8th ICCIT 2005
Islamic University of Technology (IUT), 28-30 December 2005

# Development of a Bangla Speech Driven Application

**Kashif Nizam Khan, Md. Zahidul Islam, Jinat Rehana and Md. Saidur Rahman**
Computer Science and Engineering Discipline
Khulna University, Khulna-9208, Bangladesh
kashif570@yahoo.com, zahidbabubd@yahoo.com, jinat0230@yahoo.com, msrku@yahoo.com

## Abstract

*This paper work is concerned with the development of a speech driven application using automatic segmentation and recognition system for continuous speech. This system will read voice input from the microphone port of the sound card and then by processing the voice input the system itself will communicate with the calling application and make it execute. In this approach firstly a front-end processing is performed by the system for automatic segmentation of continuous speech into voiced segments. These segments are further used for recognition. The Mel Frequency Cepstrum Coefficents(MFCCs) are extracted by the system for recognition. The Euclidean distance measurement technique is employed to calculate the distance of the feature vectors of an unknown segment with the stored ones. A simple drawing application is developed to interface it with Bangla voice commands. The system is tested with speakers of different age and sex and a satisfactory accuracy of approximately 75% with a maximum of 90% has been achieved.*

**Keywords**: Critical band filter, Fourier Transform (FFT), Hamming Window, Mel frequency cepstrum coefficient (MFCC), Pre-emphasis.

## I. INTRODUCTION

Spoken language is the most efficient media of communication, automatic speech driven application is probably the most important step towards natural human-machine interaction. Technology has developed immensely in the field of ubiquitous computing. It appears that most computer users can create and edit documents and interact with their computer more quickly with conventional input devices, a keyboard and mouse, despite the fact that most people are able to speak considerably faster than they can type. Even people who are physically disabled can interact via speech. A number of significant research efforts in the field of automatic speech understanding and recognition resulted in some remarkable achievements during the last few decades. Besides English language, there are a lot of research experiments and achieved results in various languages throughout the world. But satisfactory advancement in speech research in Bangla Language is not achieved yet. In fact, we are only in the initial level in this field.

Obstacles to robust recognition include acoustical degradations produced by additive noise, the effects of linear filtering, nonlinearities in transduction or transmission, as well as impulsive interfering sources, and diminished accuracy caused by changes in articulation produced by the presence of high-intensity noise sources. Speaker-to-speaker and machine-to-machine differences impose a different type of variability, producing variations in speech rate, co-articulation, context, and dialect. As speech recognition and spoken language technologies are being transferred to real applications, the need for greater robustness in recognition technology is becoming increasingly apparent.

This research is an important step to give full stops to conventional input devices to give commands to the applications. In this speech driven application, recognition of bangle voice commands is done using the speech recognition tools. As it is very preliminary stage of the research, a system is developed that assists monosyllabic commands.

## II. KEY CONCEPTS

Speech is input via microphone and its analog waveform is digitized. The recognition system extracts necessary features from the waveform needed to identify the correct decision [2]. The recognition system typically consists of two phases:

- Processing
- Recognition

In the processing phase, data are fed to the system; the system forms a reference pattern or template for each command. In the recognition phase, the system computes the features of pattern for unknown command and identifies it as the command whose reference pattern matches these features most closely.

The design of this system generally centers around three problems:

- Segmentation
- Feature extraction
- Command matching

### A. Segmentation of Continuous Speech

Segmentation refers to the separation of different regions of continuous speech for further processing. Usually segmentation is done by detecting the proper end points of the speech events and then separated into different pieces containing the audio signals on the basis of the detected endpoints.

ISBN 984-32-2873-1 © ICCIT 2005

### A.1 Events in Continuous Speech

Mainly there are three phases of continuous speech signal

1. **Silent zone (S)**, where no speech is produced.
2. **Unvoiced zone (U)**, in which the vocal cord is vibrating so the resulting speech waveform is a periodic or random in nature.
3. **Voiced zone (V)**, where the vocal cords are tensed and therefore vibrate periodically.

### B. Feature Extraction

The first step towards any recognition system is to extract features suitable for recognition. There are several alternative candidates those can be used as feature for speech. Feature extraction and Feature selection are the two commonly used preprocessing techniques. Feature extraction means new features are generated from the raw data by applying one or more transformations [9]. A schematic overview of feature extraction is shown in fig 2.1.

