MFCC-Based Voice Recognition System for Home Automation Using Dynamic Programming

Sandeep Joshi[1], **Sneha Nagar**[2]

[1]PG Student, Embedded Systems, Oriental University Indore

[2]Assistant Professor, Electronics & Communication Engineering, Oriental University Indore

Abstract: ***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.***

Keywords: ASR, Mel frequency, Cepstrum coefficient, feature & vector space, distance measurement, embedded system

1. Introduction

In speech recognition, the main goal of the feature extraction step is to compute a parsimonious sequence of feature vectors providing a compact representation of the given input signal. The feature extraction is usually performed in three stages. The first stage is called the speech analysis or the acoustic front end. It performs some kind of spectro temporal analysis of the signal and generates raw features describing the envelope of the power spectrum of short speech intervals. The second stage compiles an extended feature vector composed of static and dynamic features. Finally, the last stage (which is not always present) transforms these extended feature vectors into more compact and robust vectors that are then supplied to the recognizer. Although there is no real consensus as to what the optimal feature sets should look like, one usually would like them to have the following properties: they should allow an automatic system to discriminate between different through similar sounding speech sounds, they should allow for the automatic creation of acoustic models for these sounds without the need for an excessive amount of training data, and they should exhibit statistics which are largely invariant cross speakers and speaking environment.

Mel Spectral Coefficients

The human ear does not show a linear frequency resolution but builds several groups of frequencies and integrates the spectral energies within a given group. Furthermore, the mid-frequency and bandwidth of these groups are non–linearly distributed. The non–linear warping of the frequency axis can be modeled by the so–called mel-scale. The frequency groups are assumed to be linearly distributed along the mel-scale. The so–called mel–frequency f_mel can be computed from the frequency f as follows:

$$\mathbf{f\_mel\ (f)=2595 \cdot log(1+f/(700\ Hz))\ (a)}$$

The human ear has high frequency resolution in low–frequency parts of the spectrum and low frequency resolution in the high–frequency parts of the spectrum. The coefficients of the power spectrum 〖|V (n) |〗 ^2 are now transformed to reflect the frequency resolution of the human ear.

Cepstral Transformation

Since the transmission function of the vocal tract H(f) is multiplied with the spectrum of the excitation signal X(f), we had those un-wanted ―ripples” in the spectrum. For the speech recognition task, a smoothed spectrum is required which should represent H(f) but not X(f). To cope with this problem, cepstral analysis is used. If we look at (3.2), we can separate the product of spectral functions into the interesting vocal tract spectrum and the part describing the excitation and emission properties:

$$\mathbf{S(f) = X(f) \bullet H(f) \bullet R(f)=H(f) \bullet U(f)} \qquad \mathbf{(b)}$$

We can now transform the product of the spectral functions to a sum by taking the logarithm on both sides of the equation:

$$\log〖(S(f))=\log〖(H(f)〗〗\cdot U(f))$$
$$\mathbf{=\log〖(H(f))+\log〖(U(f))〗〗} \qquad \mathbf{(c)}$$

This holds also for the absolute values of the power spectrum and also for their squares:

$$\log〖(|S(f)|^2)=\log〖(|H(f)|^2\cdot|U(f)|^2 〗〗)$$
$$\mathbf{=\log〖(|H(f)|^2 〗)+\log〖(|U(f)|^2)〗} \qquad \mathbf{(d)}$$

In figure 1 we see an example of the log power spectrum, which contains unwanted ripples caused by the excitation signal

$$\mathbf{U(f)=X(f) \bullet R(f).}$$

In the log–spectral domain we could now subtract the unwanted portion of the signal, if we knew | 〖U(f)|〗 ^2 exactly. But all we know is that U(f) produces the ―ripples”, which now are an additive component in the log–spectral domain, and that if we would interpret this log–spectrum as a time signal, the ―ripples” would have a ―high frequency” compared to the spectral shape of |H(f)|. To get rid of the influence of U(f), one would have to get rid of the ―high-frequency” parts of the log–spectrum (remember, we are dealing with the spectral coefficients as if they would represent a time signal). This would be a kind of low–pass filtering. The filtering can be done by transforming the log–spectrum back into the time–domain (in the following, 〖FT〗 ^(-1) denotes the inverse Fourier transform):

s ̃(d)= 〖FT〗 ^(-1) {log⁡(|S(f)|^2) }
= 〖FT〗 ^(-1) {log(|H(f)|^2) }+ 〖FT〗 ^(-1){log(|U(f)|^2)}

