PHASE BASED SPEECH RECOGNITION

BY

SRIDHAR POLINA - 031035
SRIKIRTI PUNUGU - 031053
RAJA BHARATH KUCHAMPUDI -031061

May 2007

Submitted in partial fulfillment of the Degree of Bachelor of Technology.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION, JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY-WAKNAGHAT.

CERTIFICATE

This is to certify that the work entitled, "Phase based speech recognition" submitted by Sridhar Polina ID 031035, Srikirti Punugu ID 031053, Raja Bharath.K ID 031061 in partial fulfillment for the award of degree of Bachelor of Technology in Electronics And Communications of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Date : 21/5/07

Signature :

(Dr.Sunil Bhooshan)
HOD, ECE Dept .

(Mr. Vinay Kumar)
Senior Lecturer, ECE Dept

ACKNOWLEDGMENT

It has been a great pleasure working under the able guidance of faculty and staff in the department of Electronics and Communication at Jaypee University of Information Technology, during our study as a B.Tech student. Thanks a lot to HOD of our department Dr. Sunil Bhooshan and our project guide Mr.Vinay Kumar for their technical and moral support, without whom this research work would ever have been possible.

We are indebted to all those who provided reviews and suggestions for improving the results and the topics covered in this project, and extend our apologies to anyone whom we may have failed to mention.

SRIDHAR POLINA (031035)
SRIKIRTI PUNUGU (031053)
RAJA BHARATH. K (031061)

TABLE OF CONTENTS

LIST OF ABREVIATIONS

1. LBG Algorithm ------ Linde – Buzo - Gray Algorithm.
2 HMM ----------------- Hidden Markov Model.
3 VQ -------------------- Vector Quantization.
4 SNR------------------- Signal to Noise Ratio.
5 ANN ------------------ Artificial Neural Networks

LIST OF FIGURES

ABSTRACT

In this information technology age, speech recognition plays a prominent role in making many of the activities easier as well as secured. As a result there has been many recognition algorithms developed all around considering the major recognizing attributes as amplitude or frequency.

Here we consider phase to be one of the attribute as well in contributing to the effective speech recognition.

In this report we start with the importance of phase in the speech recognition and processing systems. In the later stages we shall explain some steps that we have gone through in our modeling of the speech recognition process.

We have mainly divided the process into two parts. They are – Vector Quantization (VQ) and Artificial Neural Networks(ANN).

The algorithm that we have used for the VQ processing was the LBG algorithm. We shall even present the code that we have used in assigning the codeword for each vector and also developing a codebook.

We shall also give you a brief idea and introduction about ANN and the method of applying the ANN in speech recognition process.

INTRODUCTION

1.1 What is a speech signal

A speech signal is a sound wave containing information. Considering here that the sounds generated are only of the human and since each human has got a unique frequency and pitch characteristics.

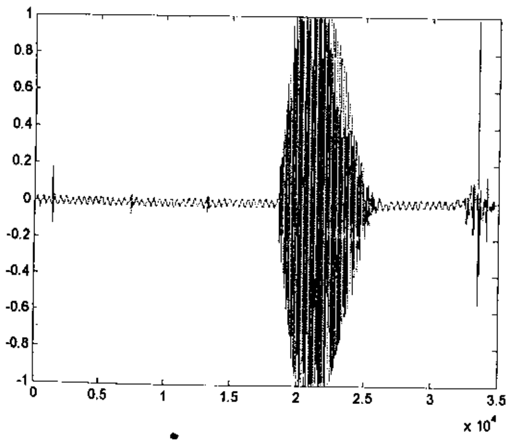

Fig 1

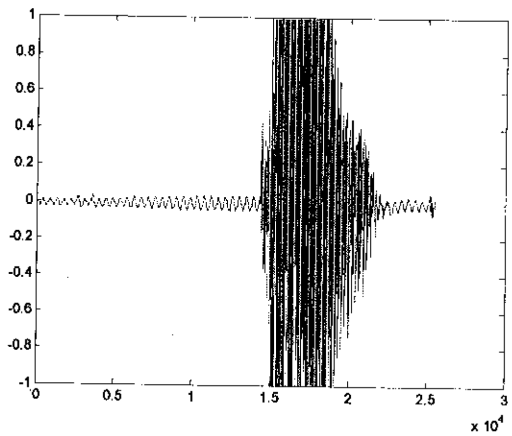

Fig 2

Fig 1 & Fig 2 – show the plot of the recorded speech signal of letter 'a' and letter 'k' with normalized amplitude as its y axis and time intervals as it x axis.

1.2 Attributes of a signal

The main attributes of a signal are Amplitude, Frequency and Phase .Current artificial speech recognition systems, utilize only the amplitude of the incoming sound waves at different frequencies in order to recognize speech. First of all, the use of spectral transformations for speech processing may not be the ideal representation, although they are almost universally used by all speech recognition systems. Second, the phases of the different frequencies are often tossed out without much attention or processing, open question in speech processing has been whether phase should play a more dominant role in the speech recognition process.

