Robust Speech Recognition System Designed by Combining Empirical Mode Decomposition and a Genetic Algorithm

Dr.A Srinivasa Murthy
Guide, Associate Professor

Niranjana Holla V P
PG Student

Dept of Electronics & Communication Engg.
UVCE, Bangalore

Abstract: Speech recognition applications are becoming more and more useful nowadays. Various interactive speech aware applications are available in the market. But they are usually meant for and executed on the traditional general-purpose computers. With growth in the needs for embedded computing and the demand for emerging embedded platforms, it is required that the speech recognition systems are available on them too. PDAs and other handheld devices are becoming more and more powerful and affordable as well. It has become possible to run multimedia on these devices. Speech recognition systems emerge as efficient alternatives for such devices where typing becomes difficult attributed to their small screen limitations.

Speech technology and systems in human computer interaction have witnessed a stable and remarkable advancement over last two decades. These technologies enable machines to respond correctly and reliably to human voices, and provide useful and valuable services. Recent research concentrates on developing system that would have been much robust against variability in environmental noise, speaker and language.

In this paper we mainly focus on robust speech recognition, and the environmental noise problem is its main concern. To accelerate the recognition speed we use discrete hidden Markov model to lessen the computation burden inherent in speech recognition.

Furthermore, the empirical mode decomposition is used to decompose the measured speech signal contaminated by several noise into several intrinsic mode functions (IMF's).The IMF's are weighted and summed to reconstruct the original speech signal, weights for each IMF's are obtained by the genetic algorithm to get optimal solution, by doing so we can achieve a better speech recognition rate for speech subjected to various environmental noise.

Keywords-Empirical Mode Decomposition (EMD), intrinsic mode functions (IMF's), Genetic algorithms (GA), Discrete Hidden Markov Model, MFCC, Vector quantization, LBG algorithm.

I. Introduction

Speech recognition has a long history. Yet, the speech recognition of speech subject to environmental noise remains an open problem. The most important problem in robust speech recognition is the mismatch problem that arises from the discrepancy between the testing and application environments concerning noise. Not surprisingly, there is a great deal of literature on this topic, e.g., [1]–[13]. Current methods for handling the mismatch problem can be classified into two categories, i.e., the feature- and model-based methods [7]. Feature-based methods focus on the feature parameters rather than on model parameters for speech or noise [2]–[7]. Model-based

methods exploit prior knowledge about the distributions of speech and noise for speech feature enhancement [8]–[13]. Since feature-based methods can work without prior knowledge of the distribution of speech and noise and are therefore suitable to applications in various environments, this paper focuses on the feature-based method. In this paper, speech signals with high noise interference will be processed to eliminate the noise components before capturing the speech features. In this way, the captured speech features become clearer. These speech features are then fed into the speech recognition system for recognition. A better recognition rate for those speech signals subject to noise interference can be then obtained.

To eliminate unwanted noise in speech signals, this paper applies the empirical mode decomposition (EMD) to decompose high-interference speech signals into several components, which will include either speech signals or noise. The EMD was first proposed by Prof. Huang in combination with the Hilbert transform (HT) to analyze nonlinear and nonstationary time series. The combination of the EMD and the HT is therefore referred to as the Hilbert–Huang transform [14]. The advantage of the EMD over other frequency-domain transformations is that the components decomposed from a mixed signal are related to specific physical sources. This allows us to examine the physical phenomena of a signal through the components obtained by the EMD. Initially, the EMD was applied to such things as signal analysis in the field of geoscience, strength analysis of material structures, trend analysis of the stock market, etc. More recently, the EMD has been applied in the measurement and the enhancement of speech signals [15], [16], and short circuit detection [17]. In [18], the EMD is used to separate audio sources from a single mixture. In that paper, an experiment is performed by mixing two specific audio sources into one and then reversing the process by separating the two audio sources from the mixed signal using the EMD. As discussed in the previous paragraph, the noise cancellation or suppression problem is critical to speech measurement [19]. Unfortunately, robust speech recognition for speech subject to various environmental noises is not well explored in those papers [15], [16], and [18]. According to the experiment in this paper, noise exists in some IMFs, which is neglected in those studies. Ignoring these IMFs will result in the loss of a part of the speech information and, consequently, an inferior recognition rate. In this paper, the original speech signals are recovered by combining more IMFs with different weightings to produce a better result. In addition, it is one of the goals of this paper to find the weights corresponding to different IMFs and combine these weighted IMFs to recover the original speech signals.

In this paper, the weights for each IMF are trained by the genetic algorithm (GA) to find an optimal combination of IMFs. The reason why the GA is used to train the weights to find an optimal combination of IMFs is the outstanding performance of the GA used in many research works (see, for example, [20] and the references therein). The experimental results in this paper will demonstrate that the proposed method based on the EMD outperforms the methods outlined in other literature.