(e)

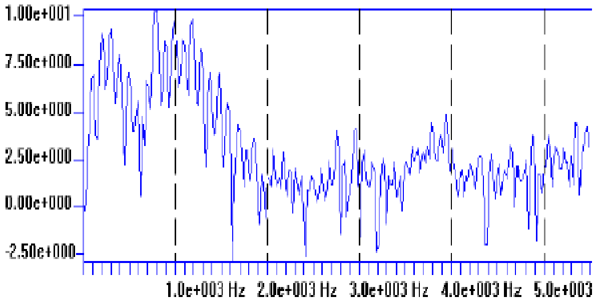

Figure 1: Log power spectrum of the vowel /a: / (f_s = 11 kHz). The ripples in the spectrum are caused by X (f)

The inverse Fourier transform brings us back to the time–domain (d is also called the delay or frequency), giving the so–called cepstrum (a reversed ―spectrum”). The resulting cepstrum is real–valued, since 〖|U(f)|〗 ^2 and 〖|H(f)|〗 ^2 are both real-valued and both are even: 〖|U(f)|〗 ^2= 〖|U(-f)|〗 ^2 and 〖|H(f)|〗 ^2= 〖|H(-f)|〗 ^2. Applying the inverse DFT to the log power spectrum coefficients log⁡ 〖 〖|V(n)|〗 ^2 〗 yields:

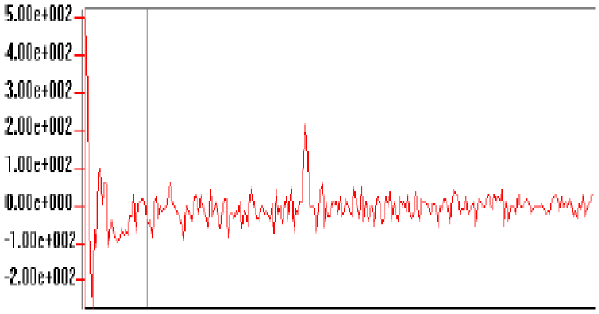

Figure 2: Cepstrum of the vowel /a: / (f_s = 11 kHz, N = 512). The ripples in the spectrum result in a peak in the cepstrum

Mel Cepstrum

Now that we are familiar with the cepstral transformation and cepstral smoothing, we will compute the mel cepstrum commonly used in speech recognition. As stated above, for speech recognition, the mel spectrum is used to reflect the perception characteristics of the human ear. In analogy to computing the cepstrum, we now take the logarithm of the mel power spectrum (3.11) (instead of the power spectrum itself) and transform it into the frequency domain to compute the so–called mel cepstrum. Only the first Q (less than 14) coefficients of the mel cepstrum are used in typical speech recognition systems. The restriction to the first Q coefficients reflects the low–pass liftering process as described above.

Since the mel power spectrum is symmetric due to (e), the Fourier-Transform can be replaced by a simple cosine transform:

$$c(q) = \sum_{k=0}^{K-1} \boldsymbol{log}\big(\boldsymbol{G(k)}\big)\cdot \boldsymbol{cos}\left(\frac{\boldsymbol{\pi q(2k+1)}}{\mathbf{2K}}\right) ; \boldsymbol{q} = \mathbf{0,1,\dots,}\boldsymbol{Q}-\mathbf{1} \quad \textbf{(f)}$$

While successive coefficients G(k) of the mel power spectrum are correlated, the Mel Frequency Cepstral Coefficients (MFCC) resulting from the cosine transform (f) are de-correlated. The MFCC are used directly for further processing in the speech recognition system instead of transforming them back to the frequency domain.

2. Feature and Vector Space

Until now, we have seen that the speech signal can be characterized by a set of parameters (features), which will be measured in short intervals of time during a preprocessing step. Before we start to look at the speech recognition task, we will first get familiar with the concept of feature vectors and vector space.

