A RESEARCH PAPER ON ISOLATED SPOKEN WORD RECOGNITION USING HIDDEN MARKOV MODEL

[1]Ratna Priya Kanchan,[2]Dr.Manoj Soni

[1]*Student, Dept. of MAE, IGDTUW,* [2]*Associate Professor, Dept. of MAE, IGDTUW*

ABSTRACT

The main aim of the project is to develop an isolated spoken word recognition system using Hidden Markov Model (HMM) and that will be implemented on a 3R robotic arm with a good accuracy at all the possible frequency range of human voice. Here different command words like left, right, ready, up, down etc. are recorded by the speaker which can be male or female and results are compared with different feature extraction methods. The spoken word recognition system is mainly divided into two major blocks. First part includes recording data base of the command signals and feature extraction of those recorded signals. Here we use Mel frequency cepstral coefficients and fundamental frequency as feature extraction methods. To obtain Mel frequency cepstral coefficients signal we have to go through the following: framing, applying window function, Fast Fourier transform, filter bank and then discrete cosine transform.

Keywords: *HMM, Feature Extraction, Word recognition*.

1. INTRODUCTION

Natural form of communication for human beings is human voice. It is a speech signal which contains a sequence of sounds. Speech is the ability to express thoughts and feelings by articulate sound. Voice signals are generated by nature. Since they are naturally occurring hence are random signals. There are several models put forth by researchers based on their perception of voice signal.

Speech recognition is a very challenging problem on which a lot of work has been done. Most of the successful results have been calculated and obtained using Hidden Markov Models explained by Rabiner in 1989 [1]. A speech recognizer would efficiently enable more efficient and accurate communication for everybody, but especially for children, analphabets and people with disabilities. A speech recognizer also acts as a subsystem in a speech-to-speech translator.

The isolated speech recognition system implemented during the project trains Hidden Markov Model for each to be recognized. The models are trained well with labeled training data, and the classification is performed by passing the features of each of the model and then selecting the best match.

Further the result will be implemented on a 3R robotic arm, which performs the work as directed. It is given commands such as “left, right, forward” and many more through a Bluetooth module and Arduino. The commands are spoken to an android application through which the robot listens and acts accordingly.

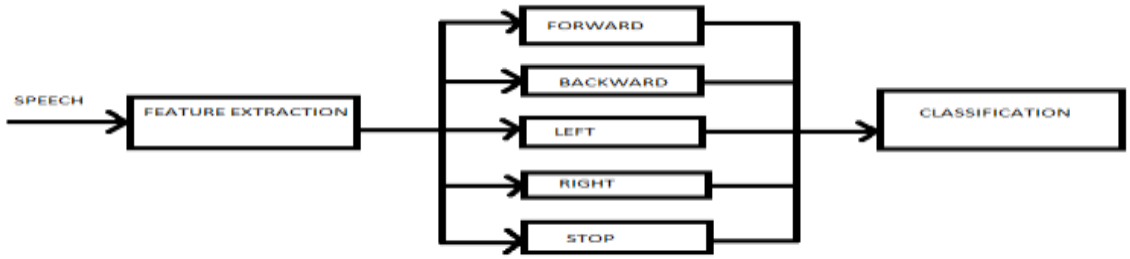

Figure 1.1: Flow chart of the system.

The features extracted from each of the speech signal are passed to each of the word model and then the best match is selected and worked on further.

2. BACKGROUND THEORY

2.1 Hidden Markov models

Hidden Markov Model is defined as a statistical finite state machine, in which the system being modeled is assumed to be in the Markov process. It consists of number of states to model the sequence of observation data. Here, the states are not visible directly but the output depending or corresponding the states are known to us. The Markov process states that the transition to the next state totally depends upon only on the current state and its output probability of the current state but not on all the previous states. Generally, the above process is called first order Markov process [4]. Supposing that the transition to next state depends on 'k' previous states, then the process is called the 'kth' order Markov process.

2.2 Isolated Spoken Word Recognition

The spoken word recognizer used, affectively maps the sequences of spoken vectors or so called observation vectors with the wanted symbol sequences that are given to be recognized [5]. The system implementation therefore becomes difficult with the two problems being identified and discussed further. Firstly, the mapping from symbols to speech is not a one-to-one, since different given symbols can give rise to the similar spoken sounds. Furthermore, there are large variations in the realized spoken word waveform due to speaker speech parameter variability, speaking mood, environment in which spoken, and also due to different speakers etc. Secondly, the boundaries between symbols cannot be recognized explicitly from the speech waveforms obtained.

