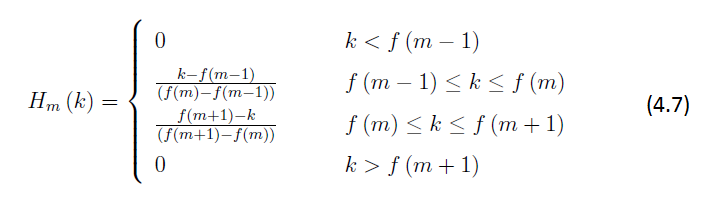
If the number of FFT points, *N*, is larger than the frame size *N*0, *N-N*0 zeros are usually inserted after the *N*0 speech samples.

4.1.4 Spectral magnitudes

Generally speaking, the signal *X* (*k*) is a complex value containing the real and image parts. But in the speech recognition system which deals with the real speech signal, the complex value is always ignored by researchers. Therefore, only the magnitude of the complex value *X* (*k*) is utilized in this situation. If we assume the real and image parts of *X* (*k*) are Re (*X* (*k*)) and Im (*X* (*k*)), then the spectral magnitude of the speech signal should be

4.1.5 Mel-frequency filterbank

In order to represent the static acoustic properties, the Mel-Frequency Cepstral Coefficient (MFCC) is used as the acoustic feature in the cepstral domain. This is a fundamental concept which uses a set of non-linear filters to approximate the behavior of the auditory system. It adopts the characteristic of human ears that human is assumed to hear only frequencies lying on the range between 300Hz to 3400Hz. Besides, human's ears are more sensitive and have higher resolution to low frequency compared to high frequency. Therefore, the filterbank should be defined to emphasize the low frequency over the high frequency. Eq. 4.7 is the filterbank with *M* filters (*m=1,2,, M*), where filter *m* is the triangular filter given by:



which satisfies

The central frequency of each mel-scale filter is uniformly spaced below 1 kHz and it follows a logarithmic scale above 1 kHz as shown in Eq. 4.8 and Fig. 4.3. More filters process the spectrum below 1 kHz since the speech signal contains most of its useful information such as first formant in lower frequencies (Fig. 4.4).

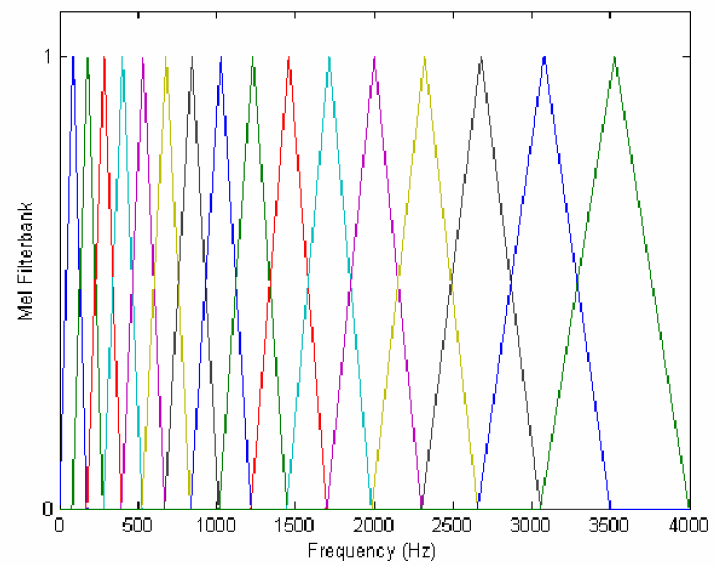


Figure 4.3: Frequency response of mel-filter bank

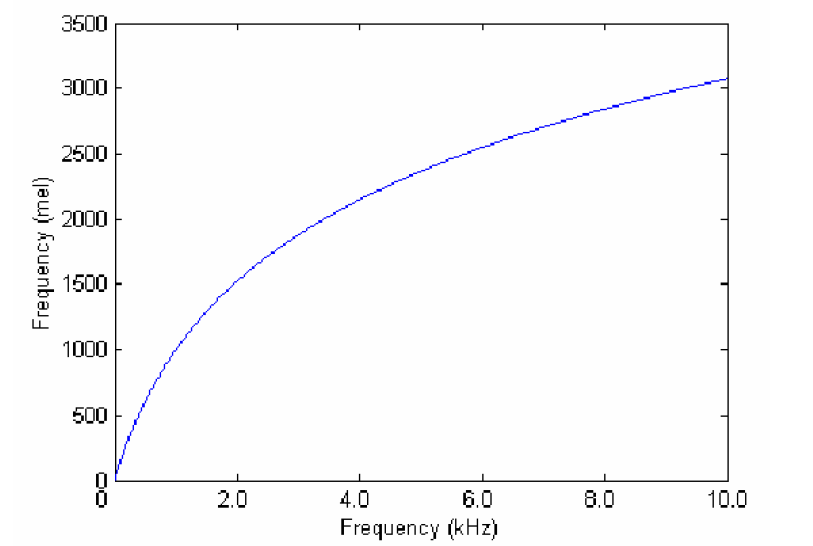


Figure 4.4: Mel scale of central frequency in each filter

If we define *fl* and *fh* be the lowest and highest frequencies of the filterbank in Hz, *Fs* the sampling frequency in Hz, *M* the number of filters and *N* the size of FFT, the centering frequency *f* (*m*) of the *mth* filterbank is:

where the Mel-scale Mel is given by Eq. 4.8 and its inverse

4.1.6 Logarithm of filter energies

After passing through the filterbanks, the log-energy at the output of each filter is calculated as shown in Eq. 4.11

Human ears also smooth the spectrum and use logarithmic scale approximately.

4.1.7 Discrete Cosine Transformation (DCT)

The inverse DFT is performed on the output of the filterbank. Since the log power spectrum is symmetric and real, the inverse DFT is reduced to discrete cosine transformation (DCT). This transformation decorrelates features, which leads to using diagonal covariance matrices instead of full covariance matrices while modeling the feature coeffcients by linear combinations of Gaussian functions. Therefore complexity and computational cost can be reduced. This is especially useful for 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 coefficients, *K*, for recognition ranges between 8 and 13. The equation is as following

4.1.8 Cepstral Weighting

A weighting window (named liftering window) is applied after decorrelating the cepstral coefficients. In this work, the sinusoidal lifter (as shown in Eq. 4.13) is utilized to minimize the sensitivities by lessening the higher and lower cepstral coefficients.

4.1.9 Dynamic featuring

In addition to the cepstral coefficients, the time derivative approximations are used as feature vectors to represent the dynamic characteristic of speech signal. To combine the dynamic properties of speech, the first and/or second order differences of these cepstral coefficients may be used which are called the delta and delta-delta coefficients. And these dynamic features have been shown to be beneficial to ASR performance[24]. The first-order delta MFCC may be described as

where denotes the *kth* cepstral coefficient at frame *t* after liftering and *P* is typically set to the value 2 (i.e., five consecutive frames are involved). The second-order delta MFCC is obtained in a straight forward manner.

In this work, we use the MFCCs, energy and their first-order delta coefficients to form the feature vectors Ot; t={1,2,...T}. It is worth noting that we do not add the second-order delta coefficients, because they do not show significant improvement in our task and it would definitely increase the computational cost and need more memory to store the model parameters. Also, we apply end-point detection and a speech enhancement technique, namely spectral subtraction to the front-end in an attempt to make the recognition system more robust to noise.