If you have a set of numbers representing certain features of an object you want to describe, it is useful for further processing to construct a vector out of these numbers by assigning each measured value to one component of the vector. As an example, think of an air conditioning system which will measure the temperature and relative humidity in your office. If you measure those parameters every second or so and you put the temperature into the first component and the humidity into the second component of a vector, you will get a series of two–dimensional vectors describing how the air in your office changes in time. Since these so–called feature vectors have two components, we can interpret the vectors as points in a two–dimensional vector space. Thus we can draw a two–dimensional map of our measurements as sketched below. Each point in our map represents the temperature and humidity in our office at a given time. As we know, there are certain values of temperature and humidity which we find more comfortable than other values. In the map the comfortable value– pairs are shown as points labeled ―+” and the less comfortable ones are shown as ―-”. You can see that they form regions of convenience and inconvenience, respectively.

Let‘s assume we would want to know if a value–pair we measured in our office would be judged as comfortable or as uncomfortable by you. One way to find out is to initially run a test series trying out many value–pairs and labeling each points either ―+” or ―-” in order to draw a map as the one you saw above.

Now if you have measured a new value–pair and you are to judge if it will be convenient or not to a person, you would have to judge if it lies within those regions which are marked in your map as ―+” or if it lies in those marked as ―-”.This is our first example of a classification task: We have two classes (―comfortable” and ―uncomfortable”) and a vector in feature space which has to be assigned to one of these classes. — But how do you describe the shape of the regions and how can you decide if a measured vector lies within or without a given region? In the following chapter we will learn how to represent the regions by prototypes and how to measure the distance of a point to a region.

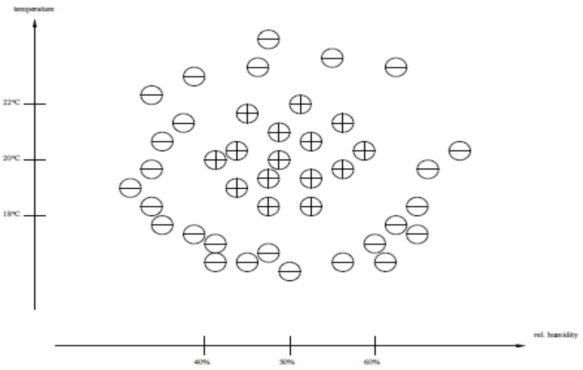

Figure 3: A map of feature vectors

Classification of Vectors

A) Prototype Vectors

The problem of how to represent the regions of ―comfortable” and ―uncomfortable” feature vectors of our classification task can be solved by several approaches. One of the easiest is to select several of the feature vectors we measured in our experiments for each of our classes (in our example we have only two classes) and to declare the selected vectors as ―prototypes” representing their class. We will later discuss how one can find a good selection of prototypes using the ―kmeans algorithm”. For now, we simply assume that we were able to make a good choice of the prototypes, as shown in figure 4.