On considering each spoken word that are to be represented by a sequence of speech vectors or observations, can be stated as below:

$$O = o_1, o_2, o_3 \dots \dots \dots, o_N \quad (1)$$

Where, O_t = the spoken word vector observed at time t

The problem of isolated word recognition can be stated as that of evaluating argument of the probability using the equation stated below:

$$arg\ max_i\{P(w_i|O)\} \quad (2)$$

Where, $w_i = i$th vocabulary word

The above probability cannot be calculated directly using the Bayes' Rule given by the expression stated below:

$$P(w_i|O) = \frac{P(O|w_i)P(w_i)}{P(O)} \quad 3$$

Thus, for a given prior set of probabilities $P(w_i)$, the most probable spoken word depends only on the happening of probability and can be given as $(O|w_i)$. For the stated dimensionality [6] of the observation vector, the sequence *O*, the direct evaluation that of the joint conditional probability, $P = (O_1, O_2, O_3 \dots \dots O_N|w_i)$ from examples of the spoken words is not achievable. Furthermore, if statistical finite machine like HMM is assumed, then evaluation from the data is possible since the problem of estimating the class conditional observation densities $P(O|w_i)$ gets replaced by much simpler problem of estimating the Markov model parameters[6].

In spoken word recognition that is based on HMM, it is quite understood that the sequence of observed spoken word vectors respective to each word are generated by a Markov model. HMM is defined as a finite state statistical machine [7] that keeps on changing state, for each time t that a state j is entered, a speech vector O_t is generated from the probability density $b_j(O_t)$. And furthermore, the transition from state i to state j is also probabilistic and is monitored by the discrete probability defined as a_{ij}. The joint probability O is generated [8] by the model M that is moving through the state sequence X, and is calculated simply as the product of the transition probabilities and the output probabilities of states. So, the state sequence can be described as given below:

$$X\ P(O, X|M) = a_{12} b_2(O_1) a_{22} b_2(O_2) a_{23} b_2(O_3) \quad 4$$

However, only the observation sequence O is known to us, i.e. visible and the underlying state sequence X is not visible and is hidden. Therefore called a Hidden Markov Model. For an unknown X, the required likelihood is generally evaluated by combining all the possible state sequences given as $= x(1), x(2), x(3), \dots \dots x(T)$, and thus can be explained as:

$$P(O|M) = \sum_X a_{x_0} x_1 \prod_{t=1}^{T} b_{x_t}(O_t) a_{x_t} x_{t+1} \quad 5$$

Where, $x(0)$ = constrained to be the model entry state of HMM

$x(T+1)$ = constrained to be the model exit state of HMM

An alternative to equation 5 can be given by a generalized evaluation of the likelihood that can be obtained by only considering the most likely state sequence given by:

$$\hat{P}(O|M) = max_X \left(\sum_X a_{x_0} x_1 \prod_{t=1}^{T} b_{x_t}(O_t) a_{x_t} x_{t+1} \right) \quad 6$$

However the evaluation of equations 5 and 6 is not practicable, only existing simple recursive procedures will allow both quantities to be evaluated very efficiently and effectively using recursion. The problem for isolated spoken word recognition can be resolved if equation 4 is being evaluated by Markov model. For a given set of models M with respect to words w_i, then equation 4 can be solved as:

$$P(O|w_i) = P(O|M_i) \quad 7$$

All of the above considers that the parameters of transition probabilities $\{a_{ij}\}$ and output probabilities $\{b_j(O_t)\}$ are known for each model M_i. For a given set of training sample, the spoken words with respect to a particular model, the parameters of which can be computed automatically by an effective and efficient re-estimation procedure known prior to evaluation. Thus, for a sufficient number of representatives, training samples of each word are collected and then a Hidden Markov Model (HMM) is developed which implicitly models all of the many parameters of variability given inherent in real speech [9]. An HMM is trained first for each vocabulary word using a number of examples of the word already given. In this case, the vocabulary consists of words: "forward", "backward", "left", "right", and "stop". Furthermore, to recognize some of the unknown words, the likelihood of each model that are generating that word is calculated first and then the most likely model captures the word [10] to be recognized.