The organization of this paper is as follows. In Section II, the general overview of the model for speech recognition is introduced. In Section III, the methods used in the model, including the training of codebook and the modeling of the DHMM by using speech features are investigated. Section IV describes the method of noise elimination by combining EMD and GA in the model. A

number of experiments that explore the relationship between the speech signals and the IMFs generated from the EMD are examined. Furthermore, the parameters used in the GA will be discussed in this section. Thereafter, the numerical results from the experiments carried out in the proposed speech recognition strategy are given in Section V, and the advantages of this paper are discussed and also show the experimental results of speech recognition for speech subject to various environmental noises. Finally, some conclusions are made in Section VI.

II Model for speech recognition

The model for speech recognition is shown in Fig. 1. In the proposed strategy, all noise-affected speech is first decomposed into several IMFs, using the EMD process. Since each IMF contains a greater or lesser speech signal, these IMFs are then weighted by their corresponding weights and then summed to recover the original speech signal. It is noted that the weights are initially randomized and will be trained by the GA thereafter. The MFCC process is then performed on the recovered speech to extract its features. In the training phase of this system, the speech features are used to train the codebook for the modeling of the DHMM, whereas in the testing phase, the speech features are fed into the DHMM for recognition. Also, in the testing phase, the GA is used to train the weights to get an acceptable recognition rate. The stop criterion of the GA depends on the recognition rate and the number of generations evolved in the GA. All the details of the procedures in the model will be discussed later.

III Modeling DHMM

This section introduces the modeling of the DHMM. First we extract feature vectors from speech signal to be trained and tested using MFCC process, a codebook for feature vectors are generated using vector quantization by LBG algorithm. Thereafter, a DHMM is modeled for speech recognition by using training speech features through the codebook.

A.MFCC Process

The purpose of this module is to convert the speech waveform to some type of parametric representation for further analysis and processing. This is often referred to as the signal-processing front end.

The LPC features were very popular in the early speech-identification systems. However, comparison of two LPC feature vectors requires the use of computationally expensive similarity measures such as the Itakura-Saito distance and hence LPC features are unsuitable for use in real-time systems. Furui suggested the use of the Cepstrum, defined as the inverse Fourier transform of the logarithm of the magnitude spectrum, in speech-recognition applications. The use of the cepstrum allows for the similarity between two cepstral feature vectors to be computed as a simple Euclidean distance. Furthermore, it has demonstrated that the cepstrum derived from the MFCC features rather than LPC features results in the best performance in terms of FAR [False Acceptance Ratio] and FRR [False Rejection Ratio] for speech recognition.

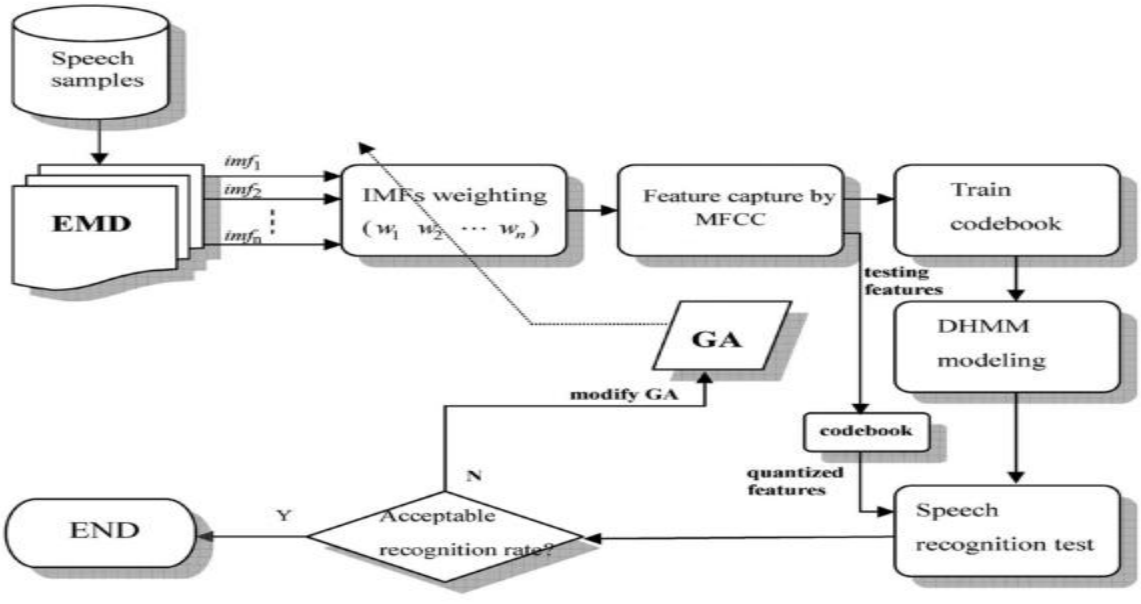

Fig.1: Model for speech recognition system.