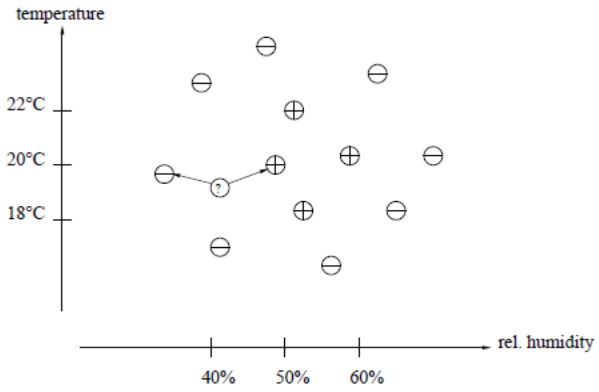

Figure 4: Selected prototypes

B) Nearest Neighbor Classification

The classification of an unknown vector is now accomplished as follows: Measure the distance of the unknown vector to all classes. Then assign the unknown vector to the class with the smallest distance. The distance of the unknown vector to a given class is defined as the smallest distance between the unknown vector and all of the prototypes representing the given class. One could also verbalize the classification task as: Find the nearest prototype to the unknown vector and assign the unknown vector to the class this ―nearest neighbor” represents (Hence the name). Fig. 3.13 shows the unknown vector and the two ―nearest neighbors” of prototypes of the two classes. The classification task we described can be formalized as follows: Let Ω ={ ω_1,ω_2. ..ω_((V -1))} be the set of classes, V being the total number of classes. Each class is represented by its prototype vectors p (k,ω_v), where k = 0,1,...,(K_(ω_v)- 1). Let x denote the unclassified vector. Let the distance measure between the vector and a prototype be denoted as d (x,p (k,ω_v)) Then the class distance between x ⃗ and the class ω_v is defined as:

$$\boldsymbol{d_{\omega_v}(\vec{x}) = \min_k\{d(\vec{x},\vec{p}_{k,\omega_v})\}; k = 0, 1, \dots, (k-1)} \qquad \textbf{(g)}$$

3. Proposed System

In this work we have investigated the use of mel frequency spectral coefficient technique for automatic speech recognition and then we have implemented this on matlab and used matlab to send serial data to microcontroller which then controls the home lights according to the speech signal.

The result has shown that the system designed by us is capable of recognizing 10 different speech signals. Although the system is designed to work for 10 speech signals it can be very easily upgraded for any number of signals. The words that could be recognized have no limits or any special characteristics and the system performance in terms of speech recognition is very efficient.

The words that can be recognized must be isolated words and we have found that best results have came when the word have a duration of less than 0.75 sec. Longer words or group of words are difficult to recognize with this method.

Another factor that we have analyzed is the total time elapsed for detection. Our systems main characteristics are that it is very efficient and time for detection is very less. The system takes an average of only 0.7 sec for analyzing and recognizing the speech signal.

The embedded system that we have used is based on P89V51RD2 microcontroller and its main purpose is to take serial data in real time form the matlab and then according to the signals taken it turns the lights ON and OFF. Thus the main results could be summarized as below:

1) Speech processing algorithm: - Mel frequency Cepstral Coefficient.
2) Speech recognition algorithm: - Dynamic programming.
3) Recognition type: - isolated word recognition.
4) Word timing for best recognition: - 0.75 sec.
5) Implementation of software :- Matlab 11.0
6) Hardware implementation platform: - P89V51RD2 based.
7) Interface standard used :- Serial data RS 232

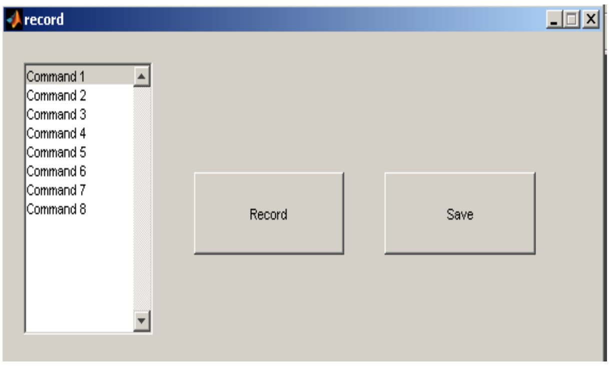

Figure 5: Record GUI

Figure above shows the graphical user interface for recording user voice patterns. As can be seen in the above diagram that a maximum of 8 commands can be recorder and then the Matlab will process them according to MFCC (mel frequency spectral coefficient). Thus this GUI will take user voice and records its sample. The maximum duration of word that can be stored is 0.75 sec.

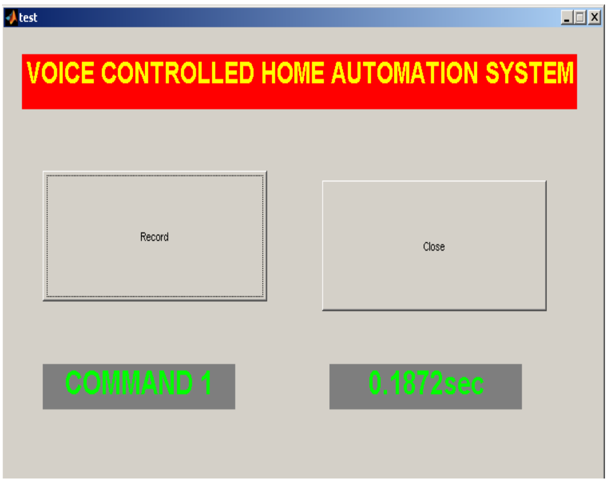

Figure 6: Final GUI showing Command 5 on the utterance of the word stored at that location