2.3 The Forward Algorithm and Backward Algorithm

In hidden Markov Model, forward algorithm is generally used to calculate the belief state, the probability of a given state at a certain or a given or prescribed time. Forward algorithm is also known as a process called filtering. It is

very closely related to an algorithm known as Viterbi algorithm. It is used to solve the problems related to decoding. Generally, the calculation of joint probability is the goal of forward algorithm.

2.4 The Baum-Welch Algorithm

For calculating the parameters of HMM, the most accurate parameters i.e., in the maximum likelihood sense are found out using the Baum-Welch re-estimation method. The Baum-Welch algorithm is an iterative expectation-maximization (EM) algorithm that converges to a locally optimal solution from the initialization values.

Consideration of multiple data streams does not affect much since each of the streams is considered to be taken statistically independently. The transition probabilities are considered to be the mixture weights [11] that are viewed as a special form of sub-state of the mixture components. Therefore, the computation or estimation of the means and variances of HMM is of prime importance in which the output distribution for each state is a single component Gaussian, and is expressed as:

$$b_j(O_t) = \frac{1}{\sqrt{2\pi^n|\varepsilon_j|}} e^{\frac{-1}{2}(O_t-\mu_t)^T {\varepsilon_j}^{-1}(O_t-\mu_j)} \quad 8$$

If only one state i.e., j is considered in HMM, this estimation of parameter thus makes it easy. Thus the calculation of maximum likelihood estimates of μ_j and ε_j would be the averages that are given as:

$$\mu_j = \frac{1}{T}\sum_{t-1}^{T} O_t \quad 9$$

And,

$$\varepsilon_j = \frac{1}{T}\sum_{t-1}^{T}(O_t - \mu_j)\,(O_t - \mu_j)^T \quad 10$$

Although, there being many states and no direct alignment of observation vectors to the individual states are possible since the underlying state sequence is unknown. If approximate assignment of vectors to that of states is made, then equations 8 and 9 could be used to give the initial values for the given parameters. The training observation vectors are divided equally amongst the model states and then equations 8 and 9 are used to give initial values for mean and variance of each given state. Then the maximum likelihood state sequence is calculated using the Viterbi algorithm, therefore reassigning the observation vectors to states and then using equations 8 and 9 again so as to get better initial values than before. This process is repeated further until the estimates do not change and becomes accurate. The likelihood for each observation sequence is based completely on the summation of all the state sequences, and also each observation vector O_t contributes to the calculation of the maximum likelihood parameter values for each state *j*. Also, instead of assigning each observation vector to a particular state, each observation is assigned particularly to every state in proportion to the probability of the model being in that state on observing the vector. Thus, $L_j(t)$ denotes the probability of being in state *j* at time *t*, then the equations 8 and 9 stated above becomes in the following weighted averages form:

$$\mu_j = \sum_{t=1}^{T} \frac{L_j(t)O_t}{L_j(t)} \quad 11$$

$$\varepsilon_j = \frac{\sum_{t=1}^{T} L_j(t)\,(o_t-\mu_j)(o_t-\mu_j)^T}{\sum_{t=1}^{T} L_j(t)} \quad 12$$

Where, the summations in the denominators are used to give the required normalization.

Equations 11 and 12 are the Baum-Welch re-estimation formulae that are used for computing the means and covariance's of HMM. To apply equations 10 and 11, the probability of being in state $L_j(t)$ must be calculated beforehand. This is done efficiently and correctly using the Forward-Backward algorithm. Assuming the forward probability $\alpha_{j(t)}$ for some model M with N states is defined as:

$$\alpha_j(t) = P\big(O_1, \ldots\ldots O_2, x(t) = (j|M)\big) \quad 13$$

Let $\alpha_j(t)$ be the joint probability for observing the first t speech vectors and be in state *j* at time t. Thus, the forward probability can be efficiently and accurately calculated by the following recursion equation stated below:

$$\alpha_{j(t)} = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)\alpha_{ij}\right] b_j(O_t) \quad 14$$

The above recursion equation depends on the fact that the probability of being in state j at a given time *t* and on seeing the observation O_t can be deduced by summing all the forward probabilities for predecessor states *i* weighted by the transition probability given as α_{ij}. Odd limits are caused because states 1 and N are non-emitting. The initial conditions for the above given recursion are elaborated as below:

$$\alpha_{1(1)} = 1$$

$$\alpha_{j(1)} = \alpha_{1j} b_j(O_t) \quad 15$$

For 1<j<N and the final conditions are given by:

$$\alpha_N(t) = \left[\sum_{i=2}^{N-1} \alpha_i(T)\alpha_{iN}\right] \quad 16$$

The calculation for the forward probability also yields or gives the total likelihood $P(O|M)$.