1.3 What is speech recognition?

Speech recognition is a process that attempts to identify the person speaking, as opposed to what is being said. It is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program. Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. They can also serve as the input to further linguistic processing in order to achieve speech understanding. Speech recognition applications that have emerged over the last few years include voice dialing, call routing, data entry, preparation of structured documents, demotic appliances control and content-based spoken audio search.

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy is measured with the word error rate, whereas speed is measured with the real time factor.

1.4Applications of speech recognition

* Automatic Translation
* Automotive speech recognition
* Command recognition - Voice user interface with the computer.
* Dictation
* Hands-free computing
* Home automation
* Interactive Voice Response
* Medical Transcription
* Mobile telephony
* Pronunciation evaluation in computer-aided language learning applications[1]
* Robotics

2. RECOGNITION PROCESSES

2.1 Conventional Recognition process

Current artificial speech recognition systems, for the most part, utilize only the amplitude of the incoming sound waves at different frequencies in order to recognize speech. Current state-of-the-art speech recognition system can achieve high recognition accuracy rates (>90%) in noise free environments. However, their performance significantly degrades in adverse noise situations. Since most environments do contain noise, a solution must be found to enable robust and accurate speech recognition in all practical solutions. Most state-of-the-art speech recognition systems only utilize the magnitude of the Fourier transform of the time-domain speech segments. This means that the corresponding Fourier transform phases are discarded. Several studies have indicated that it may be a fruitful effort to directly model and incorporate the phase into the recognition process.

2.2 Importance of phase in recognition

Every recorded signal can be expressed as a summation of sinusoids, These sinusoids would be of the form $A\cos(2\prod ft-\Phi)$, where A is the magnitude of the sinusoid, f is the frequency in Hz, and finally, Φ is the phase or the starting point of the sinusoid. Now, as it turns out, there are some characteristics of phase that make it very unique for speech processing applications. First of all, just like FM radio receivers the estimation of phase can often be done with more reliability than the estimation of amplitude. Another important aspect of phase is the timing information that it provides. Under mild conditions, a finite duration signal can be reconstructed to within a scale factor by its phase (where the phase is determined over the duration of the signal or on a short-time basis). This is not true for magnitude.

The degree of the importance of phase also seems to be an SNR-dependent one, such that at lower SNRs the effects of phase uncertainty are more pronounced than at higher SNRs. Results show that at higher SNR values phase is not as important for recognition of speech as it is at lower SNR values. The results indicated that human perception of phase varies with frequency, especially for low pitched speakers .It was shown that the phase spectra play an important role in specifying a stop consonant.

Assume that a sound signal consists of three sinusoids (one with frequency 110Hz) is recorded by a microphone for a total duration of 50ms.The recorded signal x (t) can be expressed as

$X(t) = \sin(2\Pi * 100t) + \sin(2\Pi * 105t) + \sin(2\Pi * 110t)$ as illustrated in the figures below.

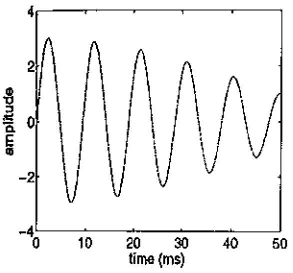

Fig – 2.1

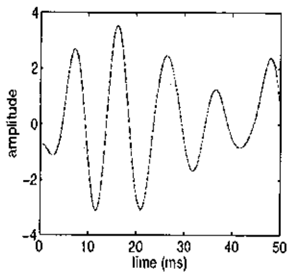

Fig – 2.2

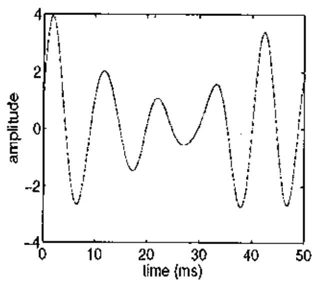

Fig-2.3

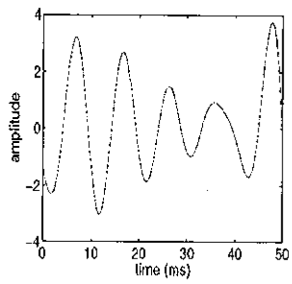

Fig – 2.4

After x (t) is transformed into the frequency domain, its phase is tossed away and only its spectral magnitude is kept. As a result of this, any of the time domain signals that are shown in figure 2.2, figure 2.3, figure 2.4. Clearly, these signals look different that the original signal. Yet, since the phase information is tossed away, there is no simple way to distinguish between the different possibilities. This ambiguity, of course, could be a significant problem when it comes time for speech recognition since most recognition systems do not utilize phase directly

3. PROJECT DESCRIPTION

3.1 Data Warehouse of speech signals

Initially we create a database for training our prototype and comparing and identifying a new signal as well. For this we have used a normal microphone to record the sample speech signals in the wav format.