Fig.2 shows the block diagram of MFCC process. The speech samples are first preemphasized then windowed to do further short term frame analysis on speech samples. The framed speech samples are sent through DFT to extract speech features further passed through various MEL filter banks to separate various frequency components, Mel filtered output is passed on to DCT to obtain Mel cepstrum coefficients.

A.1.Pre-Emphasis:

The pre-emphasis unit is used to boost the energy of high frequency components present in the speech signal. A special high pass filter is used for achieving this purpose. The spectrum of voiced segments of speech signal (speech signal is categorized into voiced and unvoiced segments) shows high levels of energy at lower frequency components and low levels of energy at high frequency regions. The Pre-Emphasis Filter is a First Order High Pass Filter & the Difference Equation is given by

$$y[n] = x[n] - \alpha.x[n-1]$$

Where,α is the gain controlling parameter for which the range is given by $0.9<\alpha<1$ and the System Function is given by

$$H[z] = 1 - \alpha.z^{-1}$$

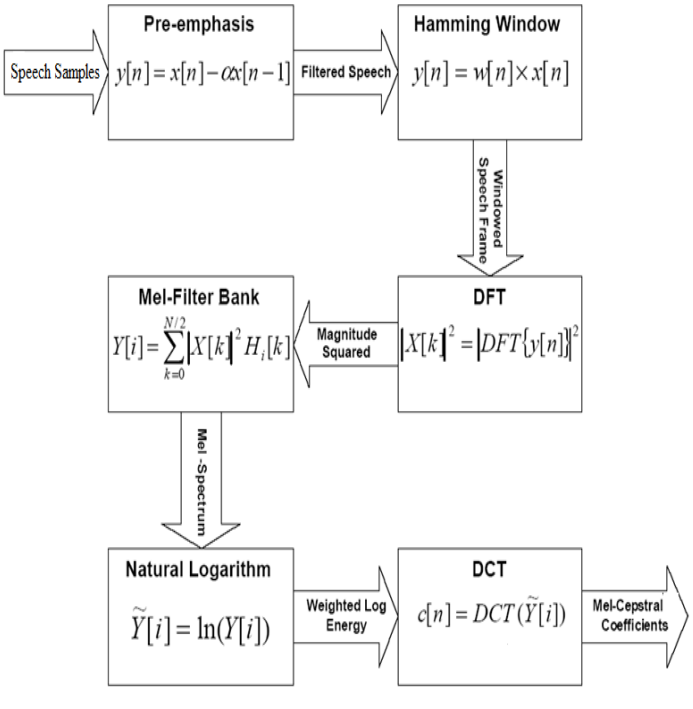

Fig.2: Block diagram of MFCC process.

A.2.Frame Blocking and Windowing:

The next step to divide the obtained signal from pre-emphasis into speech frames and apply a window to each frame. The speech signal is framed in order to capture the time characteristics of the speech signal. In order to minimize the signal discontinuities at the beginning and end of each frame is multiplied by an appropriate window function. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as,

w(n) ,0<n<N-1

Where *N* is the number of samples in each frame, then the result of windowing is the signal.

$$y_l(n) = x_l(n)w(n), \quad 0 \le n \le N-1$$

Typically the ***Hamming*** **window** is used, which has the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N$$

A.3.Discrete Fourier transform (DFT):

In order to derive the spectral information of the signal we use Discrete Fourier Transform. The DFT converts each frame of N samples from the time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples $\{x_n\}$ as follows

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

Denoting $W_n = e^{-j2\pi/N}$, the DFT can be expressed as

$$X[k] = \sum_{n=0}^{N-1} x[n](W_n)^{kn}$$

A.4.Mel filter bank:

As from perception experiments, 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:

$$f_{mel}(f) = 2595 \cdot \log\left(1 + \frac{f}{700Hz}\right)$$

The input-output relation of MEL Filter bank is given below.

$$Y[i] = \sum_{k=0}^{N/2} |X[k]|^2 H_i[k] \qquad i=0, 1 \ldots L-1$$

Where, Y[i] =MEL Filter Bank Output signal.

|X[k] |= DFT samples of the signal from the DFT block.

Hi[k] = The frequency response of the Lth filter in the MEL filter bank.

L = The number of MEL in the filter bank.

