A Comparative Study Of LPCC And MFCC Features For The Recognition Of Assamese Phonemes

Utpal Bhattacharjee

Department of Computer Science and Engineering, Rajiv Gandhi University, Rono Hills, Doimukh, Arunachal Pradesh, India, Pin-791112

Abstract

In this paper two popular feature extraction techniques Linear Predictive Cepstral Coefficients (LPCC) and Mel Frequency Cepstral Coefficients (MFCC) have been investigated and their performances have been evaluated for the recognition of Assamese phonemes. A multilayer perceptron based baseline phoneme recognizer has been built and all the experiments have been carried out using that recognizer. In the present study, attempt has been made to evaluate the performance of the speech recognition system with different feature set in quiet environmental condition as well as at different level of noise. It has been observed that at noise free operating environment when same speaker is used for training and testing the system, the system given 100% recognition accuracy for the recognition of Assamese phones for both the feature set. However, the performance of the system degrades considerably with increase in environmental noise level.It has been observed that the performance of LPCC based system degrades more rapidly compare to MFCC based system under environmental noise condition whereas under speaker variability conditions, LPCC shows relative robustness compare to MFCC though the performance of both the systems degrades considerably.

Key Terms: **Speech Recognition, LPCC, MFCC, MLP**

1. Introduction

Automatic speech recognition is the task of recognizing the spoken word from speech signal. A survey in the robustness issues associated with automatic speech recognition has been reported by several workers [1, 2]. In our present study, the difficulties due to speaker variability and environmental factors are considered.

A word may be uttered by the same user differently because of the difference in emotional level, health status, surrounding environment (noise/quietness) etc. Again, utterance of the same word varies due to gender, age, dialect, influence of other languages on the speaker etc. Another layer of variation is introduced by the acoustical environment where the speech recogniser operates. These variations are due to background noise, microphone, transmission channel, reverberation etc. In this paper we evaluate the performance of LPCC and MFCC feature vectors as front-end of a speech recognizer under environmental variability and speaker variability conditions.

In the present study a Multi-layer perceptron based baseline system has been built for the recognition of Assamese phonemes. To categorize the related features into different classes and remove repeating data, Self Organized Map (SOM) has been used. The feature trajectory obtained from the phoneme signal has been reduced into six cluster centres. The reduced feature vector has been fed to the MLP based phoneme recognizer.

Assamese is the major language in the North Easter part of India with its own unique identity, culture and language through its origins root back to Indo-European family of language. Assamese is the easternmost member of the new Indo-Aryan (NIA) subfamily of languages spoken in Assam and many part of North-Eastern India. The Assamese phonemic inventory consists of eight oral vowel phonemes, three nasalized vowel phonemes and twenty-two consonant phonemes. The phonemes of the Assamese language are given below[4]:

Table 1(a): Vowels of Assamese language

Vowel Type	Position	Front	Central	Back
Oral Vowel	High	*i*		*u*
	High-mid			*ʊ*
	Mid	*e*		*o*
	Low-mid	*ɛ*		*ɔ*
	Low		*a*	
Nasalized Vowel	High	*i)*		*u)*
	Low		*a)*	

Table 1(b): Consonants of Assamese language

Phoneme Type	Labial	Alveolar	Velar	Glottal
Voiceless stops	*p* p^h	*t* t^h	*k* k^h	
Voiced stops	*b* b^h	*d* d^h	*g* g^h	
Voiceless fricatives		*s*	*x*	*h*
Voiced fricatives		*z*		
Nasals	*m*	*n*	*ŋ*	
Approximants	*w*	*ɹ*		
Lateral		*l*		

The paper is organized as follows. Section II discusses LPCC and MFCC methods for speech parameterization in details. The baseline speech recognition system is described in section III. In section IV we describe the experimental setup and database used. Section V is dedicated for the description of the experiments and results obtained. The paper is concluded in section VI.

2. The LPCC and MFCC methods for Speech Parameterization

In this paper two methods of speech parameterization namely Linear Predictive Coding Cepstral Coefficients (LPCC) and Mel Frequency Cepstral Coefficients (MFCC) have been used as front-end feature extractor. The details of both the methods have been given below:

2.1 Linear Predictive Cepstral Coefficients (LPCC)

The Linear Predictive analysis is based on the assumption that the shape of the vocal tract governs the nature of the sound being produced. To study the property quantitatively, the vocal tract is modeled by a digital all-pole filter [5]. The transfer function in z-domain is given by

$$V(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} \qquad \text{---(1)}$$

Where V(z) is the vocal tract transfer function. G is the gain of the filter and $\{a_k\}$ is a set of autocorrelation coefficients called Linear Prediction Coefficients (LPC). The upper limit of summation, p, is the order of the all-pole filter. The set of LPC determines the characteristic of the vocal tract transfer function.