The backward probability $\beta_j(t)$ is defined by the following equation:

$$\beta_j(t) = P(O_{t+1}, \ldots \ldots O_T | x(t) = j, M) \quad 17$$

As in the case of forward stated above, the backward probability can also be computed efficiently using the following recursion equation:

$$\beta_j(t) = \sum_{j=2}^{N-1} a_{ij} b_j(O_{t+1}) \beta_j(t+1) \quad 18$$

With the initial condition given as:

$$\beta_j(T) = \alpha_{iN} \quad 19$$

For 1<i<N and the final condition are given by:

$$\beta_1(t) = \sum_{j=2}^{N-1} a_{1j} b_1(O_1) \beta_j(1) \quad 20$$

The forward probability is defined as a joint probability whereas the backward probability is said to be a conditional probability [12]. This asymmetric definition allows the probability of state occupation to be calculated and determined by taking the product of the two probabilities i.e., forward as well as backward. From the above definitions following equations can be deduced:

$$\alpha_j(t)\,\beta_j(t) = P(O, x(t) = j|m) \quad 21$$

Hence,

$$L_j(t) = P(x(t) = j|O, m) \quad 22$$

$$= \frac{P(O, x(t)=j|m)}{P(O|M)}$$

$$= \frac{1}{P}\alpha_j(t)\,\beta_j(t)$$

Where, $P = P(O|M)$

All of the above information needed to perform HMM parameters [13] re-estimation calculation using the Baum-Welch algorithm are all at a place. The steps in the algorithm are summarized as follows:

- For every parameter vector or matrix requiring re-estimation, storage is allocated for the numerator and denominator and summations of the form are illustrated by equations 10 and 11. These storage locations are generally referred to as accumulators.
- The forward and backward probabilities are calculated for all states 'j' at time 't' [14].
- For each of the state j and time t, the probability $L_j(t)$ is used and the current observation vector O_t is used to update the accumulators for that particular state.
- The final accumulator values are used to calculate the new parameter values.
- If the value of $P = P(O|M)$ for the iteration is not higher than the value at the previous iteration then the process is stopped, otherwise the above steps are repeated using the new re-estimated parameter values.

All the above steps assumes that the parameters for HMM are re-estimated from a single observation or command sequence, that is a single word of the spoken word. However, the use of multiple observation sequences does not increase the complexity of the algorithm. Steps 2 and 3 stated above are repeated for each and every distinct training sequence. The computation of the forward and backward probabilities involves the calculation of the product of a large number of probabilities almost all the probabilities. In practice, this actually means that the actual numbers involved during the process becomes very small. Hence, the forward-backward computation is generally computed using the normal way.

3. SYSTEM DESIGN

3.1 Feature Extraction

The frame of the signal in time as well as in frequency domain and the hamming window used are shown in the plots given below.