Raw Data
Extraction
Feature Vector

Fig 2.1 Generating a feature vector from an input data set [9]

The selection of feature is one of the most important factors in designing a speech recognition system. From the study of different previous research works it was observed that among the different features the MFCC results in best recognition rate [10].

### B.1 Pre emphasis

Pre emphasis is used compensates for the negative spectral slope of the voiced portions of the speech signal. A typical signal pre emphasis is given by

$$y(n) = s(n) - C \times s(n-1) \qquad (1)$$

where *C*, the pre emphasis constant generally falls between 0.9 and 1.0 [1]. In this research we have used the value 0.98.

### B.2 Windowing

Windowing of speech signal involves multiplying a speech signal by a finite-duration window. The type of window chosen influences the time and frequency resolution. One of the most popular windows used in speech recognition is the Hamming window defined by the equation:

$$h(n) = 0.54 - 0.46\cos\left(2\pi n/_{N-1}\right)\ldots\ldots(0 \le n \le N-1)$$

$$=0,\ldots\ldots\text{otherwise} \qquad (2)$$

where, *N* is the window length [1].

### B.3 Fourier Transform

Fourier analysis is the generic name for a group of mathematical techniques that are used for decomposing signals into sine and cosine waves. The information is encoded in the frequency, phase and amplitude of the spectral components that make up the signal. Applying the Fourier transform to a signal converts it from its time domain representation into the frequency domain representation.

### B.3.1 Discrete Fourier Transform

Discrete Fourier Transform (DFT) computes the frequency information of the equivalent time domain signal. The Short time Fourier analysis of windowed speech signal can produce a reasonable feature space for recognition. The Fourier Transform for a discreet time signal $f(kT)$ is given by

$$F(n)=\sum_{k=0}^{N-1} f(kT)e^{-j2\pi nk/N} \qquad (3)$$

which can be written as

$$F(n)=\sum_{k=0}^{N-1} f(k)W_N^{-nk} \qquad (4)$$

where $f(k)=f(kT)$ and $W_N=e^{j2\pi/N}$. $W_N$ is usually referred to as the *kernel* [11] of the transform.

### B.3.2 Fast Fourier Transform

The direct computation of the DFT involves *N* complex multiplication and N-1 complex additions for each value of *F(n)*. Since there are *N* values to be determined, then $N^2$ complex multiplication and *N(N-1)* complex additions will be performed. There are several algorithms that can considerably reduce the number of computations in a DFT. DFT implemented using such schemes is referred to as Fast Fourier Transform (FFT). The FFT is computed using Split-Radix FFT (SRFFT) decimation-in-frequency algorithm as it reduces the number of computations to $N/2\log_2 N$ complex multiplication and $N\log_2 N$ complex addition.

### B.3.2.1 Split Radix Fourier Transform

The split-radix FFT (SRFFT) algorithm exploits the idea of computing the DFT of even and odd numbered points independently using both radix-2 and radix-4 decomposition in the same FFT algorithm. First, in the radix-2 decimation-in-frequency FFT algorithm, the even-numbered samples of the *N*-point DFT are given as

$$X(2k)=\sum_{n=0}^{N/2-1}\left[x(n)+x(n+\tfrac{N}{2})\right]W_{N/2}^{nk}$$

$$k=0,1,\ldots,\tfrac{N}{2}-1 \qquad (5)$$

If we use a radix-4 decimation-in-frequency FFT algorithm for the odd-numbered samples of the *N*-point DFT, we obtain the following *N*/4-point DFTs

$$X(4k+1)=\sum_{n=0}^{N/4-1}\left\{\left[x(n)-x(n+N/2)\right]-j\left[x(n+N/4)-x(n+3N/4)\right]\right\}W_N^{n}W_{N/4}^{kn} \qquad (6)$$

$$X(4k+3)=\sum_{n=0}^{N/4-1}\left\{\left[x(n)-x(n+N/2)\right]+j\left[x(n+N/4)-x(n+3N/4)\right]\right\}W_N^{3n}W_{N/4}^{kn} \qquad (7)$$