Autocorrelation method[5] is an efficient method for evaluating the LPC set and the filter gain. It involves calculating a matrix of simultaneous equations and the autocorrelation of the windowed speech frames. The matrix of equations that need to be solved is

$$\begin{bmatrix} R[0] & R[1] & \cdots & R[p-1] \\ R[1] & R[2] & \cdots & R[p-2] \\ \vdots & \vdots & \ddots & \cdot \\ R[p-1] & R[p-2] & \cdots & R[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R[1] \\ R[2] \\ \vdots \\ R[p] \end{bmatrix} \ldots(2)$$

Where R[n] is the autocorrelation function of a windowed speech signal.

The Gain of the all-pole filter can be found by solving the following equation

$$G = \sqrt{R[0] - \sum_{k=1}^{p} a_k R[k]} \qquad \text{--- (3)}$$

Since the matrix on the left of Eq.(2) is a Toeplitz matrix, recursive algorithm can be used to solve the above equation. Levinson-Durbin recursive procedure [5] has been applied to solve the equation, which is given below

$$E^0 = R[0]$$

$$k_i = \frac{R[i] - \sum_{j=1}^{i-1} a_j^{(i-1)} R[i-j]}{E^{(i-1)}}, 1 \le i \le p$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, 1 \le j \le i-1$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

--- (4)

The above equation is solved recursively for i=1, 2, 3, …, p. When i reaches the pth iteration, the set of LPC and the filter gain are given as follows,

$$a_j = a_j^{(p)}, 1 \le j \le p \qquad \text{--- (5)}$$

The gain of the all-pole filter model, G, is given by the following equation,

$$G = \sqrt{E^{(p)}} \qquad \text{--- (6)}$$

Cepstral analysis refers to the process of finding the cepstrum of a speech sequence. Cepstral coefficients can be calculated from the LPC via a set of recursive procedure [5]. The cepstral coefficients obtained in this way are called Linear Predictive Cepstral Coefficients (LPCC). The recursive procedure is given below:

$$c[0] = \ln(G)$$

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

$$c[n] = \sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) c[n-k] a_k \ for\ n > p$$

--- (7)

2.2 Mel Frequency Cepstral Coefficients (MFCC)

Mel Frequency Cepstral Coefficients (MFCC) is one of the most commonly used feature extraction method in speech recognition. The technique is called FFT based which means that feature vectors are extracted from the frequency spectra of the windowed speech frames.

The Mel frequency filter bank is a series of triangular bandpass filters. The filter bank is based on a non-linear frequency scale called the mel-scale. According to Stevens et al[6], a 1000 Hz tone is defined as having a pitch of 1000 mel. Below 1000 Hz, the Mel scale is approximately linear to the linear frequency scale. Above the 1000 Hz reference point, the relationship between Mel scale and the linear frequency scale is non-linear and approximately logarithmic. The following equation describes the mathematical relationship between the Mel scale and the linear frequency scale

$$f_{Mel} = 1127.01 \ln\left(\frac{f}{700} + 1\right) \qquad \text{--- (8)}$$

The Mel frequency filter bank consist of triangular bandpass filters in such a way that lower boundary of one filter is situated at the center frequency of the previous filter and the upper boundary situated in the center frequency of the next filter. A fixed frequency resolution in the Mel scale is computed, corresponding to a logarithmic scaling of the repetition frequency, using $\Delta f_{Mel} = (f_{H(mel)} - f_{L(mel)})/(M+1)$ where $f_{H(mel)}$ is the highest frequency of the filter bank on the Mel scale, computed from f_{max} using equation (8), $f_{L(mel)}$ is the lowest frequency in Mel scale, having a corresponding f_{min} and M is the number of filter bank. The values considered for the parameters in the present study are: f_{max} =8 KHz, f_{min} =0 Hz and M=20. The center frequencies on the Mel scale are given by

$$f_{cm(Mel)} = f_{L(Mel)} + \frac{m(f_{H(Mel)} + f_{L(Mel)})}{M+1}, 1 \le m \le M$$