Since we are making a user dependant system we have taken the samples from only one team member. And also to make things a bit simple we have considered identifying English alphabet 'a' to 'z' individually as our main goal. So, we have recorded ten samples for each English letter and made it as our database.

3.2 Plotting the speech samples

Having created a database of all the signals we thought of sampling and then plot all the samples that we have recorded for further use in extracting attributes of the speech signal.

Hence, for this purpose we have used MATLAB *7.0.1* software with the following steps.

i. Import the .wav file in the work directory of MATLAB.
ii. A pop up window showing the attributes to be plotted along with amplitude and the frequency like shown in the following figure 3.1 appears.
iii. Now deselect the frequency (*fs*) in the left side of the window, and then finish.
iv. The popup window disappears and then in the command window of the MATLAB, type in the command - plot *('file-name')*.
v. Hence, the plotted signal is shown in another window named figure which you can save it.

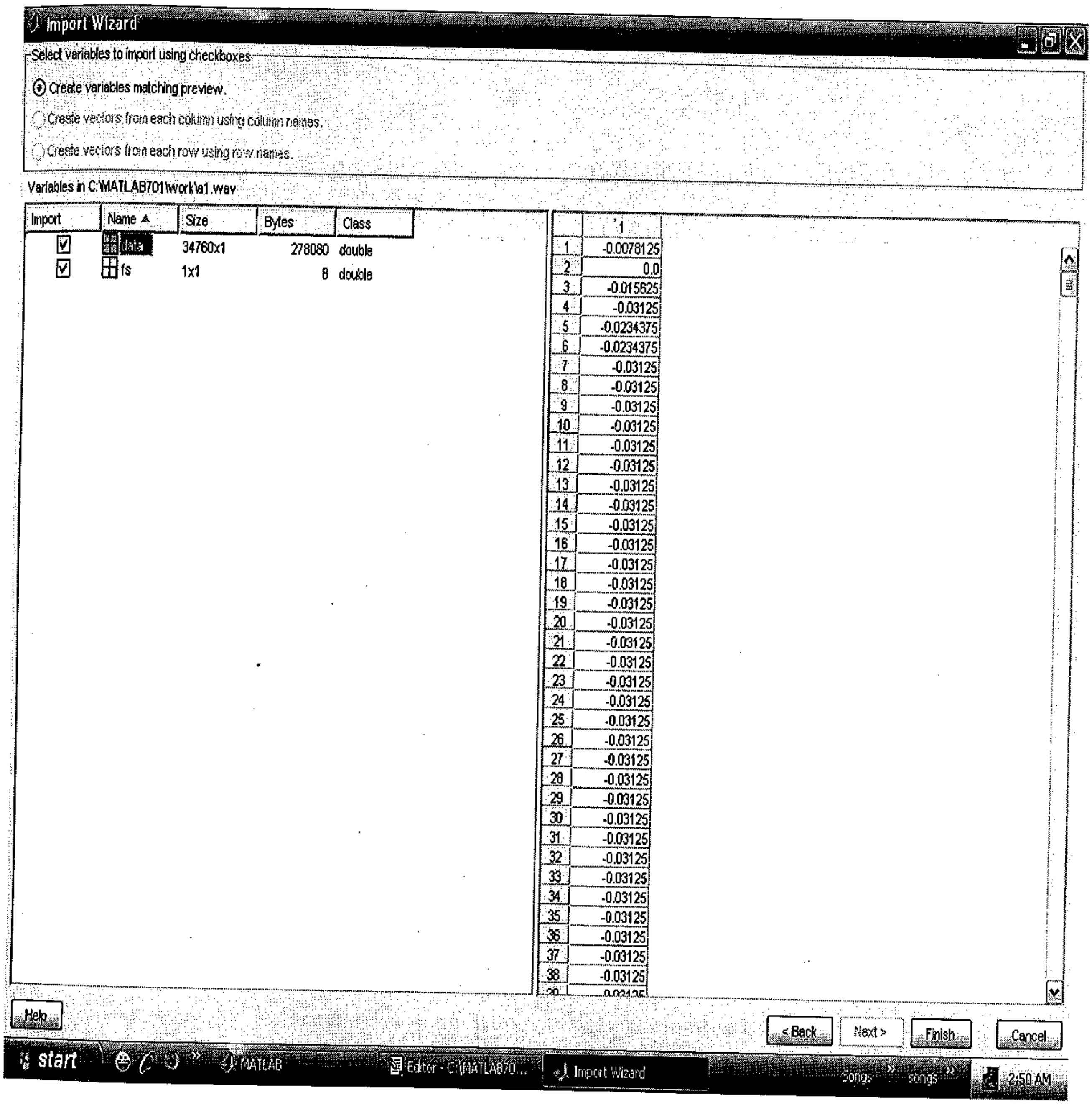
Import Wizard
Select variables to import using checkboxes
Create variables matching preview.
Create vectors from each column using column names.
Create vectors from each row using row names.
Variables in C:\MATLAB701\work\a1.wav
Import
Name
Size
Bytes
Class
data
34760x1
278080
double
fs
1x1
8
double
Help
< Back
Next >
Finish
Cancel
start
MATLAB
Import Wizard
2:50 AM

Fig – 3.1

3.3 Vector Quantization

Advances in coding theory suggest that optimal coding efficiency can be attained asymptotically as the no. of signal samples encoded simultaneously is increased. This motivated speech-coding researchers in the late 1970s and 1980s to explore the use of the methods of vector quantization.