N = Number of the DFT samples.

The human auditory system is less sensitive to the signal level variations of the high amplitude signal and more sensitive to the low amplitude signal variations. This characteristic of the human hearing system is taken into account in the Mel filter bank itself for the better acoustic model of the speech signal. The absolute values of the spectral components from each of the MEL filter bank are squared before their

logarithmic (natural logarithm) values are found. The squaring of the spectral values indicates that the input to the next block contain power/energy spectral values.

A.5.Discrete Cosine Transform-DCT

In this final step, we convert the log mel spectrum back to in to cepstral domain. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis.

Therefore if we denote those mel power spectrum coefficients that are the result of the last step are

$$\tilde{Y}[i] = \ln(Y[i])$$

We can calculate the MFCC's, as

$$c[n] = DCT(\tilde{Y}[i])$$

$$c[n] = \sum_{k=1}^{K} (\log \tilde{Y}[i]) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K} \right], n = 1,2,3...K$$

Note that we exclude the first component c[0] from the DCT since it represents the mean value of the input signal which carried little speaker specific information.

By applying the procedure described above, for each speech frame of around 30msec with overlap, a set of mel-frequency cepstrum coefficients is computed. These are result of a cosine transform of the logarithm of the short-term power spectrum expressed on a mel-frequency scale. This set of coefficients is called an acoustic vector. Therefore each input utterance is transformed into a sequence of acoustic vectors. In the next section we will see how those acoustic vectors can be used to represent and recognize the voice characteristic of the speech.

B.Vector Quantization and Codebook:

Through preprocessing and feature extraction, the feature vectors for each speech signal frame are obtained. Because the feature vectors are real number vectors, the vector quantization for these feature vectors is necessary to reduce the computation burden, and hence, a codebook is trained and used for feature vector quantization. The results of vector quantization are then a set of observation codes for DHMM modeling and speech recognition.

B.1.Code book design

Vector quantization of speech signals requires the generation of codebooks. The codebooks are designed using an iterative algorithm called Linde, Buzo and Gray (LBG) algorithm. The input to the LBG algorithm is a training sequence. The training sequence is the concatenation of a set MFCC vectors obtained from people of different groups and of different ages. In this paper the speech signals are obtained from TIMIT database (Digit0 to Digit9) is available for use in speech recognition.

The codebook generation using LBG algorithm requires the generation of an initial codebook, which is the centroid or mean obtained from the training sequence. The centroid obtained is then split into two centroids or codeword's using the splitting method. The iterative LBG algorithm splits these two codeword's into four, four into eight and the process continues till the required numbers of codeword's in the codebook are obtained.

C.Train DHMM:

Fig.3. illustrates the training process for the DHMM. First, matrices *A*, *B*, and π, which describe the DHMM and will be explained in the following text, are randomized at the initial setup. Next, the speech features are quantized through the trained codebook. The quantized features are then the observation of the DHMM. The corresponding probability of the observation can be found from the present values in *A*, *B*, and π. Using

these probabilities, we can run the Viterbi algorithm [25] to update matrices *A*, *B*, and π until the values in these matrix converge.
Matrices *A*, *B*, and π are now explained as follows. In a DHMM, the hidden states are always unobservable, whereas the outputs in each state are observable. Each hidden state has a probability distribution over the possible output tokens (i.e., the observation). Therefore, the sequence of output tokens generated by the DHMM gives some information about the sequence of states. For the purpose of clarification, the relation among the features of speech, the observation, and the hidden states of DHMM is depicted in Fig.6.