---(9)

The center frequencies in Hertz, is given by

$$f_{cm} = 700\left(e^{\frac{f_{cm\,(Mel)}}{1127.01}} - 1\right) \qquad \text{---(10)}$$

Equation (10) is inserted into equation (8) to give the Mel filter bank. Finally, the MFCCs are obtained by computing the discrete cosine transform of $X'(m)$ using

$$c(l) = \sum_{m=1}^{M} X'(m)\cos(l\frac{\pi}{M}(m-\frac{1}{2})) \qquad \text{--- (11)}$$

for$l$ = 1, 2, 3, ….., M where $c(l)$ is the l^{th} MFCC.

The time derivative is approximated by a linear regression coefficient over a finite window, which is defined as

$$\Delta\, c_t(l) = \left[\sum_{K=2}^{2} k \;\; c_{t-k}(m)\right].G,\; 1 \le l \le M \qquad \text{--- (12)}$$

where $c_t(l)$ is the l^{th} cepstral coefficient at time t and G is a constant used to make the variances of the derivative terms equal to those with the original cepstral coefficients.

3. Baseline Speech Recognition System

A baseline speech recognition system was developed during the present work using Multilayer perceptron to recognize the phonemes of Assamese language. To reduce the feature vector, Self Organized Map has been used. The details of Multilayer Perceptron and Self Organized Map have been given below:

3.1 Self-Organizing Map (SOM)

Kohonen [7] proposed a Neural Network (NN) architecture which can automatically generate self-organization properties during unsupervised learning process, namely, a Self-Organizing Map (SOM). All the input vectors of utterances are presented into the network sequentially in time without specifying the desired output. After enough input vectors have been presented, weight vectors from input to output nodes will specify cluster or vector centers that sample the input space such that the point density function of the vector centers tends to approximate the probability density function of the input vectors. In addition, the weight vectors will be organized such that topologically close nodes are sensitive to inputs that are physically similar in Euclidean distance. Kohonen has proposed an efficient learning algorithm for practical applications. This learning algorithm has been used in the proposed system.

Using the fact that the SOM is a Vector Quantization (VQ) scheme that preserves some of the topology in the original space [8], the basic idea behind the approach proposed in this work is to use the output of a SOM trained with the output of the speech processing block to obtain reduced feature vector (binary matrix) that preserve some of the behaviour of the original feature vector. The problem is now reduced to find the correct number of neurons (Dimension of SOM) for constituting the SOM. Based on the ideas stated above, the optimal dimension size of SOM has to be searched in order to ensure the SOM has enough neurons to reduce the dimensionality of the feature vector while keeping enough information to achieve high recognition accuracy [9].

i) SOM Architecture

The SOM consists of only one real layer of neurons. The SOM is arranged in a 2-D lattice. This architecture implements similarity measure using Euclidean distance measurement. In fact, it measures the cosine of the angle between normalized input and weight vectors. Since the SOM algorithm uses Euclidian metric to measure distances between data vectors, scaling of variables was deemed to be an important step and all input vectors has been normalized to the unity. The input vector is normalized between -1 and +1 before it is fed into the network. Usually, the output of the network is given by the most active neuron as the winning neuron.

ii) Learning Algorithm

The objective of the learning algorithm in SOM neural networks is the formation of the feature map whichcaptures the essential characteristics of the p-dimensional input data and maps them on the typically 2-D feature space. The learning algorithm captures two essential aspects of the map formation, namely, competition and cooperation between neurons of the output lattice.

Assuming Mij (t) = { m1ij(t), m2 ij (t)………, mNij (t)} as the weight vector of node (i,j) of the feature map at time instance t; i, j = 1, …, M are the horizontal and vertical indices of the square grid of output nodes, N is the dimension of the input vector. Denoting the input vector at time t as X(t), the learning algorithm can be summarized as follows [8]:

1. Initializing the weights

Prior to training, each node's weights must be initialized. Typically these will be set to small standardized random values. The weights in the SOM in this research are initialized so that 0 < weight < 1.