**CHAPTER-5**

**PRACTICAL PROBLEMS FACED**

**AND SOLUTION IN**

**ASR APPLICATIONS**

**CHAPTER-5**

PRACTICAL PROBLEMS FACED AND

SOLUTION IN ASR APPLICATIONS

**5.1 Practical issues**

Although signal processing technologies show promise in leading to robust systems, the fundamental method for improving robustness is to understand the reason of the mismatch between the training data used in system development and testing data gathered in real environment. Fig. 5.1 illustrates the main causes of acoustic variations resulting from the speech production processes[25]. Generally speaking, there are two main sources of this mismatch:

1. Additive noise, such as fan running, other speaker's speech, car engines, etc.

2. Channel distortion, such as reverberation, telephone line, microphones, etc.

According to the different sources of the mismatch, robustness techniques may be split into three categories:

1. **Speech enhancement** directly deals with these problems on speech data. It tries to make corrupted speech data much cleaner and matched to the acoustic condition of the clean speech data. Spectral subtraction (SS) is a typical example.

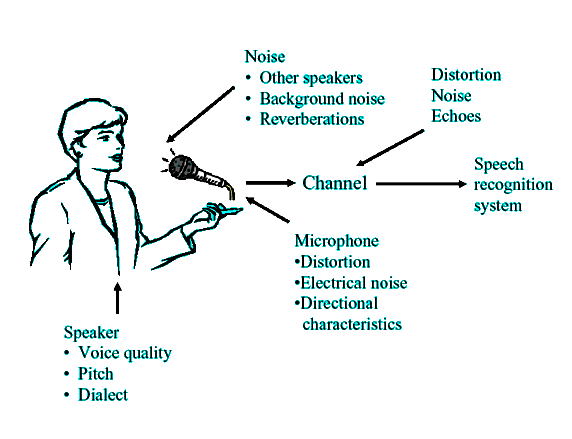


Figure 5.1 : Causes of acoustic variation in speech

2. **Robust speech feature extraction techniques** are used to extract feature more immune to additive noise or channel distortion. It includes Cepstral Mean Subtraction (CMS), Relative Spectral (RASTA), etc.

3. **Model-based compensation approaches** try to adapt the pre-trained recognition models such that they are more matched to the real acoustic environment. The famous examples are Vector Taylor Series (VTS), Parallel Model Combination (PMC), etc.

However, there is another problem in practical implementation of ASR algorithms. In theoretical analysis, a speech recognizer is assumed that the start and end of an utterance is known. It starts the recognition search from the beginning frame and ends the search at the end frame. However, in a real application, utterance segmentation is unknown. One should find a way to detect the beginning and end of an utterance. Some systems use a push-to-talk button to let the users decide the utterance segmentation; therefore utterance detection algorithm is not needed. This model sometimes requires you to push and hold while talking. You push to indicate when the speech begins and then release when the end of speech. The disadvantage is the necessity to activate the speech processing manually each time one speaks. To make the system more flexible, we use an automatic utterance detection algorithm, called end-point detection (EPD), to decide when to start recognition, when to stop recognition in real time[26].

* 1. **Model of environment**

The main effects of environment are additive noise and channel distortion. Additive noise, such as door slams, a fan running in the background, or other speaker's speech, is common in our daily life. Channel distortion may be caused by reverberation, the presence of an electrical filter in the A/D circuitry, the response of the local loop of a telephone line, etc. Fig. 5.2 shows a widely used model of the speech signal corrupted by both additive noise and channel distortion[27].

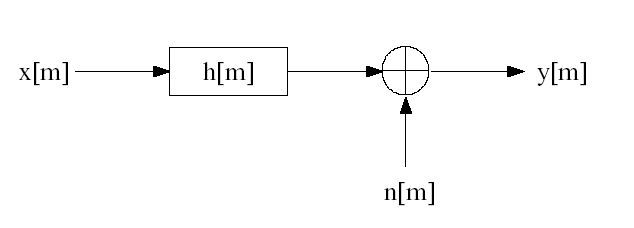


Figure 5.2 : Model of the environment

Assume *x* [*m*] is clean speech signal, *h* [*m*] is the impulse response of channel distortion, *n* [*m*] is noise signal, and *y* [*m*] is corrupted speech signal. In time domain, the relation about these values is:

Additive noise *n* [*m*] is defined as white noise. And it is assumed to be stationary and uncorrelated with the clean speech signal *x* [*m*]. In this work, we ignore the effect of channel distortion.

**5.3 End-point detection (EPD)**

When implementing the end-point detection method, it is often assumed that during several frames (10 to 200 milliseconds) at the beginning of the incoming speech signal, the speaker has not said anything as shown in Fig. 5.3. Thus, within this interval, the statistics of the background silence or noise is measured. The end-point detection is often based on the energy threshold which is a function of time. Actually, several energy-based measurements have been proposed for end-point detection[28].This is defined as the average of the squared sum of the speech samples in a frame in time domain, which is shown in Eq 5.2.

where N is the number of speech samples of one frame.

If we suppose there are *M* frames of silence or noise at the beginning of the speech utterance, the estimated noise can be achieved by smoothing the MSE derived from these *M* frames of signals. Depending on the estimated noise, we can set thresholds or noise levels to detect the boundary of speech. If the MSE of a frame is higher than the first predefined threshold, it is possibly believed as the beginning point. Then, if the MSE of a following frame is lower than the second predefined threshold, the end of the speech may be detected.

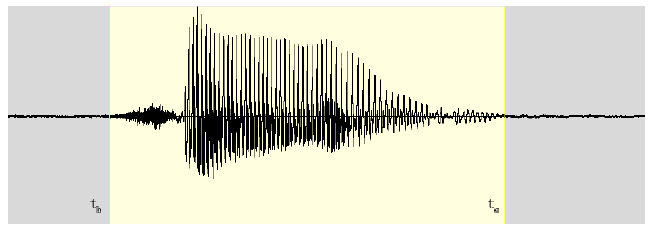
****

Figure 5.3: End-point detection with boundary *tb* and *te*

* 1. **Spectral subtraction (SS)**

In spectral subtraction, we assume that there is no channel distortion but only additive noise in speech data. Then the environment model Eq. 5.1 can be simplified as:

After Fast Fourier Transform, the noisy speech signal is transformed to frequency domain. Basically, most methods of speech enhancement have in common the assumption that the power spectrum of a signal corrupted by uncorrelated noise is equal to the sum of the signal spectrum and the noise spectrum.

However, taking this assumption as a reasonable approximation for short time analysis (20-25msec), it can lead to a simple noise subtraction method. In order to reduce the additive noise, what we need to do is to estimate |*N (f)*|*²* by the received speech data |*Y(f)*|*²*. Commonly, a method to achieve this is to use the average |*Y(f)*|*²* over *M* frames which are known to be just silence/noise (i.e., when no speech is present) as noise estimate. If the noise is stationary, this estimate will be rather accurate.

Then, an estimate |*X(f)*|*²* can be achieved by subtracting the noise estimate from the speech signal |*Y(f)*|*²*:

where frequency-dependent signal-to-noise ratio *SNR*(*f*) is defined as

Eq. 5.6 shows the magnitude of the frequency-domain speech signal but not the phase. It is not a problem since the phase information is always ignored in speech recognitions.

**CHAPTER-6**

**HMM BASED**

**ACOUSTIC MODELLING**

**CHAPTER-6**

HMM BASED ACOUSTIC MODELLING

In a speech recognition system, the incoming speech features from the front-end part are modelled by hidden Markov models (HMM). Since people traditionally suppose the speech signal to be short-time stationary and the speech features carry the information of the speech acoustic properties, the features should obey some kind of distribution. So HMMs, which have long dominated the world of acoustic modeling, are used to characterize the speech signal as a parametric stochastic process and statistically represent the variation of a speech unit (a word/phoneme/syllable).