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm.
A VQ is nothing more than an approximator. The idea is similar to that of ``rounding-off'' to the nearest integer incase of the example below. An example of a 1-dimensional VQ is shown below:

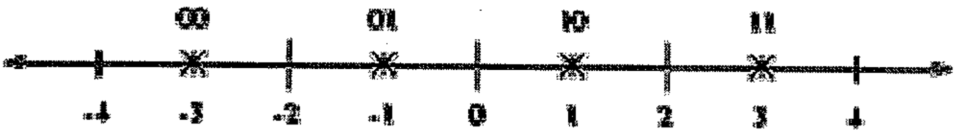

Fig – 3.2

Here, every number less than -2 is approximated by -3. Every number between -2 and 0 are approximated by -1. Every number between 0 and 2 is approximated by +1. Every number greater than 2 is approximated by +3.

Note that the approximate values are uniquely represented by 2 bits. This is a 1-dimensional, 2 -bit VQ. It has a rate of 2 bits/dimension.

Vector quantization aims at encoding an entire vector of samples or coefficients simultaneously. An example of a 2-dimensional VQ is shown below:

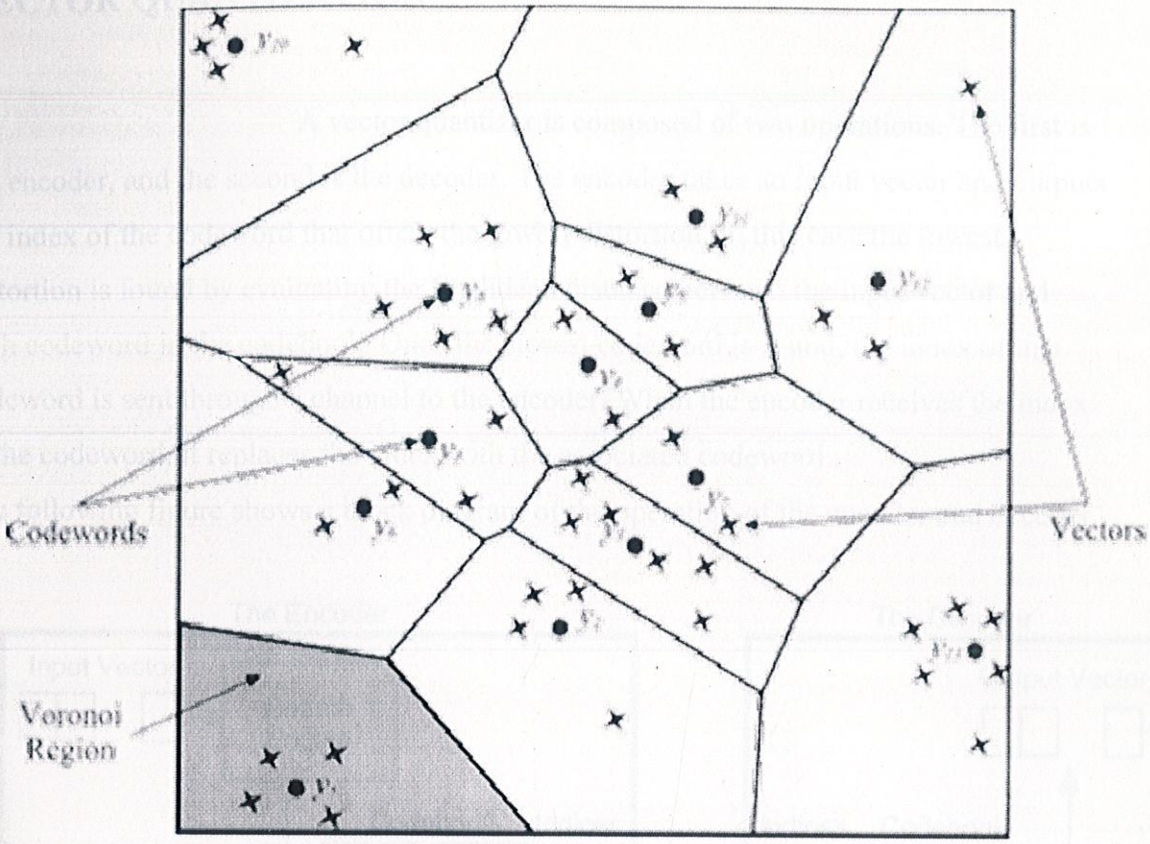

Fig – 3.3

3.4 Artificial Neural Network

An **artificial neural network** (ANN) or commonly just **neural network** (NN) is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

VECTOR QUANTIZATION

A vector quantizer is composed of two operations. The first is the encoder, and the second is the decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel to the encoder. When the encoder receives the index of the codeword, it replaces the index with the associated codeword.
The following figure shows a block diagram of the operation of the encoder and decoder.