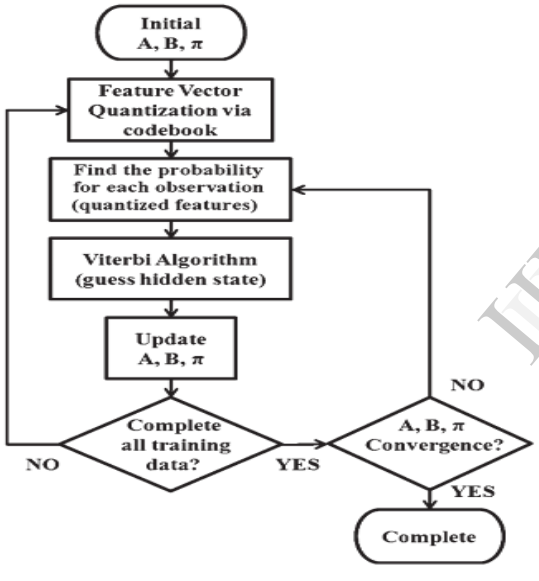

Fig.3.DHMM Training process

In the following, the definition and the detail training method of the parameters in the DHMM are introduced [25]. First, the definition of parameters in the DHMM is introduced.
$\boldsymbol{\lambda}$ DHMM. $\lambda = (A, B, \pi)$.
A $A = [aij]$. aij is the probability of state xi transferring to state xj , and $aij = P(qt = xj \mid qt-1 = xi)$.
B $B = [bj(k)]$. $bj(k)$ is the probability of kth observation, which is observed from state xj , i.e., $bj(k) = P(ot = vk \mid qt = xj)$.
$\boldsymbol{\pi}$ $\pi = [\pi i]$. πi is the probability of the case where the initial state is xi, and $\boldsymbol{\pi i} = P(q1 = xi)$.
X State vectors of DHMM. $X = (x1, x2, \ldots xN)$.
V Observation event vector of DHMM. $V = (v1, v2, \ldots, vM)$.
O Observation results of DHMM. $O = o1, o2, \ldots, oT$.
Q Resulting states of DHMM. $Q = q1, q2, \ldots, qT$.
To train the DHMM parameters $\lambda = (A, B, \pi)$ based on existing data, some notations are defined for convenience.
Eij Event of the transition from state xi to state xj .
Ei• Event of the transition from state xi to other states.
E•j Event of the transition from other states to state xj .
Ehi Event of state xi that appears at the initial state.
n(Eij) Number of the transition from state xi to state xj .
n(Ei•) Number of the transition from state xi to other states.
n(E•j) Number of the transition from other states to state xj .
n(E•j, o = vk) Number of enter to state xj and observation code is vk.
n(Ehi) Number of the event of state xi appears at the initial state.

In the process of training matrices *A*, *B*, and π of the DHMM, the hidden states for each observation are first estimated through the initial *A*, *B*, and π values by using the Viterbi algorithm. Then, values $n(Eij)$, $n(Ei\bullet)$, $n(E\bullet j)$, $n(E\bullet j, o = vk)$, and $n(Ehi)$ are computed according to the estimated hidden states for the whole training data. Subsequently, the elements in matrices *A*, *B*, and π are updated as follows:

$$\overline{a_{ij}} = \frac{n(E_{ij})}{n(E_{I\bullet})}$$

$$\overline{b_j}(k) = \frac{n(E_{\bullet j}, o = v_k)}{n(E_{\bullet j})}$$

$$\overline{\pi_i} = \frac{n(E_{hi})}{n_{TD}}$$

Where *nTD* is the number of training data. Using these updated *A*, *B*, and π values, we run the Viterbi algorithm again. The aforementioned steps are repeated until matrices *A*, *B*, and π converge. The training process for the DHMM is then completed.
Subsequently, the procedure for recognizing speech is depicted in Fig. 4. In the training phase, the DHMMs corresponding to each speech are first trained by using the training speech features through a trained codebook. In the test phase, the feature of the tested speech will be derived first. Through the trained codebook, this feature is then quantized and becomes an observation of the DHMM. For each observation, the probabilities for all DHMMs are calculated. The speech corresponding to the DHMM with the greatest probability is then the recognized speech. During the speech recognition process, the probability of the observations according to model $\lambda = (A, B, \pi)$ is calculated by

$$P(O|\lambda) = \sum_{Q} P(O|Q,\lambda)P(Q|\lambda)$$
$$= \sum_{q_1 \ldots q_T} \pi_{q_1} b_{q_1}(o_1) \cdot a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T).$$

This equation enables us to evaluate the probability of observations *O* based on DHMM $\lambda = (A, B, \pi)$. However, the time taken to evaluate $P(O|\lambda)$ directly would be an exponential function of the observation number *T*. For this reason, the forward algorithm is applied to reduce the computation time and is described below.

C.1.The Forward Algorithm

The forward algorithm can be described in three steps, i.e., initialization, recursion, and termination. The details are listed below based on the aforementioned parameters defined for the DHMM and are depicted in Fig. 5.

Initialization: The initial intermediate probabilities $\alpha 1(i)$ for the first observation $o1$ are calculated at the beginning as follows:

$$\alpha_1(i) \equiv \pi_i b_i(o_1) \quad 1 \le i \le N.$$

Recursion: For each observation *ot*, *t* = 2, . . . , *T* , the partial probabilities $\alpha t(j)$ are calculated for each state, i.e.,

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}) \qquad 1 \le t \le T-1$$
$$1 \le j \le N$$

in which *j* is the index number for hidden states. In this step, we calculate the product of the observation probability of *ot*+1 and the sum over all possible routes to that state from the states in previous observation *ot*. Then, the recursion is performed by using these values from the previous time step.