**6.1 HMMs for ASR**

Hidden Markov modelling is a *stochastic* technique, which means that it models variation (the variation in the signal) by using probabilities.Usually, each sound (sometimes: word or word sequence) is represented by a hidden Markov model (HMM). Whole utterances are then modelled as a sequence of words, each of which constitutes a sequence of sounds. The sequence of words with the highest probability is “recognised”.

The a-priori probability of the words (lexicon) and of word sequences (language) model play an important role in computing the most likely sequence of HMMs to have generated an acoustic signal.

* Markov models consist of states which are connected by transitions.
* When the automaton is in a specific state, it emits a symbol (e.g. an acoustic vector)
* Each transitions between two states has a probability associated with it.

A simple example, in which the states are represented by containers with coloured balls.

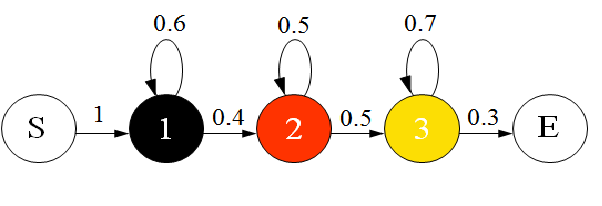


Figure 6.1 Example of Markov Model

We start in state S, which does not emit a symbol. From there, we go to state 1 with probability = 1.There we take a black ball from the container. Then we either continue to the 2nd state (p = 0.4) and take a red ball or we go to state 1 again (self-loop) and take another black ball from the container.We continue until we get to state E and have collected a sequence of coloured balls.

We can now compute the probabilty that the HMM displayed below generated the shown sequence of observations as

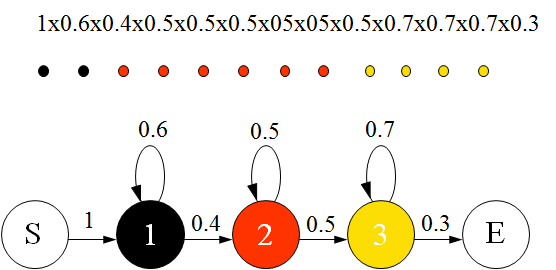


Figure 6.2 Markov Model sequence

Hidden Markov models (HMMs) differ from Markov models in that the state emissions cannot be allotted to only a particular state.In this example this would be the case, if all three (emitting) containers are filled with red, black and yellow balls.The percentage of balls of the different colours can be different for the three containers, so that the colour emissions have different probabilities for each of the three states. We start in state S, which does not emit a symbol. From there, we go to state 1 with probability = 1. There we take a ball from the container, which can now be red, black or yellow.

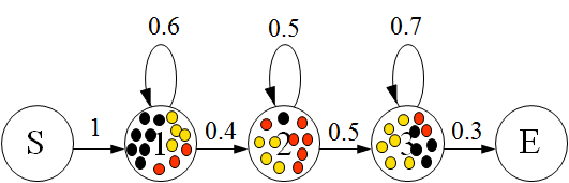
****

Figure 6.3 Example of Hidden Markov model

Then we go on to the 2nd state (p = 0.4) and take a ball from the container or we go to state 1 again and take another ball from that container.We continue until we get to state E and have collected a sequence of coloured balls.

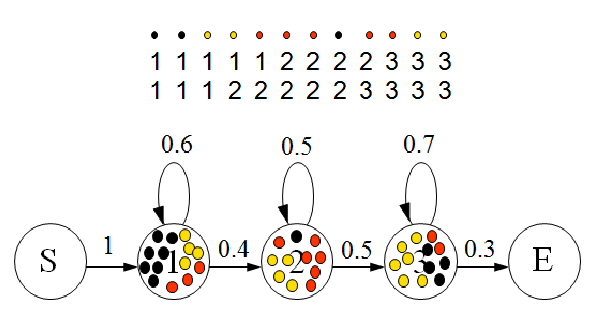


Figure 6.4 Hidden Markov model sequence

In the situation that we can see a sequence of coloured balls it is now impossible to *recognise* with certainty in which states (from which container) each ball has been taken. The states are “hidden”, that is why we speak of *hidden* Markov modelling.

**6.2 HMM for the digit recognition task**

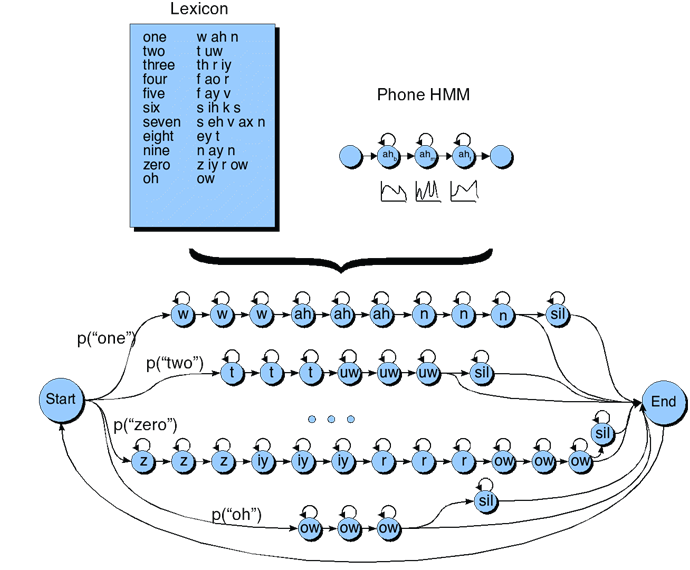


Figure 6.5 : HMM for digit recognition task.

**6.3 Viterbi Search Engine**

The algorithm for finding the best state sequence (Viterbi, 1967; Forney, 1973) is known as the Viterbi algorithm. It is an application of dynamic programming for finding a best scoring path in a directed graph with weighted arcs. This trellis framework is shown in Fig. 6.6.

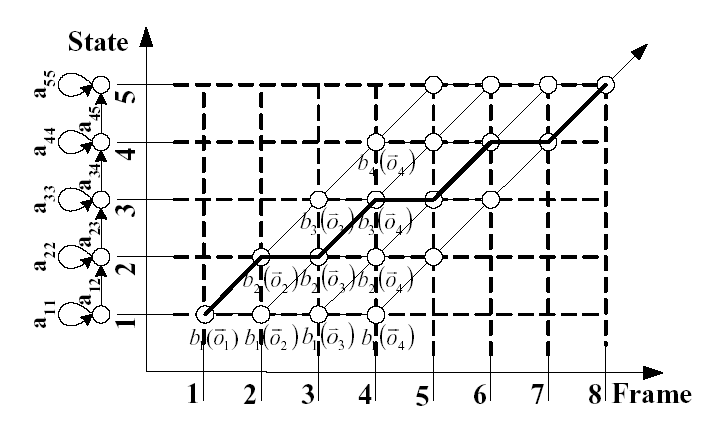


Figure 6.6 : Viterbi trellis computation for the HMM

Generally speaking, the Viterbi algorithm for isolated word recognition can be separated into three major parts:

* Initialization
* Recursion
* Termination
  1. **Viterbi trellis for “five”**

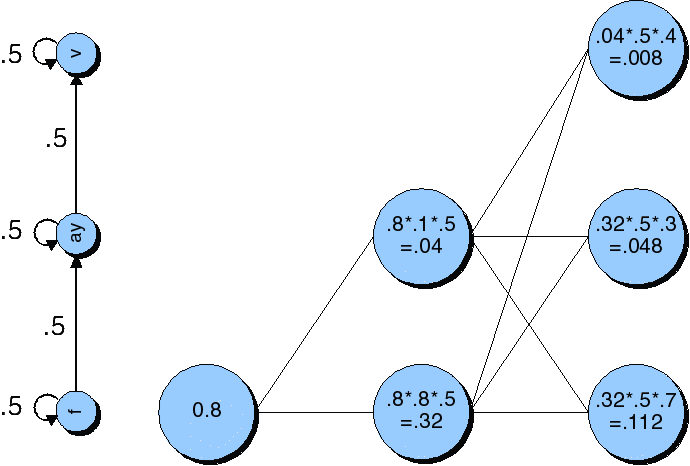


Figure 6.7 : Viterbi trellis for “five”.