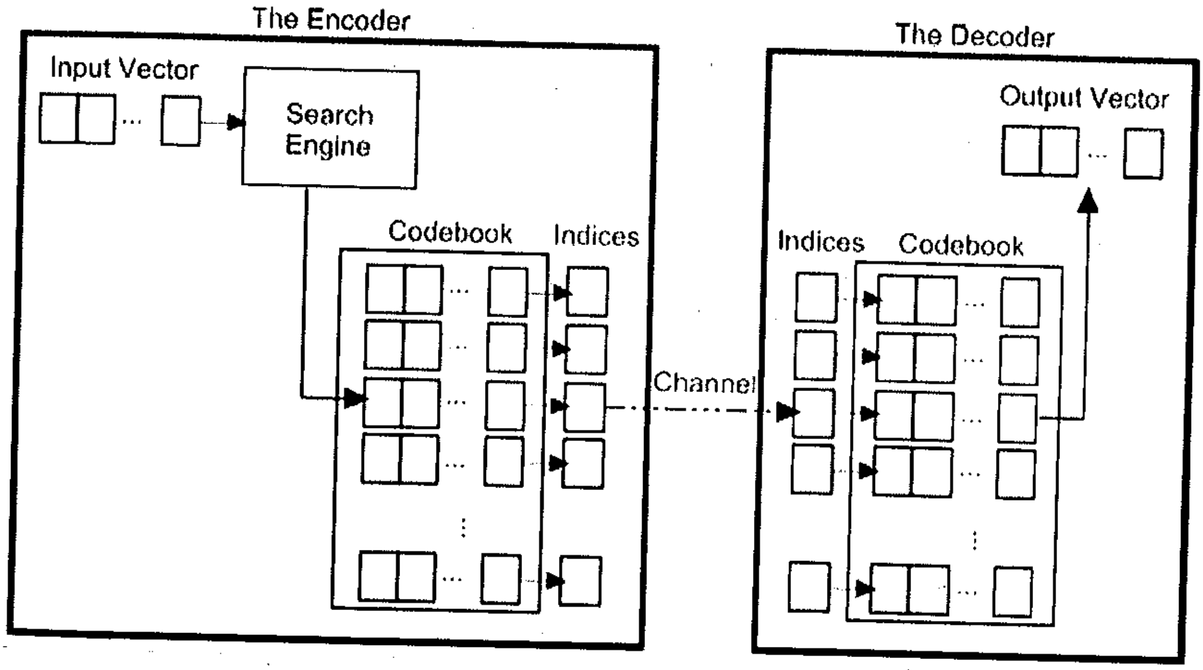

Fig – 4.1

VQ is just a pattern matching process. The key to high precision reconstruction quality is to have a codebook that matches the average pattern to be coded well. For this reason, codebook construction is a very important component of VQ.

4.1 LBG Algorithm

There are a number of ways to construct an initial codebook. We will use the LBG algorithm approach.