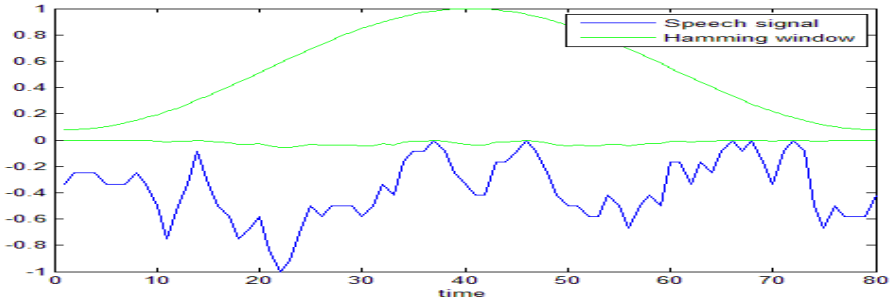

Figure 3.1: frame of a signal in time domain

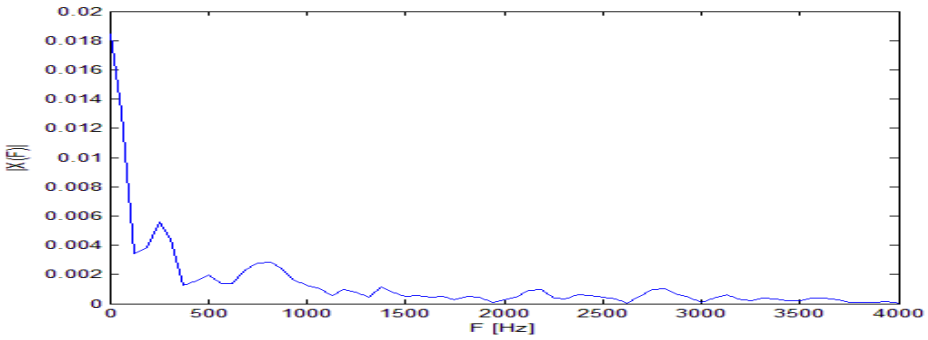

Figure 3.2: frame of a signal in frequency domain

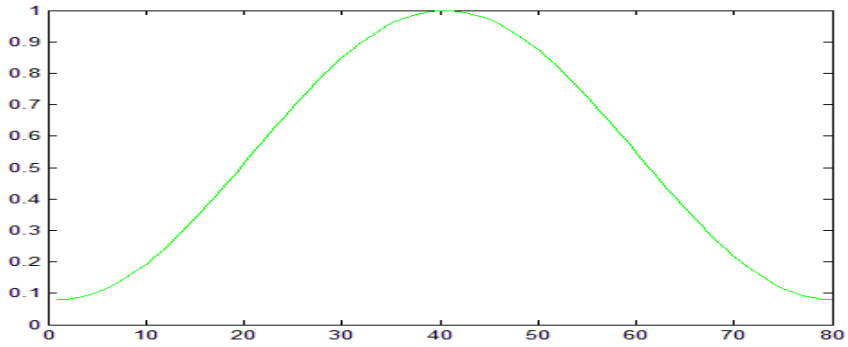

Figure 3.3: hamming window

4. TRAINING

The training constitutes both of supervised and unsupervised techniques. We train the hidden Markov model per utterance of the command. Each state in the word should represent a phoneme. The clustering of the Gaussians is unsupervised and depends on the initial values that are used for the Baum-Welch algorithm. The diagonal covariance matrix for the training of the data was used for all states. The training examples for each of the word are concatenated or collected together, and Baum-Welch is run for 15 iterations for each of the command. The waveform of few commands are shown as under.