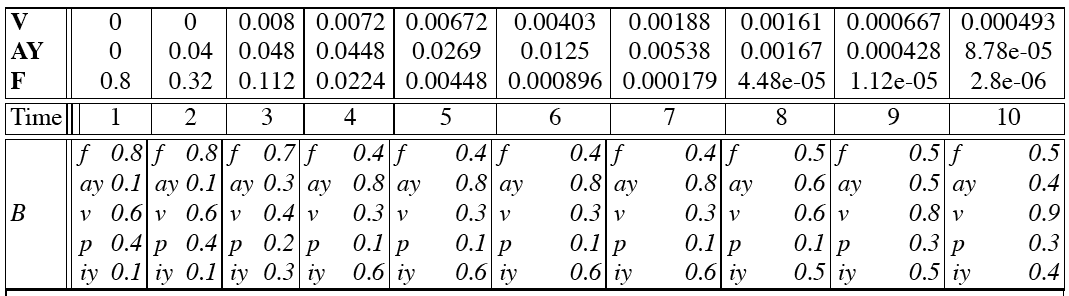


Table 6.1 : Viterbi trellis computation for “five”.

**6.5 Isolated word recognition (IWR) and Connected word recognition (CWR)**

In this work, we mainly focus on the isolated word recognition algorithms which are computationally saving.

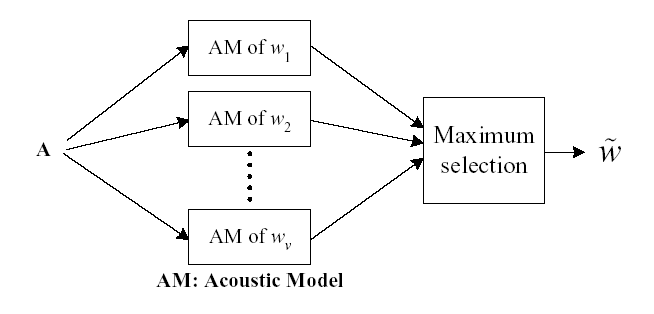
6.5.1 Isolated word recognition

Isolated word recognition, by its name, is a task within which only a single word is recognized. Each word forms a unique model which can be a digit, a letter or a string of syllables. These words or strings are all modelled in the vocabulary. The vocabulary size is always finite, and there are only a few models can be recognized.

Let *V* = {w1,w2, …,wv}be a set of *v* words to be recognized. If *A* is an input utterance to be recognized and all words are equally probable to be spoken, then the recognized word is



The process of an isolated word recognition system is shown in Fig. 6.8.

Figure 6.8: The Process of isolated word recognition

6.5.2 Connected word recognition (CWR)

Connected word recognition is an extension of isolated word recognition. For connected word recognition, the exit state of one HMM is connected to the entry state of another HMM. In this case, path extension can be within or across models. The recognition result is given by the best model sequence W= {w1,w2, …,wn} that has the highest accumulated probability at the last frame. The search network of connected word recognition is shown in Fig. 6.9.

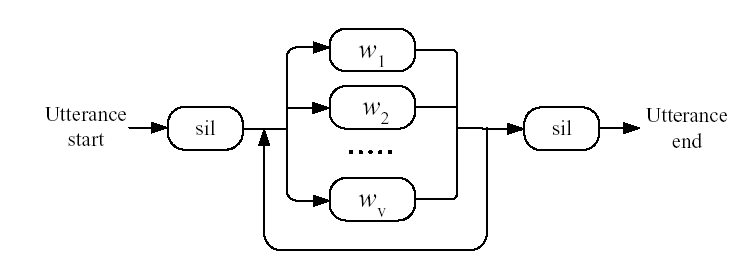


Figure 6.9 : Search network of connected word recognition

**CHAPTER-7**

**REMOVAL OF WIDEBAND NOISE**

**FROM SPEECH SIGNAL AND RESULTS FOR SPEECH RECOGNITION**

**CHAPTER-7**

REMOVAL OF WIDEBAND NOISE FROM SPEECH

SIGNAL AND RESULT FOR SPEECH RECOGNITION

During the past decades there has been an increasing interest in the cancellation of noise from audio signals, mainly due to the improved performance of the digital equipment used for recording, storage and reproduction of sound. As a consequence, the enhancement of old and poorly recorded audio material is often necessary, especially prior to the transfer of such signals to higher resolution digital storage media. Audio noise cancellation has been an active area during the past decades. Although, much effort was related to the speech signals many newer techniques are directed towards the broadband audio signals. However, in many cases, methods developed primarily for speech signals have also found application in the restoration of audio signals. Although some commercial products are now available in this field, this area is still open to techniques which can enhance severely degraded signals without the introduction of additional processing distortions.

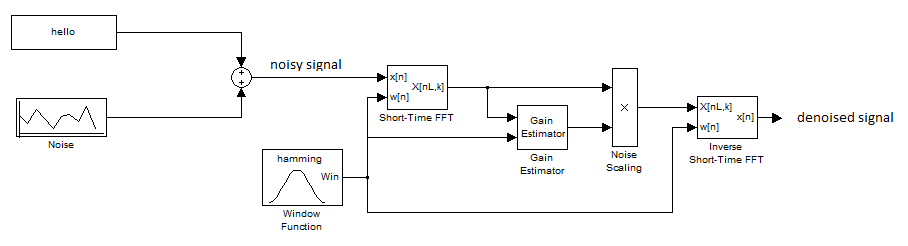


Figure 7.1 Removal of noise.