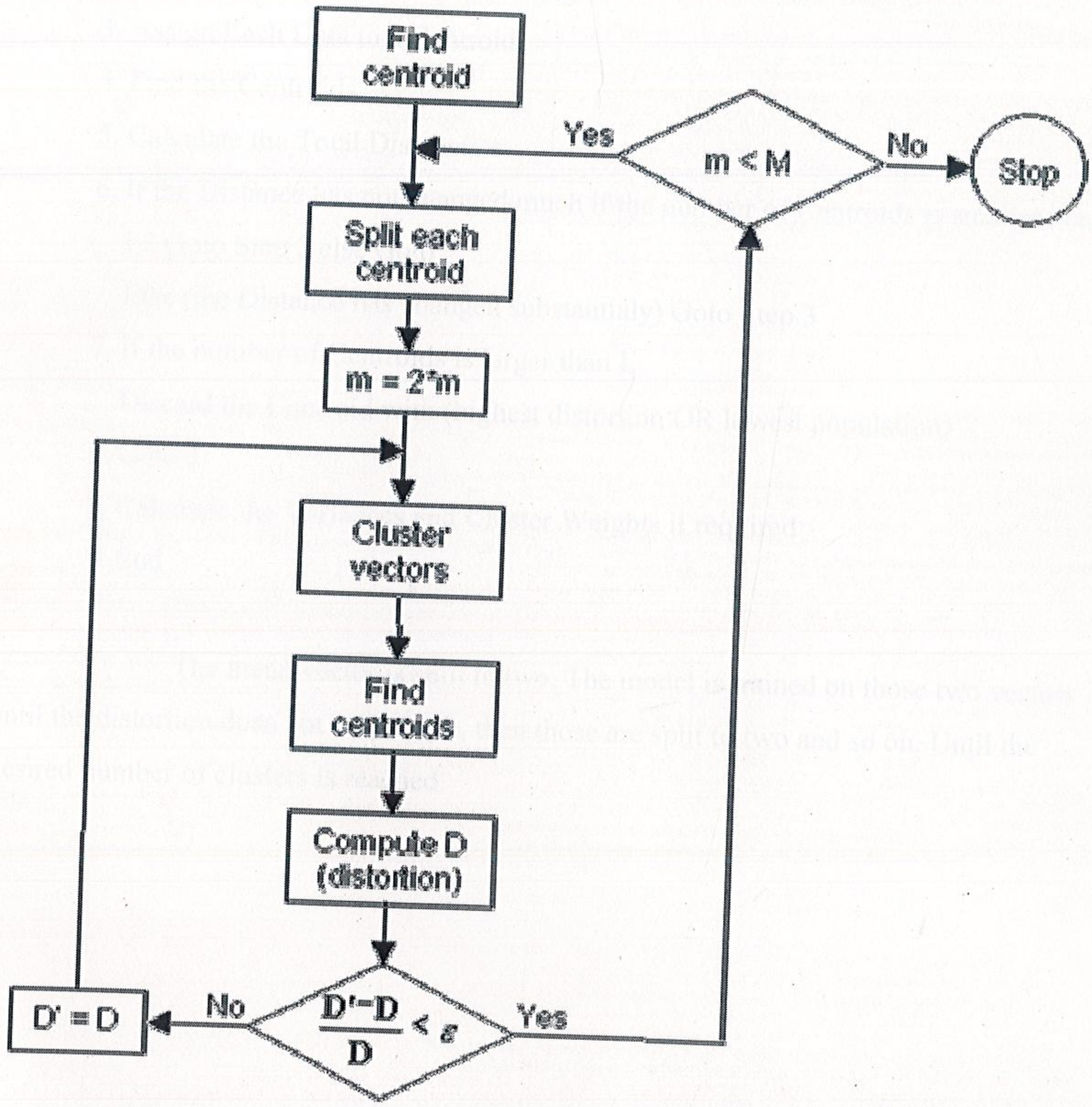

Fig - 4.2

Algorithm:

1. Find the Mean.
2. Split each Centroid to two (Splitting).
3. Assign Each Data to a Centroid.
4. Find the Centroids.
5. Calculate the Total Distance.
6. If the Distance has not changed much if the number of Centroids is smaller than L2 Goto Step 2 else Goto 7

   Else (the Distance has changed substantialy) Goto Step 3
7. If the number of Centroids is larger than L

   Discard the Centroid with (highest distortion OR lowest population)

   Goto 3
8. Calculate the Variances and Cluster Weights if required
9. End

The mean vector is split to two. The model is trained on those two vectors until the distortion does not vary much, then those are split to two and so on. Until the desired number of clusters is reached.

Artificial Neural Network-(ANN)

5.1 Introduction to ANN

An **artificial neural network** (ANN) or commonly just **neural network** (NN) is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing based on a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

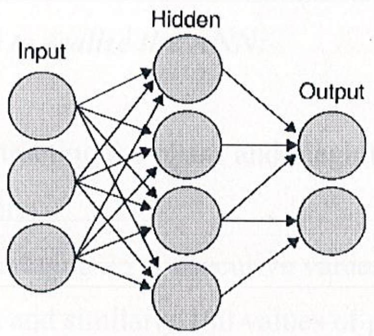

Fig – 5.1

The original inspiration for the technique was from examination of the central nervous system and the neurons (and their axons, dendrites and synapses) which constitute one of its most significant information processing elements. In a neural network model, simple nodes (called variously "neurons", "neurodes", "PEs" ("processing elements") or "units") are connected together to form a network of nodes — hence the term "neural network." While a neural network does not have to be adaptive per se, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow. These networks are also similar to the biological neural networks in the sense that functions are performed collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned

In modern software implementations of artificial neural networks the approach inspired by biology has more or less been abandoned for a more practical approach based on statistics and signal processing. In some of these systems neural networks, or parts of neural networks (such as artificial neurons are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do however have in common is the principle of non-linear, distributed, parallel and local processing and adaptation.