### B.4 Critical Band Filters [1]

It is important that human can detect a variety of information from sound sources including components of different frequencies. This is done by frequency selection by masking. Frequency components in a sound are detected by a series of overlapping band-pass filter centered continuously at the frequencies throughout the normal range of hearing. The ability to discriminate between two simultaneously presented sounds which contain frequencies that are very close is limited to the width of one of these auditory band-pass filters, the critical band. An expression for critical bandwidth is

$$BW_{critical} = 25 + 75\left[1 + 1.4\left(f/1000\right)^2\right]^{0.69} \qquad (8)$$

This transformation can be used to compute bandwidths on a perceptual scale for filters at a given frequency on *Bark* or *mel* scale. A critical band filter bank is simply a bank of linear phase Finite Impulse Response (FIR) band-pass filters that are arranged linearly along the *Bark* or *mel* scale.

### B.5 Mel Frequency Cepstrum Co-efficient

Human perception of the frequency content of sounds does not follow a linear scale but uses a logarithmic distribution. Mel-frequency cepstrum coefficients (MFCCs) are based on the spectral information of a sound, but are modeled to capture the perceptually relevant parts of the auditory spectrum [1]. To obtain the Mel Frequency Cepstrum coefficients (MFCCs) at first a Fourier transformation of short speech segments into the frequency domain is performed. Then a computation of the logarithm of the amplitude spectrum forces the signal to be minimum phase, and finally an inverse Fourier transform results the signal back to the time domain and MFCCs are obtained. A set of critical band filters evenly spaced along the Mel scale smoothes and averages the signal into a smaller number of coefficients. The Computation Steps for MFCC can be summarized as shown in Fig.2.2 [1].

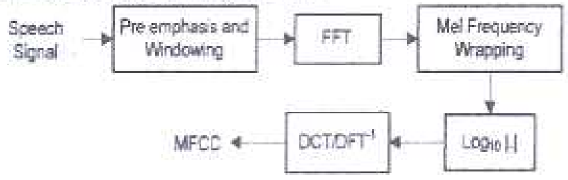

Fig 2.2 The computation steps for converting speech signal into a set of MFCC features

### C. Pattern Matching

Pattern matching is the technique in which the unknown test pattern is compared with each of the reference patterns and measure of similarity (distance) between the test pattern and each reference pattern is computed. The process finds the best match between the test pattern and the reference patterns. Several distance measurement techniques are used in pattern comparison. One of the most common methods used in distance measurement is the Euclidean distance. The Euclidean distance that is defined by:

$$d(X,Y)_{euc} = \sqrt{\left(\sum_{i=1}^{N}(X_i - Y_i)^2\right)} \qquad (9)$$

where, $N$ is the dimensionality of the vector.

## III. SOFTWARE SYSTEM IMPLEMENTATION

From earlier discussions it is clear that speech signal varies for each sound due to various reasons, it should be analyzed on short windowed segments. The following figure shows the sequence of computational steps for segmentation and feature extraction. Each segment needs to be grouped into a set of samples called a frame which typically represents 16 msec (128 samples) of speech. Then, preprocessing of the signal includes the operations pre emphasis and windowing.

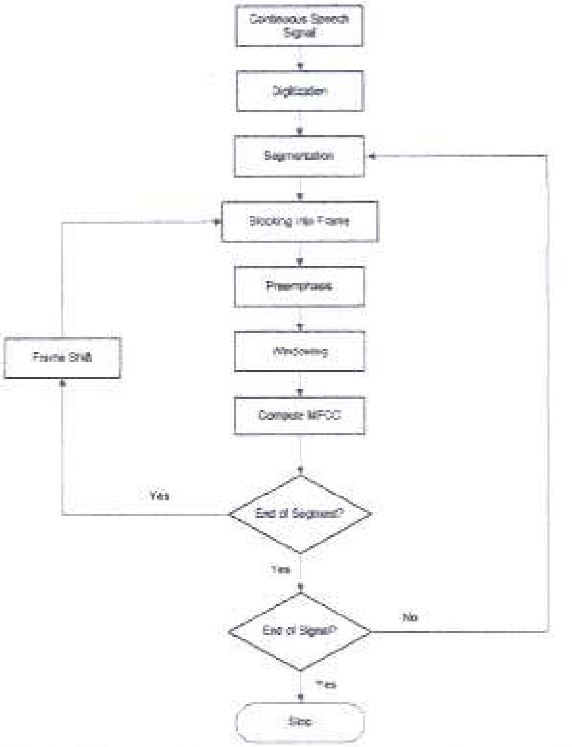

Fig. 3.1 The operation sequence to convert continuous speech into a set of features suitable for recognition.