**7.1 Results for Noise Removal**

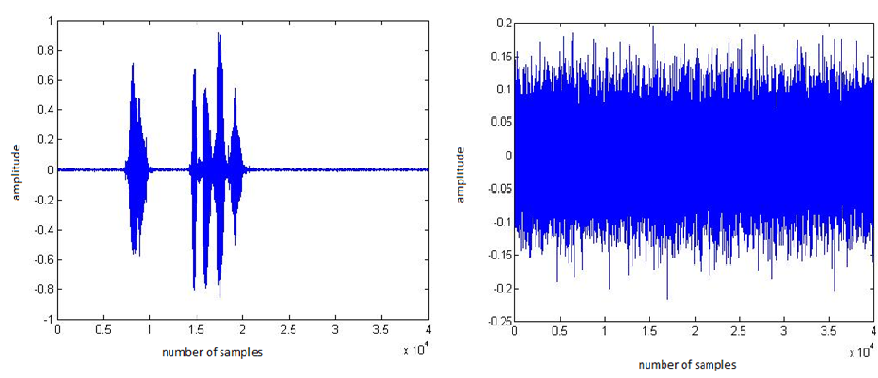


Figure 7.2 Speech signal “hello, this is matlab” Figure 7.3 Noise

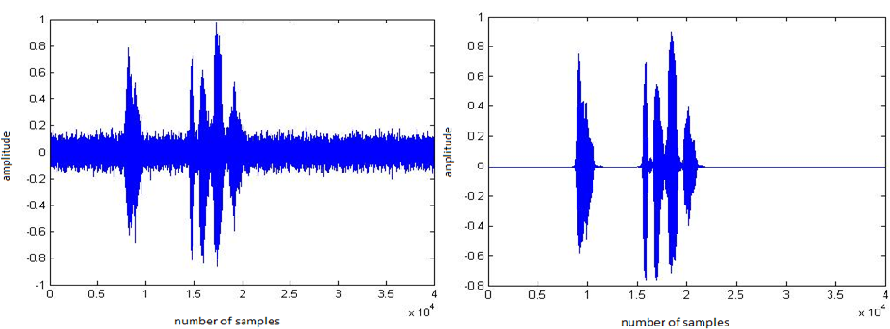


Figure 7.4 Noisy signal Figure 7.5 Denoised signal

**7.2 Results for Speech Recognition**

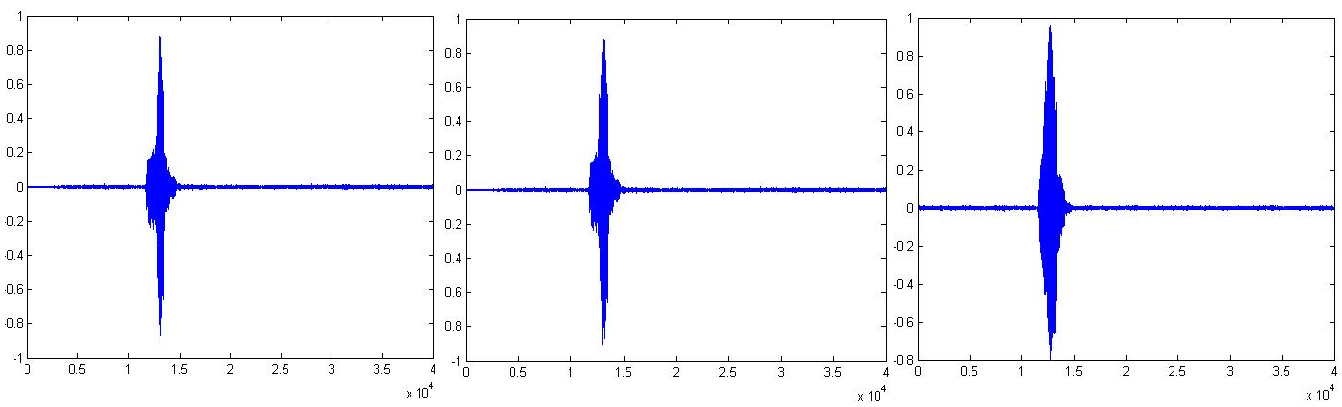


Figure 7.6 Three different utterance of word “one” after digitization.

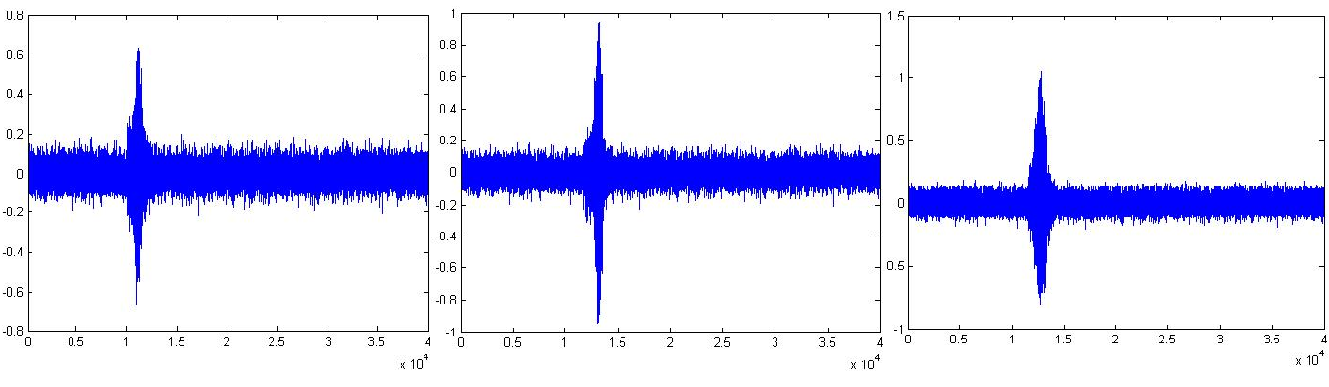


Figure 7.7 Three different utterance of word “one” with noise.

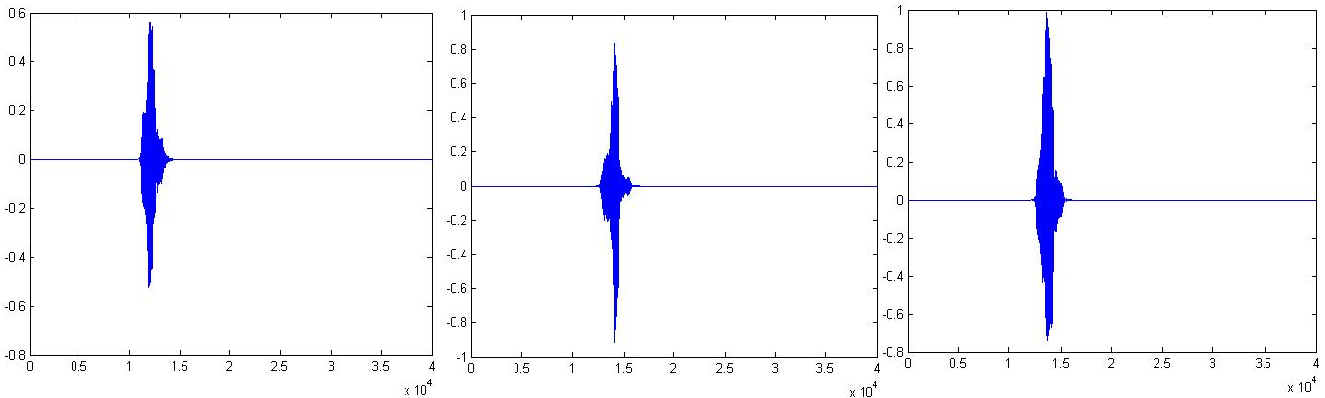
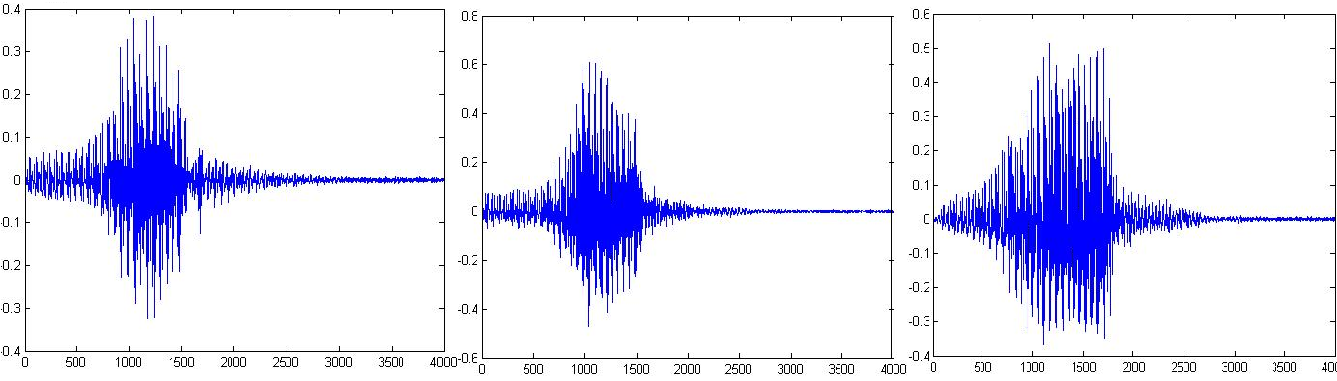
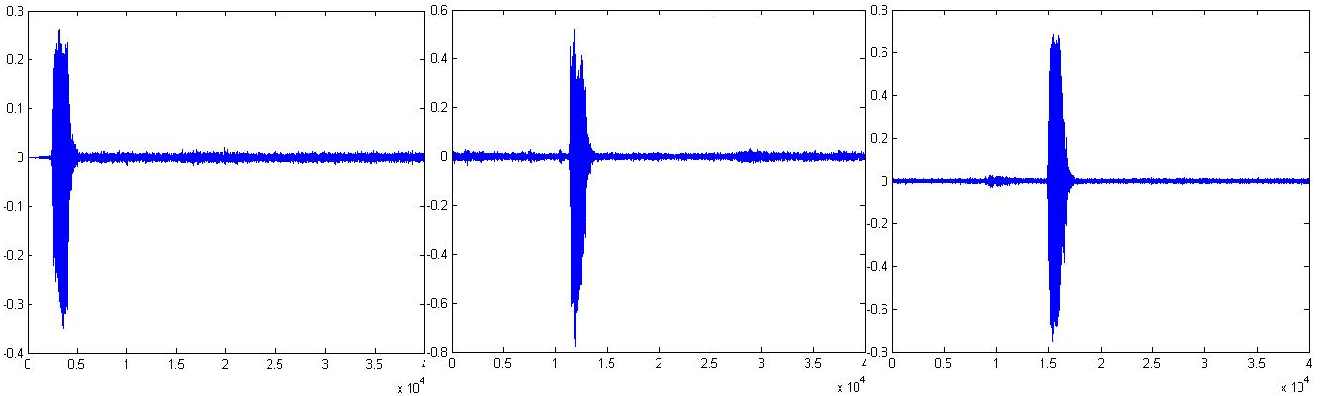