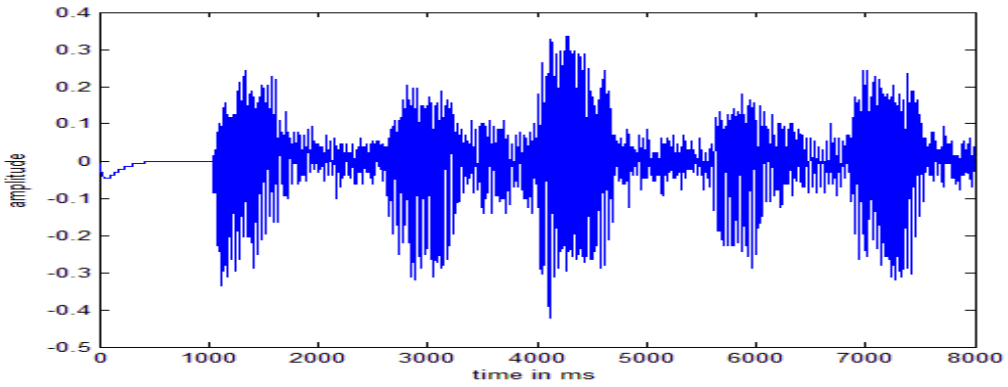

Figure 4.1: waveform of backward command

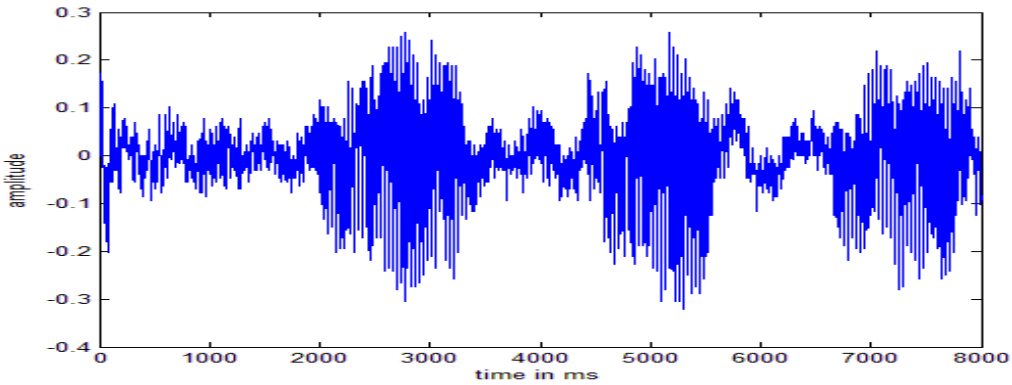

Figure 4.2: waveform of left command

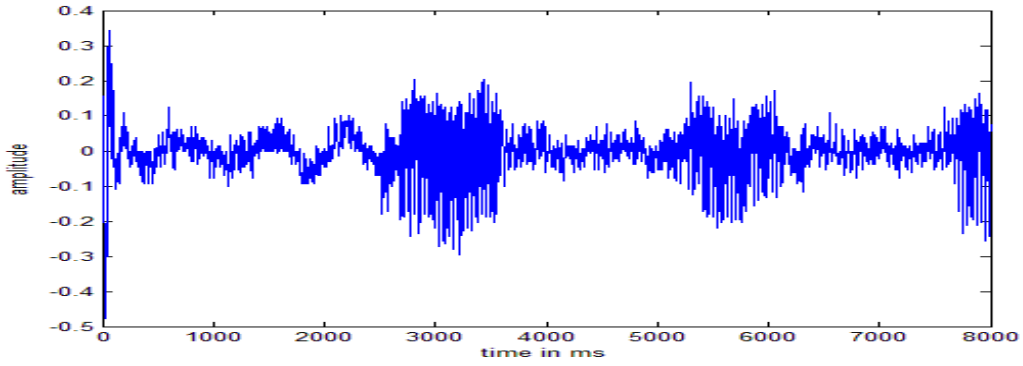

Figure 4.3: waveform of stop command

5. EXPERIMENTAL SETUP AND RESULTS

For each of the words, 15 utterances each were recorded. The performance of the system was measured. Experimentation indicated that, the two most important parameters were the number of hidden states, given by N, and also the number of frequencies extracted from each of the frame, D. The cross-validation was therefore run with different set of values for the parameters.