As mentioned earlier we have used MFCC as feature to recognize command, when the feature vector is extracted from the continuous speech signal, the Euclidian distance of the feature vectors from the reference patterns is computed. Then the command with the minimum distance is recognized and executed. The overall system can be depicted as the fig 3.2.

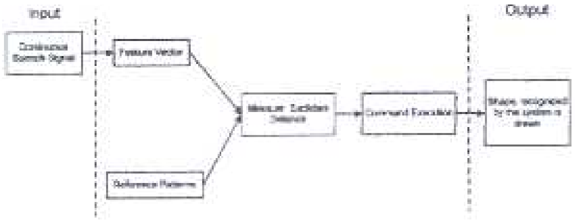

Fig. 3.2 Overall system at a glance.

### A. The Segmentation Algorithm

The algorithm divides an input speech signal into a number of segments. Here a threshold value has been used to separate the voiced portion of the speech signal from the continuous speech. This portion contains the most significant speech features. On the basis of observation threshold has been set to 10 by taking the average of consecutive speech samples. Sound data is recorded from the sound buffer directly and stored in a intermediate buffer. Then it is tested whether it is a silence area or not, by comparing it with the threshold value. The algorithm simply checks whether the input signal contains any silence or unvoiced area or not and omits that if any. The voiced portion of the signal is then stored in another buffer. This algorithm has been used by the other programs for speech segmentation. The algorithm is as follows

```
Start with the input buffer containing continuous
speech data
Set count ←buffer length. threshold←10
for i← 0 to count with increment 1 do
    data←integer value of sound data
end
for i←0 to count with increment 1 do
    set . flag←0
    while Data[i]>threshold or Data[i]< -threshold
        set flag←1
        increment i by 1
        increment k by 1
    end
end
Create a random access file & open it in append
mode
if k>=300   /* k=300 indicates that the sound data
                is a valid command */
    then call mfcc with the segmented data
        for i←0 to 256 with increment 1 do
                Write str as bytes in the file opened
                before
        end
Close the file      /* the file now contains the mfcc
                        data of the command */
end
```

### B. The Command Matching Algorithm

This algorithm compares the input sound command with the ones already stored and takes decision which task to perform. The reference filenames containing MFCC data are stored in an array. The Euclidian distance of the input command from each of the reference commands is measured and the one with less distance is the desired command. The algorithm is given below

```
st← {"command1.txt"," command2.txt "," com
        mand3.txt ",...}   /* st is the string array
                              containing reference file name*/
for j←0 to number of reference files with incre-
            ment 1 do
    set sum ←0
    for i←0 to 256 with increment 1 do
        sum ←sum+((data_command[i]-
        data_reference[i])* (data_command[i]
                                    -data_reference[i]))
    end
    sum1[j]←Square root of sum
    set min ←30000
    for i←0 to number of reference files with
            increment 1 do
        if sum1[i]<min
            then set temp ←i
                min ←sum1[i]
        end
    end
    switch temp
        case 1: Execute command 1
        case 2: Execute command 2
        case 3: Execute command 3
                ...
                ...
    end
end
```

### C. Reference Samples Database

The reference database contains the reference patterns in which there is a single pattern for each distinct command. The reference for each pattern is the feature vector that best represent the pattern. The reference pattern has been selected with the observation of the best recognition performance with different reference pattern for each distinct command. As MFCC has been selected as the feature for recognition, the reference database usually contains MFCC features for each reference pattern. The reference pattern is chosen out of several trials, which gives the best match, for each command.

## IV. GRAPHICAL ANALYSIS OF SPEECH FEATURES

The graphical analysis of the MFCC data of various reference and input commands is as follows (For simplicity we are showing the graph of first 100 data). The graphs show various reference patterns and their matching with input commands.