Termination: Finally, we sum all the partial probabilities at the final time step *T* to obtain the final result as follows:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

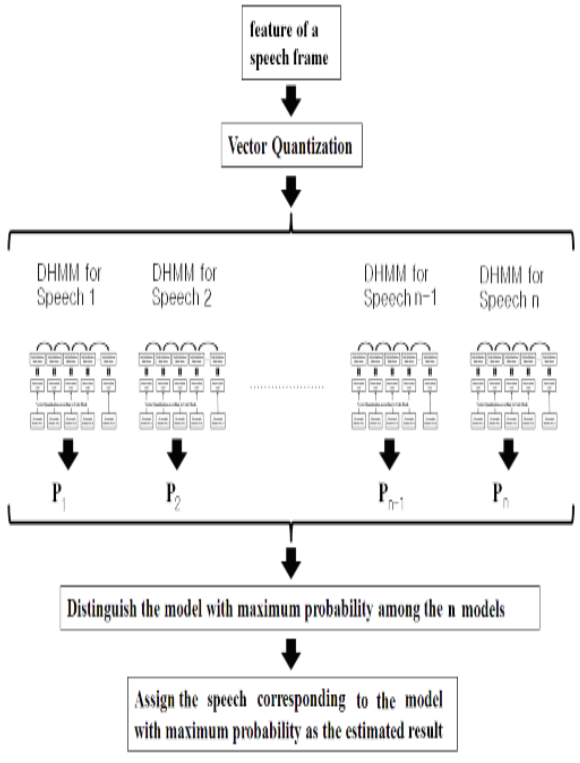

Fig.4. Procedure for speech recognition via the DHMM.

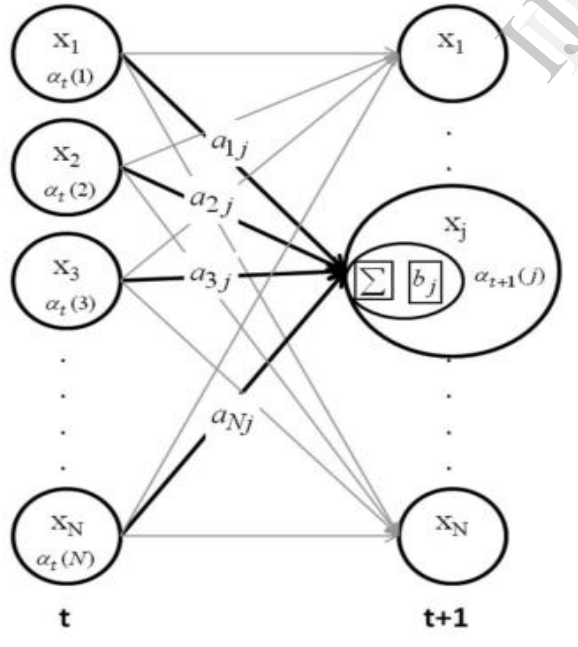

Fig.5. Illustration of the forward algorithm.

IV.EMD

This paper applies the EMD to decompose environmental noise and a clean speech signal from a contaminated speech signal. In this section, the procedure for performing the EMD is introduced. In addition, a strategy based on the GA and the EMD for robust speech recognition is proposed.

A. Procedure for EMD Operation

The purpose of the EMD is to decompose any multicomponent signal into a set of nearly monocomponent signals, referred to as intrinsic mode functions (IMFs).

The EMD is developed on the assumption that any signal consists of many different IMFs. Consequently, this project performs EMD operations to divide a speech signal into several IMFs.

Once the IMFs are obtained from a signal, the instantaneous frequency of each IMF can be then determined. Physically, the necessary conditions to define a meaningful instantaneous frequency are such that the inspected signal must be symmetric with respect to the local zero mean and have the same number of zero crossings and extrema. Hence, the condition for the data series to be an IMF can be described below

1) In the whole data series, the number of local extremes and the number of zero crossings must either be equal or differ at most by one.

2) At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

According to the aforementioned definition, an overview of the EMD can be now described in the following steps. First, identify all the local extrema from the given signal, and then, connect all the local

extrema with some cubic-spline lines to produce the upper and lower envelopes. The upper and lower envelopes should cover the entire signal between them. Then, compute their mean and the difference between the signal and the mean as the first component $h(t)$. Consider $h(t)$ as a temporary signal, and repeat the aforementioned steps until the difference between the temporary signal and the mean satisfy the two conditions of the IMF. Thus, the first IMF is obtained. Find the residue of the signal by subtracting the first IMF from the original signal. Define the residue as a new signal, and repeat the same aforementioned steps. We can then find the remaining IMFs for the original signal.