5.2 CODES AND BRIEF EXPLANATION

Steps to realize the ANN:

- First, extracting the phase and magnitude attributes of a wave file into 2 text files respectively.
- Then averaging 25 consecutive values of magnitude in the magnitude array in the text document and similarly 100 values of phase in the phase array in the text file respectively so as to create inputs for each of the 201 nodes of the ANN.
- Normalize these values of each of the training wave file inputs so that they don't cross the upper limit during the calculations.
- Train the ANN by using the above new normalized averaged magnitude array and phase arrays. During the hit and trail method we have come to know that with 33 hidden nodes and with 75 iterations we get an accurate classification with a maximum accuracy of 25%.
- After the optimization and the rest of the output files created, we consider only the text files having the weights_in andweights_out of the 33 hidden node values.
- Then, we parse these weights into the ANN by using the above two weight text files.
- Having done all the above steps, our ANN is ready!!

- CODES

Extracting the magnitude and phase attributes

```
for i=97:122
  X=sprintf('%s',i);
  for j=1:10
     b=sprintf('%d.wav',j);
     c=strcat(X,b);
     temp=wavread(c);

     temp1=fft(temp);
     mag=abs(temp1);
     file_name=strcat(c,'_mag.txt');
     fid=fopen(file_name,'w');
     for k=1:1025
        fprintf(fid,'%lf\n',mag(k));
     end
     pha=unwrap(angle(temp1));
     file_name1=strcat(c,'_pha.txt');
     fid1=fopen(file_name1,'w');
     for l=1:40000
        fprintf(fid1,'%lf\n',pha(l));
     end
     fclose(fid);
     fclose(fid1);
    end
  end
```

Compute input files for the ANN-(Prgrm1)

The code takes in the magnitude array text file and the phase array text file from the above code and averages each of the 25 values in the magnitude array file and averages 100 values in the phase array file correspondingly.

This determines the number of input nodes in the ANN program. As a result now the number of input nodes for the ANN network is 41 magnitude nodes and 160 phase nodes adding to 201.

Normalize

The code here takes in each attribute, normalizes them, in which the maximum will be 1 and minimum will be 0 Here the minimum most value that is considered initially is -10000 and the maximum value is 0 which in the course of computation get over written and get to the real values of the input array values.

5.3 Application of Neural Networks

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical. The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.

Application areas include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering [*Neural Computing and Applications*, Springer-Verlag].

6. RESULTS AND ANALYSIS

accuracy count for 0 class when 33 hidden nodes0.242857
ppv[33][0]====-1.#IND00
sensitivity[33][0]=====0.000000
specificity[33][0]=====1.000000
npv[33][0]====0.242857
accuracy count for 1 class when 33 hidden nodes0.242857
ppv[33][1]====0.166667
sensitivity[33][1]=====0.125000
specificity[33][1]=====0.166667
npv[33][1]====0.125000
accuracy count for 2 class when 33 hidden nodes0.242857
ppv[33][2]====-1.#IND00
sensitivity[33][2]=====0.000000
specificity[33][2]=====1.000000
npv[33][2]====0.142857
accuracy count for 3 class when 33 hidden nodes0.242857
ppv[33][3]====0.125000
sensitivity[33][3]=====0.166667
specificity[33][3]=====0.125000
npv[33][3]====0.166667
accuracy count for 4 class when 33 hidden nodes0.242857
ppv[33][4]====-1.#IND00
sensitivity[33][4]=====0.000000
specificity[33][4]=====1.000000
npv[33][4]====0.242857
accuracy count for 5 class when 33 hidden nodes0.242857
ppv[33][5]====-1.#IND00
sensitivity[33][5]=====0.000000
specificity[33][5]=====1.000000
npv[33][5]====0.242857
accuracy count for 6 class when 33 hidden nodes0.242857
ppv[33][6]====-1.#IND00
sensitivity[33][6]=====0.000000
specificity[33][6]=====1.000000
npv[33][6]====0.242857
accuracy count for 7 class when 33 hidden nodes0.242857
ppv[33][7]====-1.#IND00
sensitivity[33][7]=====0.000000
specificity[33][7]=====1.000000
npv[33][7]====0.242857
accuracy count for 8 class when 33 hidden nodes0.242857
ppv[33][8]====-1.#IND00
sensitivity[33][8]=====0.000000
specificity[33][8]=====1.000000
npv[33][8]====0.242857
accuracy count for 9 class when 33 hidden nodes0.242857
ppv[33][9]====-1.#IND00
sensitivity[33][9]=====0.000000