Figure 7.8 Three different utterance of word “one” after removal of noise.

Figure 7.9 Three different utterance of word “one” after framing.

 Figure 7.10 Three different utterance of word “two” after digitization.

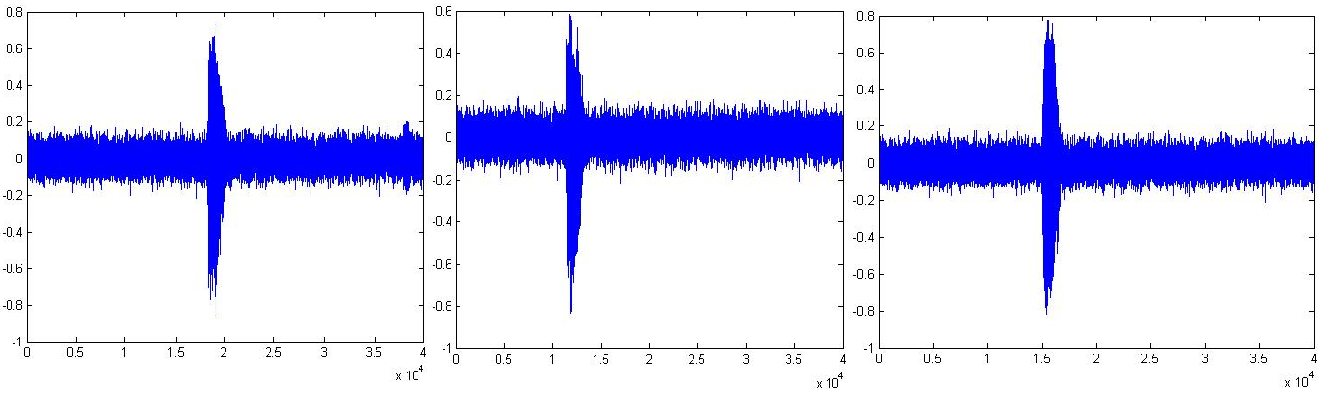


Figure 7.11 Three different utterance of word “two” with noise.

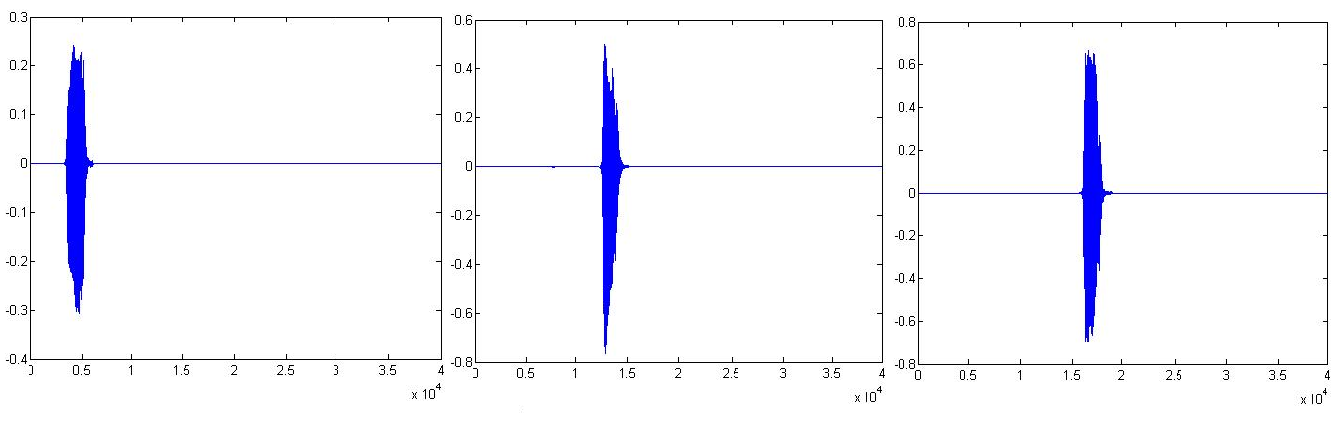


Figure 7.12 Three different utterance of word “two” after removal of noise.