The GUI shown above has the following functions:

a) To take user input and compare it with the existing database.
b) Find the distance between the voice inputs and then show the result in the text box .
c) Compute the CPU time taken for the entire process and displaying it on the text box.
d) Sending serial data to the microcontroller attached. The serial data send for each command is shown in the tabular format as below :

S.no	Command	Serial Data
1	COMMAND 1	‗a‘
2	COMMAND 2	‗b‘
3	COMMAND 3	‗c‘
4	COMMAND 4	‗d‘
5	COMMAND 5	‗e‘
6	COMMAND 6	‗f‘
7	COMMAND 7	‗g‘
8	COMMAND 8	‗h‘

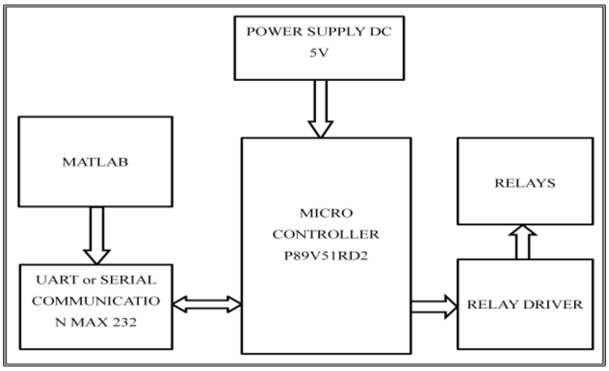
POWER SUPPLY DC 5V
MATLAB
MICRO CONTROLLER P89V51RD2
RELAYS
UART or SERIAL COMMUNICATIO N MAX 232
RELAY DRIVER

Figure 7: Block Diagram of the system

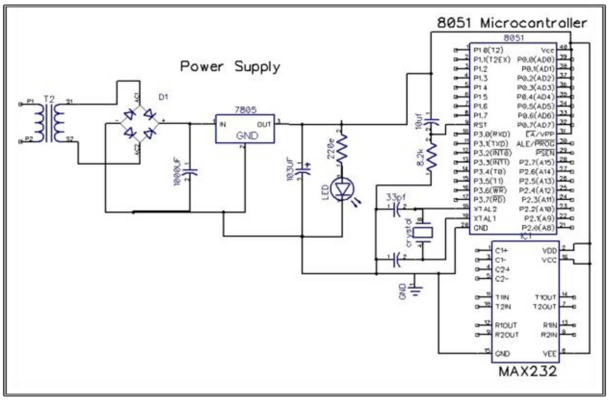

Figure 8: Circuit design

4. Conclusion

The proposed system, as promised, recognized isolated and speaker independent work with fair accuracy. The overall design of the system was satisfactory and the graphical user interface deployment provided further ease of use.

One thing about the system was its execution speed. The dynamic programming approach to speech recognition gives fairly lower hex size of the executable file because of the simplicity of algorithm. In this manner it is easier to implement the system on a mobile device which nowadays is the necessity for any computational program.

The primary concept was to use dynamic programming which was earlier used in RDBMS systems for identification of stored data. The dynamic programming concept mixed with Mel and spectrum coefficients helped us in finding the potential of the algorithm in speech recognition systems. The system though being simple and light weight is till date not efficient enough to recognize speaker dependent voices also the ambient noise cancellation has not being incorporated for the same. These features when added could make the system more robust and capable in speech recognition.

References

[1] Sadaoki Furui, 50 years of Progress in speech and Speaker Recognition Research, ECTI Transactions on Computer and Information Technology, Vol. 1. No. 2 November 2005.

[2] K. H. Davis, R. Biddulph, and S. Balashek, Automatic recognition of spoken Digits, J. Acoust. Soc. Am., 24(6):637-642, 1952.