specificity[33][9]=====1.000000
npv[33][9]====0.242857
accuracy count for 10 class when 33 hidden nodes0.242857
ppv[33][10]====-1.#IND00
sensitivity[33][10]=====0.000000
specificity[33][10]=====1.000000
npv[33][10]====0.242857
accuracy count for 11 class when 33 hidden nodes0.242857
ppv[33][11]====-1.#IND00
sensitivity[33][11]=====0.000000
specificity[33][11]=====1.000000
npv[33][11]====0.242857
accuracy count for 12 class when 33 hidden nodes0.242857
ppv[33][12]====-1.#IND00
sensitivity[33][12]=====0.000000
specificity[33][12]=====1.000000
npv[33][12]====0.242857
accuracy count for 13 class when 33 hidden nodes0.242857
ppv[33][13]====-1.#IND00
sensitivity[33][13]=====0.000000
specificity[33][13]=====1.000000
npv[33][13]====0.242857
accuracy count for 14 class when 33 hidden nodes0.242857
ppv[33][14]====-1.#IND00
sensitivity[33][14]=====0.000000
specificity[33][14]=====1.000000
npv[33][14]====0.242857
accuracy count for 15 class when 33 hidden nodes0.242857
ppv[33][15]====-1.#IND00
sensitivity[33][15]=====0.000000
specificity[33][15]=====1.000000
npv[33][15]====0.242857
accuracy count for 16 class when 33 hidden nodes0.242857
ppv[33][16]====-1.#IND00
sensitivity[33][16]=====0.000000
specificity[33][16]=====1.000000
npv[33][16]====0.242857
accuracy count for 17 class when 33 hidden nodes0.242857
ppv[33][17]====-1.#IND00
sensitivity[33][17]=====0.000000
specificity[33][17]=====1.000000
npv[33][17]====0.242857
accuracy count for 18 class when 33 hidden nodes0.242857
ppv[33][18]====-1.#IND00
sensitivity[33][18]=====0.000000
specificity[33][18]=====1.000000
npv[33][18]====0.242857
accuracy count for 19 class when 33 hidden nodes0.242857
ppv[33][19]====-1.#IND00
sensitivity[33][19]=====0.000000
specificity[33][19]=====1.000000

npv[33][19]====0.242857
accuracy count for 20 class when 33 hidden nodes0.242857
ppv[33][20]====-1.#IND00
sensitivity[33][20]=====0.000000
specificity[33][20]=====1.000000
npv[33][20]====0.242857
accuracy count for 21 class when 33 hidden nodes0.242857
ppv[33][21]====-1.#IND00
sensitivity[33][21]=====0.000000
specificity[33][21]=====1.000000
npv[33][21]====0.242857
accuracy count for 22 class when 33 hidden nodes0.242857
ppv[33][22]====-1.#IND00
sensitivity[33][22]=====0.000000
specificity[33][22]=====1.000000
npv[33][22]====0.242857
accuracy count for 23 class when 33 hidden nodes0.242857
ppv[33][23]====-1.#IND00
sensitivity[33][23]=====0.000000
specificity[33][23]=====1.000000
npv[33][23]====0.242857
accuracy count for 24 class when 33 hidden nodes0.242857
ppv[33][24]====-1.#IND00
sensitivity[33][24]=====0.000000
specificity[33][24]=====1.000000
npv[33][24]====0.242857
accuracy count for 25 class when 33 hidden nodes0.242857
ppv[33][25]====-1.#IND00
sensitivity[33][25]=====0.000000
specificity[33][25]=====1.000000
npv[33][25]====0.242857

This above result shows the output of 33 hidden noded 201 input and 26 output node ANN has got the better output of all the nodes.

7. CONCLUSION

There are many developed HMM models to recognize speech by identifying the presence of a letter in a speech sequence. Our project deals with the identification and classification of a speech signal, which mainly is of letters, and as a result we have used ANN model that has its strength in classification.

Here in this project we were able to get an accuracy of classification and identification to just over 25% which further can surely be improved to a greater extent by improving the training vector inputs.

We have considered both the important features of speech signal, both being the amplitude and phase, thus improving the accuracy in identification of a speech signal input.

Hence, in out project we have had a great learning experience of vector quantization and ANN programming, which is the cutting edge technology in the research field of speech recognition.

8. BIBLIOGRAPHY

[1]P. Ramesh Babu, Digital Signal Processing
Scitech Publications, 2003

[2] Taub and Schilling, Principles Of Communication Systems
Tata McGraw Hill Edition, 1991

[3] Rudra Pratap, Getting Started With Matlab7
Oxford University Press, 2006

[4] B.P.Lathi, Modern Digital and Analog Communication Systems
Oxford University Press, 1998

[5] Salivahanan,Vallavaraj,Gnanapriya, Digital Signal Processing
Tata McGraw Hill Edition, 2005

[6] Oppenheim and Willsky, Signals and Systems
Pearson Education Inc., 1997

[7] Pratt, Digital Image Processing
John Wiley and Sons Inc., 2002

[8] www.ieeexplore.com

[9] www.mathworks.com

[10] www.wikipedia.com

[11] Hidden Markov Model: A guided tour by Alan.B.Porits ,
Institute of Defense Analysis, Princeton - PAPER

[12] Fourier Transform Phase-Based Features for Speech Recognition,
By Rajesh Mahanand Hedge

[13] On the importance of Phase in Human Speech Recognition
By Guangji Shi,Maryam Modir Shanechi,Parham Aarabi

[14] Lecture notes in Speech Production, Speech Coding and Speech Recognition
By Mark Hasegawa-Johnson