2. *Calculating the winner node - Best Matching Unit (BMU)*

To determine the BMU, one method is to iterate through all the nodes and calculate the Euclidean distance between each node's weight vector and the current input vector. The node with a weight vector closest to the input vector is tagged as the BMU. The Euclidean distance is given as:

$$Dist = \sqrt{\sum_{i=0}^{i-n}(X_i(t) - M_{ij}(t))^2} \qquad \text{--- (13)}$$

To select the node with minimum Euclidean distance to the input vector X(t):

$$\left\|X(t) - M_{i_c j_c}(t)\right\| = \min_{i,j}\left\{\left\|X(t) - M_{ij}(t)\right\|\right\} \text{--- (14)}$$

3. Determining the Best Matching Unit's Local Neighborhood

For each iteration, after the BMU has been determined, the next step is to calculate which of the other nodes are within the BMU's neighbourhood. Radius of the neighborhood is calculated. The area of the neighborhood shrinks over time using the exponential decay function:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \textit{t=1,2,3,...} \qquad \text{---(15)}$$

where σ_0, denotes the width of the lattice at time = 0, *t* is the current time-step. If a node is found to be within the neighbourhood then its weight vector is adjusted as shown in next step.

4. Adjusting the weights

Every node within the BMU's neighborhood including the BMU (ic, jc) has its weight vector adjusted according to the following equation:

$$M_{ij}(t+1) = \begin{cases} M_{ij}(t) + \alpha(t)\left(X(t) - M_{ij}(t)\right) \\ for\ i_c - N_c(t) \le i \le i_c + N_c(t) \\ and\ j_c - N_c(t) \le j \le j_c + N_c(t) \\ M_{ij}(t+1) = M_{ij}(t), \\ \quad \text{for all other indices } (i,j) \end{cases} \qquad \text{---(16)}$$

wheret represents the time-step and α is a small variable called the learning rate, which decreases with time. Basically, this means that the new adjusted weight for the node is equal to the old weight, plus a fraction of the difference α between the old weight M and the input vector X. The decay of the learning rate is calculated each iteration using the following equation:

$$\alpha(t) = \alpha_0 \exp\left(-\frac{t}{\lambda}\right), \quad \textit{t=1,2,3,......} \qquad \text{---(17)}$$

Ideally, the amount of learning should fade over distance similar to the Gaussian decay.

So, an adjustment is made to equation (16) which shown as equation below:

$$M_{ij}(t+1) = M_{ij}(t) + \Theta(t)\alpha(t)(X(t) - M_{ij}(t)) \text{---(18)}$$

whereΘ represents the amount of influence a node's distance from the BMU has on its learning. Θ(t) is given by equation below:

$$\Theta(t) = \exp\left(\frac{dist^2}{2\sigma^2(t)}\right), \textit{t=1,2,3,.........} \qquad \text{---(19)}$$

where 'dist' is the distance a node is from the BMU and σ is the width of the neighbourhood function as calculated by equation (15). Additionally, Θ also decays over time.

5. Update time*t = t + 1*, add new input vector and go to Step 2.
6. Continue until*α(t)* approach a certain pre-defined value or t reach maximum iteration.

3.2 Multilayer Perceptron based Phoneme Recognizer

In the present study Multilayer Perceptron (MLP) has been used to design the speech recognizer to recognize the phonemes of Assamese languages. The MLP consist of input, output and three hidden layers. To train the MLP, a modified version of well-known Back Propagation Algorithm [10] has been used. To avoid the oscillations at the local minima a momentum constant has been introduced which provides optimization in the weight updating process. The algorithm is detailed below:

1) Initialization

The weights of each layer have been initialized to random number lies between -1 to 1.

2) Forward computation

In the forward pass the synaptic weight remain unaltered throughout the network and functional signal of the network is computed neuron-by-neuron basis. The induced local field $v_j^{(l)}(n)$ for neuron j in layer l which is due to the functional signal produced by neurons of layer l-1 is given by [11]

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{l-1}(n) \qquad \text{--- (20)}$$

where m is the total number of inputs, excluding bias applied to neuron j. The synaptic weight wjo, corresponds to fixed input y0=+1, equals the bias bj applied to neuron j. Hence the functional signal appearing at the output neuron j of layer l is expressed as

$$y_j^{(l)} = \psi_j(v_j(n)) \qquad \text{--- (21)}$$

If the neuron *j* is in the first hidden layer

$$y_j^{(0)} = x_j(n) \qquad \text{--- (22)}$$

wherexj(n) is the jth element of the input vector. If on the other hand, network j is in the output layer of the network, and L the depth of the network, then