However, it is hard to satisfy the second condition of the IMF in a practical application, since a zero mean value of the envelopes for all time t is almost impossible. Hence, a looser condition is utilized to replace the second condition of the IMF. An index for the mean value of the envelopes and a threshold are used to construct this looser condition. The index can be calculated through the following equation

$$\mathrm{SD}_{ik} = \frac{\sum_{t=0}^{T} \left| h_{i(k-1)}(t) - h_{i(k)}(t) \right|^2}{\sum_{t=0}^{T} h_{i(k-1)}^2(t)}$$

in which $hi(k)(t)$ is the kth iteration for the ith IMF. It is noted that function $hi(k-1)(t) - hi(k)(t)$ in the numerator of above equation is equal to mean $mi(k)(t)$, i.e., $hi(k)(t) = hi(k-1)(t) - mi(k)(t)$.

This means that SDik is the ratio of the energy of $mi(k)(t)$ to that of $hi(k-1)(t)$. In practical applications, the second condition of the IMF is then replaced by the looser condition that SDik should be smaller than the assigned threshold. In general, the threshold is assigned in the range of 0.2–0.3. That is, if SDik < 0.3, for example, and the first condition of the IMF is satisfied, then the iterations for the ith IMF stops, and we get a new IMF.

The detailed procedure of the EMD for a data series or a signal is introduced as follows. It is noted that the cubic spline is used to generate the upper and lower envelopes of the signals during the process of the EMD.

Let the original signal be $X(t)$ and a temporary signal $\mathrm{Temp}(t) = X(t)$

Step 1: Find the upper envelope $U(t)$ and the lower envelope $L(t)$ of signal $\mathrm{Temp}(t)$. Calculate the mean of the two envelopes $m(t) = [U(t) + L(t)]/2$. Component $h(t)$ of $\mathrm{Temp}(t)$ is obtained by $h(t) = \mathrm{Temp}(t) - m(t)$.

Step 2: Check whether signal $h(t)$ satisfies the conditions of the IMF or not. If it does, then the first IMF is obtained as $\mathrm{imf1}(t) = h(t)$ and proceed to the next step, or else, assign signal $h(t)$ as $\mathrm{Temp}(t)$ and go back to step 1.

Step 3: Calculate residue $r1(t)$ as $r1(t) = \mathrm{Temp}(t) - \mathrm{imf1}(t)$. Assign signal $r1(t)$ as $X(t)$, and repeat steps 1 and 2 to find $\mathrm{imf2}(t)$.

Step 4: Repeat step 3 to find the subsequent IMFs as follows

$r_n(t) = r_{n-1}(t) - \mathrm{imf}_n(t) \qquad n = 2, 3, 4, \ldots$

This step is completed when signal $r_n(t)$ is constant or is a monotone function.

After the EMD procedure steps 1–4 are finished, the following decomposition of $X(t)$ is obtained

$$X(t) = \sum_{i=1}^{n} \mathrm{imf}_i(t) + r_n(t).$$

Thus, a decomposition of the data into n-empirical modes is achieved, and a residue $r_n(t)$ obtained which can be either the mean trend or a constant.

V. Numerical Results

During training phase, fifty speech utterances for each representing English digit zero to nine spoken by men and women are used for training purpose. This means there are total of 1000 speech utterances for this experiment. DHMM generates 10 probability models for 1000 speech utterances through codebook generated by preprocessing to represent each digit.

During testing phase one speech utterance from each digit is used. Prior to the experiment, white noise was added to the clean speech utterance to be tested. The noisy speech utterance is then submitted to the designed speech recognition system for robust speech recognition. The strength of the added noise is indicated by the SNR in decibels. Noisy speeches with five different SNRs are first decomposed into IMFs, these IMFs are then trained by the GA to find the best weighting. Based on these weights, the speech is recovered by summing the weighted IMFs. The DHMM produces probability model for test speech utterance recovered and then compared with the 10 probability model obtained during training phase to distinguish the model with maximum probability to recognize speech utterance under test. Experimental results of speech recognition rates for speech, with different SNRs are illustrated in TABLE 1. Tabulated result is obtained by testing 10 speech utterances representing English digit 0-9 spoken by men and women for different SNR level with EMD & GA and also without EMD&GA manually.

VI. Conclusion

A strategy for robust speech recognition has been proposed in this project. In the proposed strategy, the EMD has been applied to noisy test speech utterance (representing one of the English digit 0-9 spoken by men or women) for decomposing into several IMFs. The GA has been used to train the weights for the IMFs obtained from the EMD to obtain the best recognition rates. The test speech recovered by summing the weighted IMFs is then used to train the codebook. Thereafter, the features of the recovered test speech signal have been used to model the DHMM through the codebook. The probability model obtained by DHMM for test speech utterance is then compared with the 10 probability model each representing a digit obtained during training phase to distinguish the model with maximum probability to recognize speech utterance under test.