[3] H. F. Olson and H. Belar, Phonetic Typewriter, J. Acoust. Soc. Am., 28(6):1072-1081, 1956.

[4] D. B. Fry, Theoritical Aspects of Mechanical speech Recognition, and P. Denes, The design and Operation of the Mechanical Speech Recognizer at Universtiy College London, J. British Inst. Radio Engr., 19:4, 211-299, 1959.

[5] J. W. Forgie and C. D. Forgie, Results obtained from a vowel recognition computer program, J. A. S. A., 31(11), pp. 1480-1489. 1959.

[6] J. Suzuki and K. Nakata, Recognition of Japanese Vowels Preliminary to the Recognition of Speech, J. Radio Res. Lab 37 (8):193-212, 1961.

[7] T. Sakai and S. Doshita, The phonetic typewriter, Information processing 1962, Proc. IFIP Congress, 1962.

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

[9] T. B. Martin, A. L. Nelson, and H. J. Zadell, Speech Recognition b Feature Abstraction Techniques, Tech. Report AL-TDR-64-176, Air Force Avionics Lab, 1964.

[10] T. K. Vintsyuk, Speech Discrimination by Dynamic Programming, Kibernetika, 4(2):81-88, Jan. -Feb. 1968.

[11] H. Sakoe and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-26(1). pp. 43- 49, 1978.

[12] D. R. Reddy, An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave, Tech. Report No. C549, Computer Science Dept., Stanford Univ., September 1966.

[13] V. M. Velichko and N. G. Zagoruyko, Automatic Recognition of 200 words, Int. J. Man-Machine Studies, 2:223, June 1970.

[14] H. Sakoe and S. Chiba, Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-26(1):43- 49, February 1978.

[15] F. Itakura, Minimum Prediction Residula Applied toSpeech Recognition, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-23(1):67-72, February 1975.

[16] C. C. Tappert, N. R. Dixon, A. S. Rabinowitz, and W. D. Chapman, Automatic Recognition of Continuous Speech Utilizing Dynamic Segmentation, Dual Classification, Sequential Decoding and Error Recover, Rome Air Dev. Cen, Rome, NY, Tech. Report TR-71-146, 1971.

[17] F. Jelinek, L. R. Bahl, and R. L. Mercer, Design of a Lingusistic Statistical Decoder for the Recognition of Continuous Speech, IEEE Trans. Information Theory, IT- 21:250-256, 1975.

[18] F. Jelinek, The Development of an Experimental Discrete Dictation Recognizer, Proc. IEEE, 73(11):1616- 624, 1985.

[19] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, Speaker Independent Recognition of Isolated Words Using Clustering Techniques, IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-27:336-349, August 1979.

[20] Zhenhao Ge, Sudhendu R. Sharma, Mark J. T. Smith, ―Adaptive Frequency spectral Coefficients for Word Mispronunciation Detection", 4th International Congress on Image and Signal Processing, 2011, pages 2388- 2391.

[21] Matthew Gibson and William Byrne, ―Unsupervised Intralingua and Cross-Lingual Speaker Adaptation for HMM-Based Speech Synthesis Using Two-Pass Decision Tree Construction", IEEE transactions on audio, speech, and language processing, VOL. 19, NO. 4, MAY 2011, pages 895- 904

[22] Matthew Gibson, ―Two-pass decision tree construction for unsupervised adaptation of HMM-based synthesis models".

[23] UmaraniJ. Suryawanshi, Prof. Dr. S. R. Ganorkar, ―Hardware Implementation of Speech Recognition Using MFCC and Euclidean Distance", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 8, August 2014.

[24] Chanwoo Kim and Kwang-deok Seo, ―Robust DTW-based Recognition Algorithm for Hand-held Consumer Devices", 0098 3063/05/$20. 00 © 2005 IEEE.

[25] Santosh K. Gaikwad, ―A Review on Speech Recognition Technique", International Journal of Computer Applications (0975 – 8887) Volume 10 No. 3, November 2010.

[26] WEI Ming-zhe, LI Xi, REN Li-mian, ―Improved DTW Speech Recognition Algorithm Based on the MEL Frequency Cepstral Coefficients", Information and Communication Technology and Smart Grid, 2010.