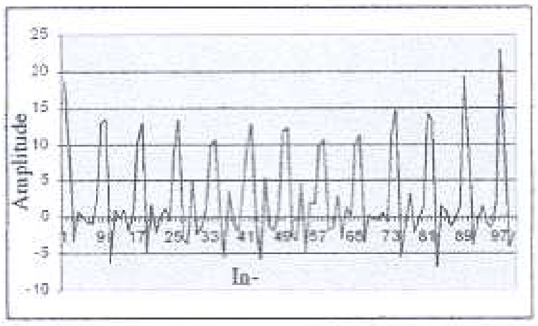

Fig 4.1 Reference " [illegible] "

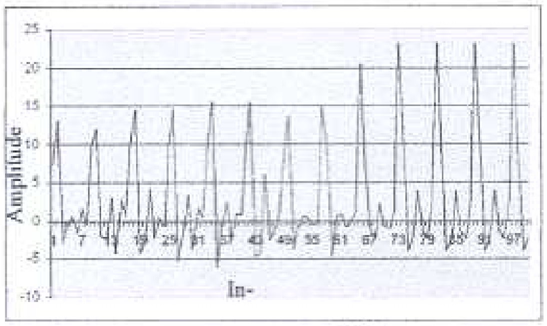

Fig 4.3 Reference " [illegible] "

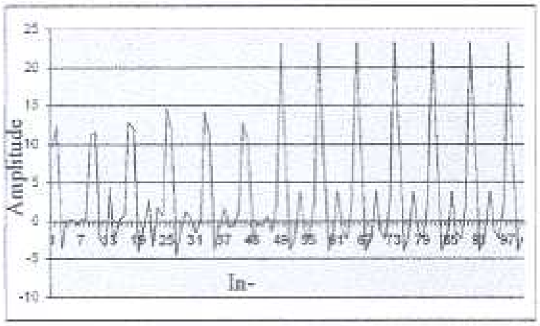

Fig 4.5 Reference " [illegible] "

## V. RESULT ANALYSIS AND DISCUSSION

As mentioned earlier the system is developed interfacing a simple drawing application which can draw circles, lines & rectangles. The tabular form of the results is shown in table I. The experiment is performed with three different speakers of different ages and sex. The result demonstrates that in this approach an accuracy of 74.44% is achieved with a maximum of 90%. The system did not employ any knowledge (syntactic or semantic) of linguistics. Inclusion of such knowledge will increase the recognition performance.

The input command is " [illegible] " and graphical analysis shows that the best match is also with the reference pattern of " [illegible] ". This is determined in the system by measuring the Euclidian distance of the input command & various reference patterns. Then simply the execution of recognized command is done as the system calls the corresponding function associated with the command.

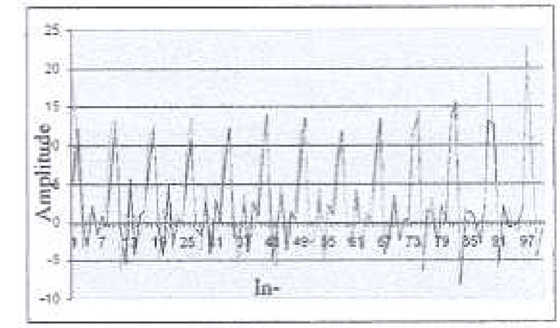

Fig 4.2 Reference " [illegible] "(▨) & Input command(■)

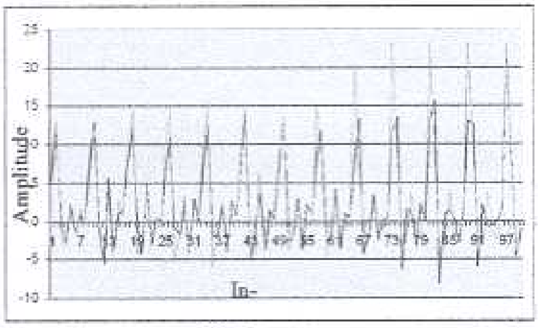

Fig 4.4 Reference " [illegible] "(▨) & Input command(■)

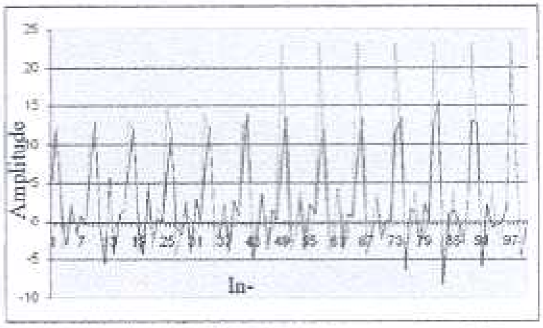

Fig 4.6 Reference " [illegible] "(▨) & Input command(■)