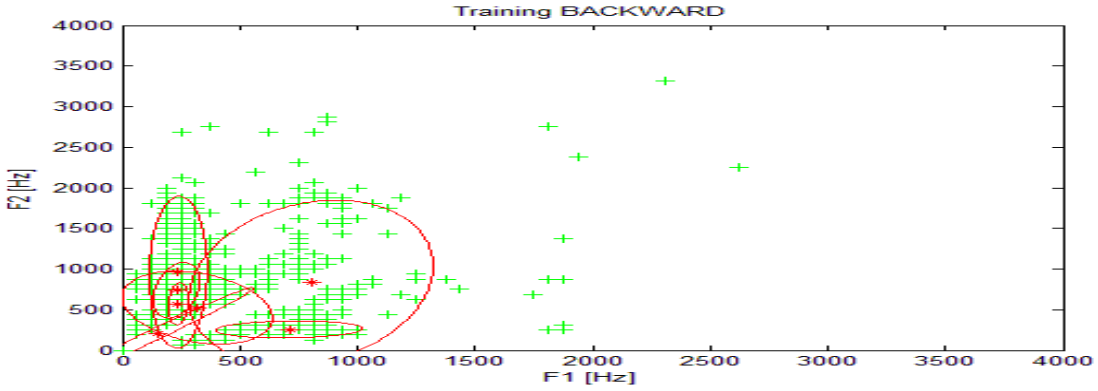

Figure 5.1(a): Training backward

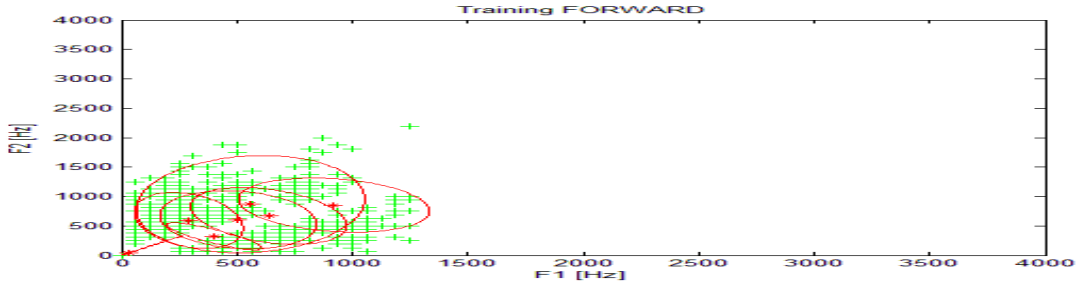

Figure 5.1(b): Training forward

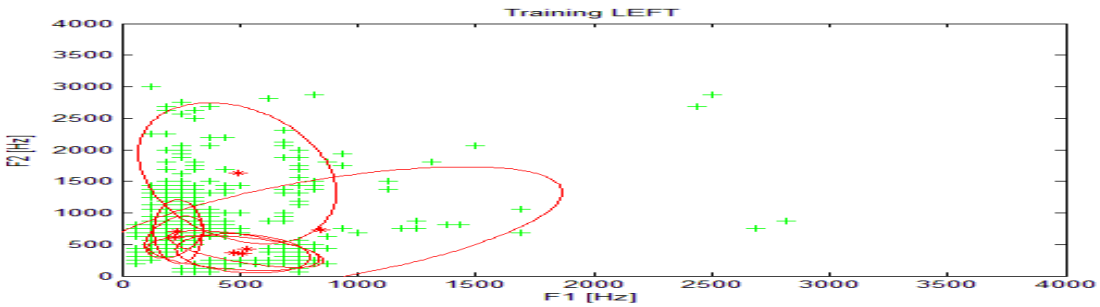

Figure 5.1(c): Training left

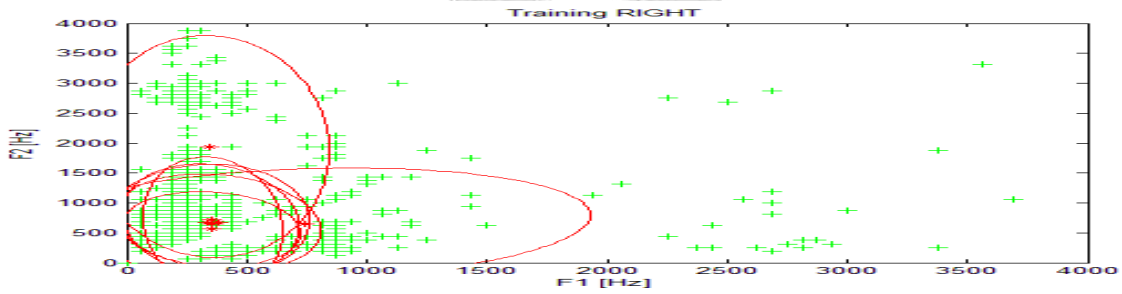

Figure 5.1(d): Training right

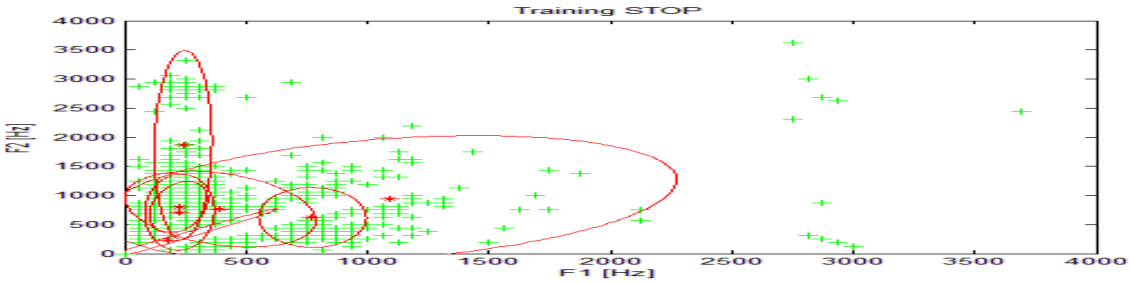

Figure 5.1(e): Training stop

Figure 5.1: HMM models of training various commands

The green plus sign here represents a frame from a training speech signal. The stars represents each Gaussian, and the ellipse indicates their confidence interval that is 75%.