$$y_j^{(L)} = o_j(n) \qquad \text{--- (23)}$$

whereoj(n) is the jth element of the output vector. The output is compared with the desired response dj(n), obtain the error signal ej(n) for the jth output neuron

$$e_j(n) = d_j(n) - o_j(n) \qquad \text{--- (24)}$$

3) Backward computation

The backward pass starts at the output layer by passing the error signal leftward through the network, layer by layer , and recursively computing the δ (i.e. the local gradient) for each neuron as follows:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\psi'\left(v_j^{(L)}(n)\right), \\ for\ neuron\ j\ in\ output\ layer\ L \\ \left(v_j^{(L)}(n)\right)\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n), \\ for\ neuron\ inhidden\ layer\ l \end{cases} \qquad \text{--- (25)}$$

The weight updation is taking place in accordance with the following rule -

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n-1)] + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n) \qquad \text{--- (26)}$$

whereη is the learning rate and α is momentum constant.

It has been observed that MLP based speech recognizer work better if the input and output lies between 0

– 1. Therefore, the input vector has been normalized with respect to their maximum and minimum value.

A momentum constant α has been used to avoid oscillation at the local minima.

The learning rate parameter has been changed gradually with each epoch number as expressed by equation given below:

$$\eta(epochNumber) = \eta_0 \exp\left(\frac{-\ epochNumber}{100}\right)$$

--- (27)

whereη_0 is the initial learning rate parameter.

4. Experimental Setup and Database Used

4.1 Experimental Setup

The baseline speech recognition process has the following steps:

(a) Digitizing the speech that is to be recognized
(b) Compute the features of the speech signal
(c) Reduce the feature set using Self-Organized Map (SOM)
(d) An MLP based phoneme classifier is used to classify each set of feature corresponding to the phoneme utterance to corresponding phoneme.

Speech is first filtered to a bandwidth of 4 KHz and then digitized at 8 KHz. sampling rate. The digitized speech is then emphasized using a simple first order digital filter with transfer function H (z) = 1 – 0.95 z -1. The pre-emphasized speech is then blocked into frames of length 256 samples. The objective is to block the speech signal into frame of 30 microseconds which contain 240 samples. However to make the FFT efficient, length is made multiple of 2. The frame frequency is 100 Hz.In order to remove the leakage effects and to smooth the edges, each frame is multiplied by a Hamming window as define by

$$w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right),$$

$$n \in [0, n-1]\ and\ N = 256 \qquad \text{--- (27)}$$

From each windowed speech signal two types of feature were extracted LPCC and MFCC. To obtain the LPCC, 12th order predictor is used and 12 LPCC coefficients were obtained by applying the method described in section II. Similarly, each windowed frame is passes through a bank of 20 triangular bandpass filter and was constrained into a frequency band of 300-3400 Hz. The 0th cepstral coefficient has not been considered at it corresponds to the energy of the whole frame. To reduce the computational load only next 12 coefficients have been used in the present study. To capture the time varying nature of the speech signal, the first order derivatives of the LPCC and MFCC feature are appended with the original feature set from each frame. Thus, we get two distinct set of 24-dimensional feature vectors.

In order to reduce the volume of data without losing the topological information, we use self-organized map (SOM) to cluster the feature vector into six clusters. The centroids of the clusters are dynamically detected. Thus both LPCC and MFCC feature vectors are reduced into 6 cluster centers each.

To carry out the recognition task a MLP-based recognizer is designed with 144 input nodes, 3-hidden layers with different numbers of node and a output layer with 33 nodes corresponds to the 33 phonemes of Assamese language. Experimentally, the numbers of nodes in the three hidden layers have been fixed at 99, 68 and 47 respectively and the same configuration has been used in all the experiments.

4.2 Databases

The database used in the present study has been described below:

Dataset-I(Clean): The dataset contain 20 utterances for each phoneme for each speaker. Speech data has been collected from 50 speakers, 27 male and 23 female. To collect the phoneme utterances, recording has been done for isolated words. The isolated words are selected in such a way that they include all the phonemes at least 20 times. The recording has been done using headphone microphone at 8 KHz sampling rate with 16 bit mono resolution. The isolated words so recorded have been manually segmented into phonemes using Praat and EasyAlign tool.

Dataset-II (20dB SNR): The Dataset-II is a noisy version of the Dataset-I. Simulated 20dB Gaussian noise has been added digitally to the samples of Dataset-I to obtain the dataset.