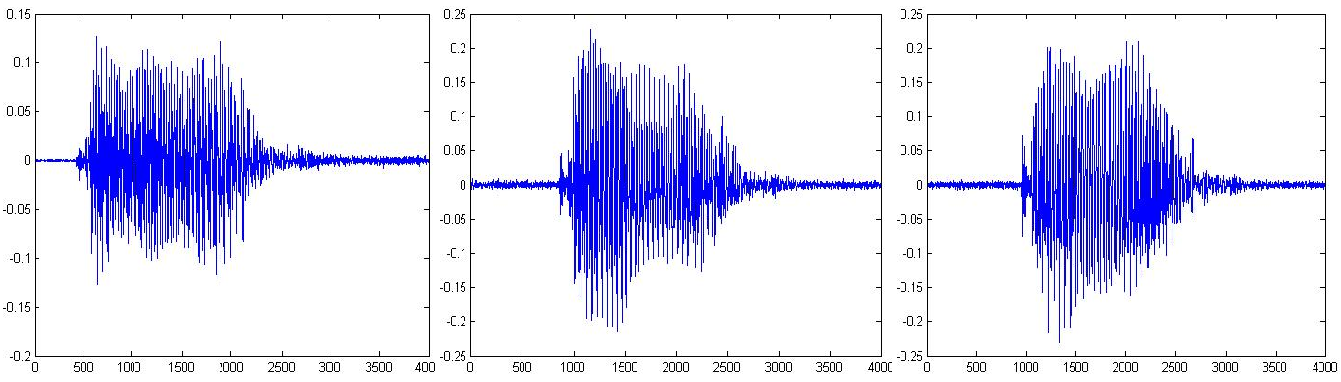


Figure 7.13 Three different utterance of word “two” after framing.

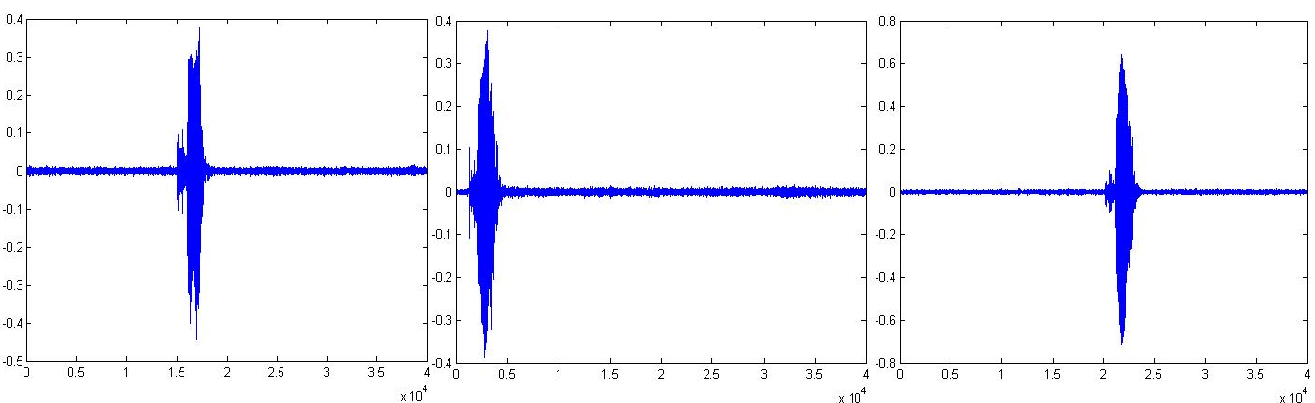


Figure 7.14 Three different utterance of word “three” after digitization.

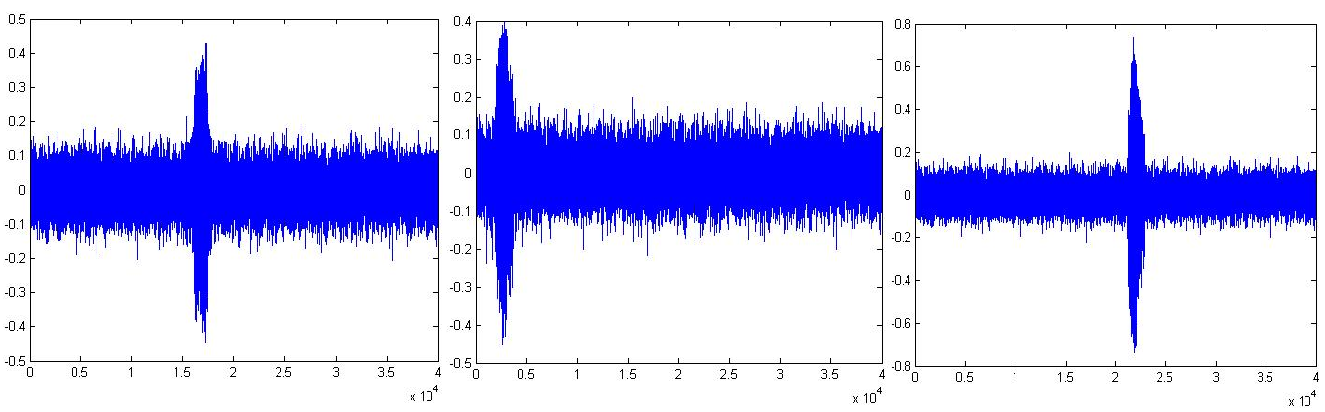


Figure 7.15 Three different utterance of word “three” with noise.

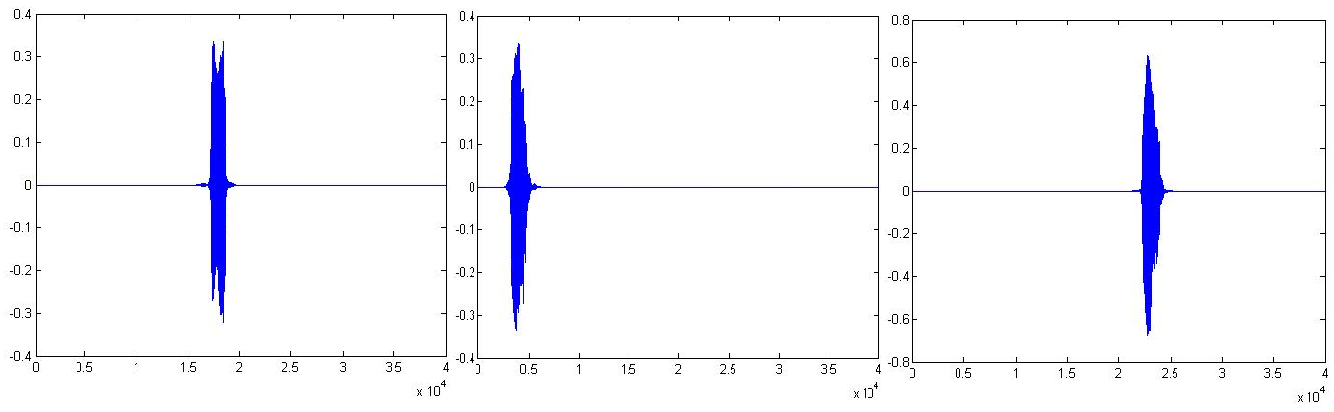


Figure 7.16 Three different utterance of word “three” after removal of noise.

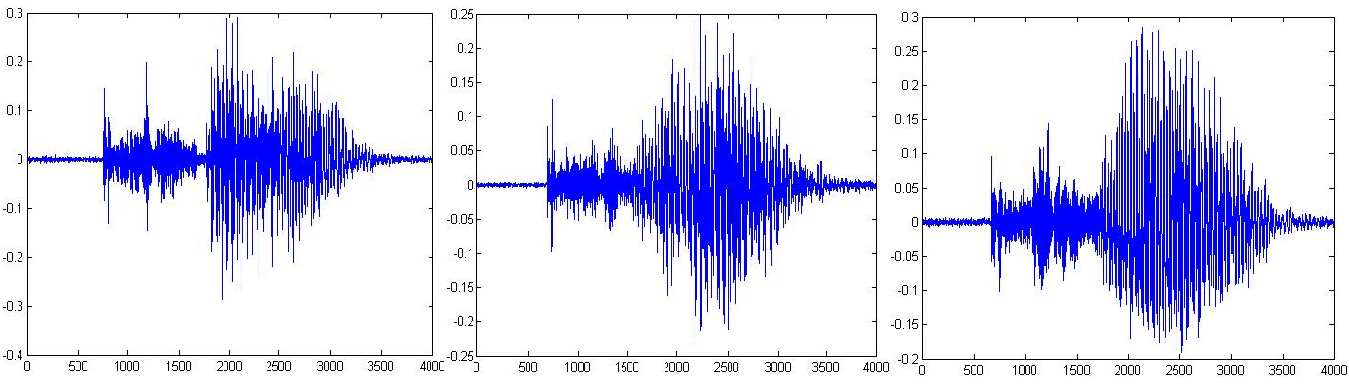


Figure 7.17 Three different utterance of word “three” after framing.