6. DISCUSSION

The results obtained from this are quite good and almost accurate compared to the simple approaches taken previously, especially in the feature extraction phase. More advanced features such as Mel-frequency Cepstral coefficients were considered. The spoken commands were thus tested on a 3R robot which resulted in execution of the command. The execution time spent during feature extraction method and in training was more. The number of samples in each frame constitutes an important parameter. If the frame is small, it becomes very difficult to pick out meaningful features, and also if it is too large, most of the temporal information is lost. Rabiner gives a well modified Baum-Welch algorithm for the multiple training examples such that concatenation is not as such necessary, but that was not tested or implemented during the project as the concatenation seemed to work well with the project.

7. CONCLUSION AND FUTURE WORK

During the project, a system for isolated-word speech recognition will be implemented on a 3R robot and tested. The cross-validation results obtained were good. Two of the works are further possible i.e., better support for several speakers, and also for continuous speech. The step towards the former would be more highlighted and more robust features are to be opted. For the latter the simple approach is sufficient to detect word boundaries and then to proceed with an isolated-word recognizer.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, pp. 257{286, Feb 1989.

[2] C. M. Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics)", Springer, 1st ed. 2006. corr. 2nd printing ed., October 2007.

[3] X. Huang, A. Acero, and H.-W. Hon, Spoken Language Processing: "A Guide to Theory, Algorithm and System Development", Prentice Hall PTR, May 2001.

[4] Huang, X., Alex, A., and Hon, H. W. (2001). Spoken Language Processing; A Guide to Theory, Algorithm and System Development. Prentice Hall, Upper Saddle River, New Jersey.

[5] M. A. M. Abu Shariah, R. N Ainon, R. Zainuddin, and O. O. Khalifa, "Human Computer Interaction Using Isolated-Words Speech Recognition Technology," IEEE Proceedings of The International Conference on Intelligent and Advanced Systems (ICIAS'07),Kuala Lumpur, Malaysia, pp. 1173 – 1178, 2007.

[6] Jelinek F. "Statistical Methods for Speech Recognition", The MIT Press, Cambridge, Massachusetts.

[7] Rabiner, L. R. and Juang, B. (1993), Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, New Jersey.

[8] Ting, H. N., Jasmy, Y. and Sheikh, Hussain, S. S. "Speaker-Independent Malay Syllable Recognition Using Singular Multi-layer Perceptron Network." Proceedings of the 2002 International Conference on Artificial Intelligence in Engineering and Technology (ICAIET 2002), Kota Kinabalu (Malaysia), 17-18 June 2002.

[9] Dimov, D., and Azmanov "Experimental specifics of using HMM in isolated word speech recognition". International Conference on Computer Systems and Technologies Comp Sys Tech'2005.

[10] N. Najkar, F. Razzazi, and H. Sameti, "A novel approach to HMM-based speech recognition system using particle swarm optimization," in BIC-TA 2009 - Proceedings, 2009 4th International Conference on Bio-Inspired Computing: Theories and Applications, 2009, pp. 296-301.

[11] Antanas Lipeika, Joana Lipeikiene, Laimutis Telksnys, "Development of Isolated Word Speech Recognition System," Informatica 2002, Vol. 13, 37-46, 2002.

[12] Steve Young, Gunnar Evermann, Mark Gales "The HTK Book" Microsoft Corporation.

[13] I. Bazzi and J. R. Glass, "Modeling out-of-vocabulary words for robust speech recognition," in Proceedings of Int. Conf. Spoken Language Processing, Beijing, China, October 2000, pp.

[14] Armin Sehr, Roland Maas, and Walter Kellermann, "Reverberation Model-Based Decoding in the Logmelspec Domain for Robust Distant-Talking Speech Recognition", IEEE Transactions On Audio, Speech, And Language Processing, Vol. 18, No. 7, September 2010. 401–404.