According to the numerical result tabulated in TABLE1, the proposed strategy performs well on improving the speech recognition rates for any speech that is subject to white noise of various strengths.

SNR	∞dB		40dB		30dB		20dB		10dB	
COMMENT	RECOGNITION RATE %									
	Without EMD GA	With EMD GA	Without EMD GA	With EMD GA	Without EMD GA	With EMD GA	Without EMD GA	With EMD GA	Without EMD GA	With EMD GA
TESTED WITH MEN UTTERENCES	60	90	30	100	10	90	10	70	10	40
TESTED WITH WOMEN UTTERENCES	90	100	30	90	10	90	10	70	10	30

TABLE1: RECOGNITION RATES FOR DIFFERENT NOISE LEVELS

VII. References

[1] D. Wang, H. Leung, A. P. Kurian, H. J. Kim, and H. Yoon, "A deconvolutive neural network for speech classification with applications to home service robot," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 12, pp. 3237– 3243, Dec. 2010.

[2] L. Buera, A. Miguel, O. Saz, A. Ortega, and E. Lleida, "Unsupervised data-driven feature vector normalization with acoustic model adaptation for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 296–309, Feb. 2010.

[3] J. W. Hung and W. H. Tu, "Incorporating codebook and utterance information in cepstral statistics normalization techniques for robust speech recognition in additive noise environments," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 473–476, Jun. 2009.

[4] L. D. Persia, D. Milone, H. L. Rufiner, and M. Yanagida, "Perceptual evaluation of blind source separation for robust speech recognition," *Signal Process.*, vol. 88, no. 10, pp. 2578–2583, 2008.

[5] J. Kim and B. J. You, "Fault detection in a microphone array by intercorrelation of features in voice activity detection," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2568–2571, Jun. 2011.

[6] Y. Zhan, H. Leung, K. C. Kwak, and H. Yoon, "Automated speaker recognition for home service robots using genetic algorithm and Dempster–Shafer fusion technique," *IEEE Trans. Instrum. Meas.*, vol. 58 ,no. 9, pp. 3058–3068, Sep. 2009.

[7] C. T. Ishi, S. Matsuda, T. Kanda, T. Jitsuhiro, H. Ishiguro, S. Nakamura, and N. Hagita, "A robust speech recognition system for communication robots in noisy environments," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 759–763, Jun. 2008.

[8] C.W. Hsu and L. S. Lee, "Higher order cepstral moment normalization for improved robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 2, pp. 205–220, Feb. 2009.

[9] A. Sankar and C. H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.

[10] C. H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Commun.*, vol. 25, no. 1–3, pp. 29–47, Aug. 1998.

[11] M. J. F. Gales and S. J. Young, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Computer. Speech Lang.*, vol. 9, no. 4, pp. 289–307, Oct. 1995.

[12] Y. Tsao and C. H. Lee, "An ensemble speaker and speaking environment modeling approach to robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 5, pp. 1025–1037, Jul. 2009.

[13] S. Windmann and R. Haeb-Umbach, "Parameter estimation of a statespace model of noise for robust speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1577–1590, Nov. 2009.

[14] N. E. Huang, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. R. Soc Lond. A*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[15] X. Li, X. Zou, R. Zhang, and G. Liu, "Method of speech enhancement based on Hilbert-Huang transform," in *Proc. 7th World Congr. Intell. Control Autom.*, Jun. 25–27, 2008, pp. 8419–8424.

[16] W.Wang, X. Li, and R. Zhang, "Speech detection based on Hilbert–Huang transform," in *Proc. 1st Int. Multi-Symp. Comput. Comput. Sci.*, Jun. 20–24, 2006, vol. 1, pp. 290–293.

[17] J. A. Rosero, L. Romeral, J. A. Ortega, and E. Rosero, "Short-circuit detection by means of empirical mode decomposition and Wigner–Ville distribution for PMSM

running under dynamic condition," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4534–4547, Nov. 2009.
[18] M. K. I. Molla and K. Hirose, "Single-mixture audio source separation by subspace decomposition of Hilbert spectrum," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 15, no. 3, pp. 893–900, Mar. 2007.
[19] S.-T. Pan, "Evolutionary computation on programmable robust IIR filter pole-placement design," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 4, pp. 1469–1479, Apr. 2011.
[20] S. T. Pan, "Design of robust S. T. Pan, "Design of robust D-stable IIR filters using genetic algorithms with embedded stability criterion," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3008–3016, Aug. 2009.