Dataset-III (15dB SNR): The Dataset-III is similar to Dataset-II expect the SNR of the simulated Gaussian noise added to the clean speech is 15 dB.

Dataset-IV (10dB SNR): The SNR of the simulated Gaussian noise added to the clean speech is 10 dB.

5. Experiment

The recognizer is trained with clean speech (Dataset-I) using modified version of back-propagation algorithm as described in section II. 100 occurrences of each phoneme have been considered for training the system collected from 10 speakers, 5 male and 5 female. Once the system is converged it is tested with the remaining phoneme occurrences of the same dataset. Testing has been done for evaluate the performance of the system when training and the testing speakers are same as well as when the speakers are different. The results of the experiments are given in Table-3.

Table-3: Speaker Recognition using clean speech

Feature Set	Recognition Accuracy (%)	
	Same Speaker	Different Speaker
LPCC	100	94.23
MFCC	100	89.14

In the next experiments, the same speech recognizer has been tested for speech data with different level of noise, i.e., with Dataset-II, III and IV. Experiments were carried out using speech data from the same group of speakers used for training the system. The performance of the speech recognition system has been reported in Table-4.

Table 4: Performance of the speech recognition system at different level of noise

SNR	Feature Set	Recognition Accuracy (%)
20 dB	LPCC	73.27
	MFCC	97.03
15 dB	LPCC	59.41
	MFCC	85.15
10 dB	LPCC	47.52
	MFCC	68.32

6. Conclusion

From the above experiments, it has been observed that both MFCC and LPCC along with its 1st order derivatives can work as efficient parameterization of the speech signal for the recognition of the phonemes of Assamese language using MLP based recognizer. However, the performance of the system degrades considerably with the change in the training and testing conditions. It has been observed that under same environmental condition, when different set of speaker is used for training and testing the MLP based recognition, LPCC feature vector gives a recognitionaccuracy of 94.23% whereas for MFCC the recognition accuracy is 89.14%. Thus LPCC appears to give better representation of the speaker independent contents of the speech signal whereas MFCC captures some of the speaker dependent properties of the speech signal along with the speech contents. However, in noisy condition it has been observed that MFCC based system gives a relatively robust performance compare to LPCC based system. At 20dB SNR level MFCC based system gives 97.03% recognition accuracy whereas under same conditions, the recognition accuracy for the LPCC based system is 73.76%, i.e. there is nearly 24% difference in recognition accuracy. The same trend has been observed in other two level of noise also.It has been observed that with increase in noise level the performance of the MFCC based system also degrades but the degradation in case of LPCC is sharply more than that of MFCC.

References

[1]. Picheny, M.; Nahamoo, D.; Goel, V.; Kingsbury, B.; Ramabhadran, B.; Rennie, S. J.; Saon, G.; , "Trends and advances in speech recognition," *IBM Journal of Research and Development* , vol.55, no.5, pp.2:1-2:18, Sept.-Oct. 2011

[2]. Mitra, V.; Hosung Nam; Espy-Wilson, C.Y.; Saltzman, E.; Goldstein, L.; , "Articulatory Information for Noise Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEETransactions on* , vol.19, no.7, pp.1913-1924, Sept. 2011

[3]. Lippmann, R.; Martin, E. and Paul, D.: Multi-Style Training for Robust Isolated-Word Speech Recognition, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 705-708, April 1987.

[4]. Technology Development for Indian Language, Department of Information Technology, http://tdil.mit.gov.in.

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

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

[7]. Kohonen, T., "*Self-Organizing Neural Networks - Recent Advances and Applications(Studies in Fuzziness and Soft Computing)*",Physica-Verlag HD , 2002.

[8]. Moosavi, SeyedVahid, and Qin Rongjun. "A New Automated Hierarchical Clustering Algorithm Based on Emergent Self Organizing Maps." *Information Visualisation (IV)*, 2012 16th International Conference on. IEEE, 2012.

[9]. Gavat, I., Valsan, Z. and Sabac, B., Combining Self-Organizing Map and Multilayer Perceptron in a Neural System for Improved Isolated Word Recognition. Communication98. 245-255, 1998.

[10]. Gelenb, E. (Eds.): Neural Network: Advances and Applications, North-Holland, New York, 1991.

[11]. Zeidenberg, M.: Neural Network Models in Artificial Intelligence, E.Horwood, London, 1990.