**CHAPTER-8**

**CONCLUSION AND**

**FUTURE PROSPECTS**

**CHAPTER-8**

CONCLUSION AND FUTURE PROSPECTS

**8.1 Conclusion**

This method is supported with a behavior model for noise cancellation in audio devices such ashearing aid. Equivalent behavioural models are generated for carrying out simulation in MATLAB.The simulation results show the enhancement in the noise cancellation usingthe proposed scheme for noise cancellation. Speech recognition technology has many applications on embedded systems, such as stand-alone devices and single-purpose command and control systems.

The development of such applications is not straightforward. The highly complex ASR algorithms have to be optimized to meet the limitations in computing power and memory resources. The optimization, which typically involves simplification and approximation, inevitably leads to the loss of precision and the degradation of recognition accuracy.

**8.2 Suggestions for future research**

There is potential work to be done in the future:

1. To make the system more robust to the noisy environment, especially for low SNR environment, some new modifications to the spectral subtraction method may be researched for low-cost applications.

2. As we can see, the end-point detection used in this work is only based on the frame energy which is not good for a noisy environment with low SNR. The error rate of determining the beginning and ending of speech segments will greatly increase which directly influence the recognition performance at the pattern recognition part. So, we should try to use some effective way to do end-point detection.

3. Although we have proposed many optimization methods to the complex speech recognition technology, there may be some other optimizations that we can apply. For example, when the vocabulary size increases, we can use pruning technique to reduce the computation time for path extension.

**REFERENCES**

[1] B. H. Juang and Tsuhan Chen: “The past, present and future of speech processing”, Signal Processing Magazine, IEEE, volume: 15, issue: 3, pages: 24-48, May 1998.

[2] K. Davis, R. Biddulph and S. Balashek: “Automatic recognition of spoken digits”, J. Acoust. Soc. Am., volume: 24, pages: 637-642, 1952.

[3] J. W. Cooley and J. W. Tukey: “An algorithm for the machine computation of complex Fourier series”, Math. Comput., volume: 19, pages: 297-301, 1965.

[4] B. Bogert, M. Healy and J. Tukey: “The frequency analysis of time series for echos”, Proc. Symp. on Time Series Analysis, pages: 209-243, Wiley, New York, USA, 1963.

[5] J. D. Markel and A. H. Gray Jr., Linear Prediction of Speech, Springer-Verlag, 1976.

[6] H. Sakoe and S. Chiba: “Dynamic programming algorithm optimization for spoken word recognition”, IEEE Transactions on Audio, Speech and Signal Processing, volume: 26, pages: 43-49, 1978.

[7] L. E. Baum and T. Petrie: “Statistical inference for probabilistic functions of finite state Markov chains”, Ann. Mathemat. Stat., volume: 37, pages: 1554-1563, 1966.

[8] L.R. Rabiner and B.H. Juang, Fundamentals of Speech Recognition, Engle- wood Cliffs Publisher, NJ, 1993.

[9] K. F. Lee, Automatic speech recognition - the development of the Sphinx system, Kluwer, 1989.

[10] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1997.

[11] Oliver Deroo: “A short introduction to speech recognition”, <http://www.babeltech.com/download/SpeechRecoIntro.pdf>.

[12] S. Yong, M. Adda-Dekker, X. Aubert, C. Dugast, J. L. Gauvain, D. J. Kershaw, L. Lamel, D. A. Leeuwen, D. Pye, A. J. Robinson, H. J. M. Steeneken and P. C.Woodland: “Multilingual large vocabulary speech recognition: the European SQALE project” , Computer Speech and Language, volume: 11, pages: 73-89, 1997.

[13] J. D. Markel and A. H. Gray Jr., Linear Prediction of Speech, Springer- Verlag Publisher, 1976.

[14] S. B. Davis and P. Mermelstein: “Comparison of parametric representations for monosyllabic word recognition in continuous spoken sentences”, IEEE Transactions on Audio, Speech and Signal Processing, volume: 28, pages: 357-366, 1980.

[15] H. Hermansky: “Perceptual linear predictive (PLP) analysis of speech”, J.Acoust. Soc. America, pages: 1738-1752, 1990.

[16] C. Myers and L. R. Rabiner: “Connected word recognition using a level building dynamic time warping algorithm”, IEEE International Conference on Acoustics, Speech, and Signal Processing, volume: 6, pages: 951-955, Apr 1981.

[17] A New Hybrid Algorithm for Speech Recognition Based on HMM Segmentation and Learning Vector Quantization Shigeru Katagiri, Member, IEEE, and Chin Hui Lee, Senior Member, IEEE.

[18] Accelerating Speech Recognition Algorithm with Synergic Hidden Markov Model and Genetic Algorithm based on Cellular Automata, Mohammad Mosleh, Dep.of Computer Eng., Islamic Azad University, Dezful,Iran.

[19] L. R. Rabiner: “A tutorial on hidden Markov models and selected applications in speech recognition”, Proceedings of the IEEE , volume: 77, issue: 2, pages: 257-286, Feb. 1989.

[20] J. C. Junqua and J. P. Haton, Robustness in Automatic Speech Recognition, Kluwer Academic Publishers, 1996.

[21] B. Delaney, N. Jayant, M. Hans, T. Simunic and A. Acquaviva: “A low-power, fixed-point, front-end feature extraction for a distributed speech recognition system”, IEEE Transactions on Audio, Speech and Signal Processing, volume: 1, pages: 793-796, May 2002.

[22] K. K. Shin, J. C. H. Poon and N. C. Li: “A fixed-point DSP based Cantonese recognition system”, ISIE ’95, volume: 1, pages: 390-393, Jul. 1995.

[23] Xuedong Huang, Alex Acero and Hsiao-Wuen Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Carnegie Mellon University, pages: 230-231, April 2001.

[24] S. Furui: “Speaker independent isolated word recognition using dynamic features of speech spectrum”, IEEE Transactions on Audio, Speech and Signal Processing, volume: 34, pages: 52-59, 1986.

[25] Sadaoki Furui: “Recent advances in robust speech recognition”, ESCANATO Workshop on Robust Speech Recognition for Unknown Communication Channels, page: 11-20, April 1997.

[26] Guanghui Hui, Kwok-Chiang Ho and Zenton Goh: “A robust speaker-independent speech recognizer on ADSP2181 fixed-point DSP", International Conference on Signal Processing Proceedings, pages: 694 - 697, 12-16 Oct. 1998.

[27] Xuedong Huang, Alex Acero and Hsiao-Wuen Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Carnegie Mellon University, page: 473-482, April 2001.

[28] G. S. Ying, C. D. Michell and L. H. Jamieson: “Endpoint detection of isolated utterances based on a modified Teager energy measurement”, IEEE International Conference on Acoustics, Speech, and Signal Processing, pages: 732-735, 1993.