Table I Experimental results

| Speakers | Commands | No of Trails | No of Success | Accuracy | Avg.Accuracy |
|---|---|---|---|---|---|
| Sp. 1 | [illegible] | 10 | 8 | 80% | 80% |
| | [illegible] | 10 | 7 | 70% | |
| | [illegible] | 10 | 9 | 90% | |
| Sp. 2 | [illegible] | 10 | 7 | 70% | 73.33% |
| | [illegible] | 10 | 7 | 70% | |
| | [illegible] | 10 | 8 | 80% | |
| Sp. 3 | [illegible] | 10 | 6 | 60% | 70% |
| | [illegible] | 10 | 8 | 80% | |
| | [illegible] | 10 | 7 | 70% | |

For segmentation constant thresholds have been used. If we could use dynamic threshold for segmentation it might produce more accurate segmentation which in turn will produce better recognition results. Future work must be able to handle the variability in loudness, speed and noise.

## VI. CONCLUSION

We have used JAVA as the programming language in this regard as JAVA gives special robustness in interfacing and audio signal processing. The segmentation for voiced signal is a very hard task and the sound pattern matching appropriately is very difficult. These have raised some unforeseen problems that have hindered development. The research was carried out in an environment which was not perfectly noise free. This degraded the system performance. Perfect segmentation method is an unavoidable prerequisite for the development of a continuous speech recognition system and speech driven application in turn. Although the developed system produces reasonable results for small vocabulary i.e., small number of commands, there may have reduction in performance with large vocabulary. Practically we have developed a drawing application ready to draw shapes according to the speech command. The level of accuracy is satisfactory. By ensuring a noise free environment & proper segmentation of the recorded signal it is possible to increase the accuracy of recognition. An efficient system should be speaker-independent. So the future researchers should employ speakers of different ages and genders. Neural networks and the Hidden Markov Model (HMM) may also be employed for improved performance. Hope our effort will help to carry out further researches on speech recognition, one-step toward ubiquitous computing & encourages further development in this interesting field.

## REFERENCES

[1] Jean-Claude Junqua, Jean-Paul Haton, "Robustness in Automatic Speech Recognition: Fundamentals and Applications," Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[2] Abul Hasanat Md. Rezaul Karim, Md. Shahidur Rahman and Md. Zafar Iqbal , "Recognition of Spoken Letters in Bangla," ICCIT-2002, Dhaka, Bangladesh.

[3] D.B.Fry, "Technical Aspects of Mechanical Speech Recognition"; and P. Denes, "The Design and Operation of dthe Mechanical Speech Recognizer at University College London", J.British Inst. Radio Engr., Vol. 19; pp. 4, 211-229; 1959.

[4] J.W.Forgie and C.D.Forgie, "Results Obtained From a Vowel Recognition Computer Program", J. Acoust. Soc. Am., Vol. 31, no. 11; pp. 1480-1489; 1959.

[5] J.Suzuki and K.Nakata, "Recognition of Japanese Vowels – Preliminary to the Recognition of Speech", J. Radio Res. Lab, vol. 37, no. 8; pp. 193-212; 1961.

[6] T.Sakai and S.Doshita, "The Phonetic Typewriter, Information Processing 1962", Proc. IFIP Congress, Munich, 1962.

[7] K.Nagata, Y.Kato, and S.Chiba, "Spoken Digit Recognizer for Japanese Language", NEC Res. Develop., No. 6, 1963.

[8] H.F.Olson and H.Belar, "Phonetic Typewriter", J.Acoust. Sco. Am., Vol. 28, no 6; pp.1072-1081; 1956.

[9] Karin Kosina , "Music Genre Recognition", Hagen berg, 2002.

[10] Md. Farukuzzaman Khan,"Computer Recognition of Bangla Speech", M.Phill. thesis, Computer Science and Technology Dept., Islamic University, Kushtia, September, 2002.

[11] M.A. Sid. Ahmed, "Image Processing: Theory, Algorithms and Architectures", McGraw Hill, New York, 1995.