A STUDY AND APPLICATION OF SPEECH RECOGNITION OF HINDI LETTERS

By

GAURAV GAYAWAR - 041031
SAH GEETANSH KRISHNA - 041091
KAPIL SACHDEVA - 041047

विद्या तत्व ज्योतिषम:

JAYPEE UNIVERSITY OF
INFORMATION TECHNOLOGY

MAY-2008

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY - WAKNAGHAT

CERTIFICATE

This is to certify that the work entitled, "**A STUDY AND APPLICATION OF SPEECH RECOGNITION OF HINDI LETTERS**" submitted by **Gaurav Gayawar, Sah Geetansh Krishna and Kapil Sachdeva** in partial fulfillment for the award of degree of Bachelor of Technology in Department of Electronics and Communication Engineering 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.

[Prof. S.V.Bhooshan]
H.O.D.
Dept. of Electronics and Communication
J.U.I.T., waknaghat.

[Mr. Vinay Kumar]
Sr. Lecturer
Dept. of Electronics and Communication
J.U.I.T., waknaghat.

ACKNOWLEDGMENT

No research endeavor is a sole exercise; various individuals in their own capacity at some point or other contributed in bringing of fruition the research endeavor, in acknowledging their guidance, support and assistance, we humbly thank them.

The sense of contentment and elation accompanies the successful completion of this project would be incomplete without mentioning the names of those people who helped us in accomplishing this project. People whose constant guidance, support and encouragement resulted in its realization.

We express a deep sense of gratitude to Prof. Sunil V. Bhooshan for providing the inspiration required for taking the project to its completion.

With great privilege we place on record our heartful gratitude and unforgettable personal indebtness to Mr. Vinay Kumar, Sr. Lecturer, Department of Electronics and Communication & coordinator of our project for this worthy guidance, constant support throughout the period of investigation and preparation of this manuscript. We are sure, without his help it would have been impossible for us to complete this venture successfully.

With great pleasure, we express our heartiest and esteem sense of gratitude to other faculty members and our colleagues for giving new dimensions to the present investigation by exploring new vistas with their critical observations, valuable suggestions and timely help whenever required, so as to reach to some logical conclusion during our course of investigation. Their unobtrusive support and suggestions bolstered our confidence as usual.

Finally, we thank the ALMIGHTY for his love, care and vision provided to us from time to time.

Gaurav Gayawar
Gaurav Gayawar
(041031)

S Krishna
Sah Geetansh Krishna
(041091)

Kapil Sachdeva
(041047)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

B.W.	-	Band Width
D.F.T.	-	Discrete Fourier Transform
D.S.P.	-	Digital Signal Processing
F.F.T.	-	Fast Fourier Transform
F.I.R.	-	Finite Impulse Response
H.M.M.	-	Hidden Markov Model
I.I.R.	-	Infinite Impulse Response
L.P.C.	-	Linear Predictive Code
V.Q.	-	Vector Quantization
X.M.L.	-	Extensible Markup Language

ABSTRACT

Speech interface to computer is the next big step that computer sciences need to take for general users. Speech recognition will play a important role in taking technology to them. The need is not only for Speech interface, but Speech interface in local languages. Our goal is to create Speech recognition software that can recognize Hindi words. This report takes a brief look at the basic building block of a Speech recognition engine. It talks about implementation of different modules. Sound Recorder, Feature Extractor and HMM training and Recognizer modules have been described in details. The results of the experiments that were conducted are also provided. The report ends with a conclusion and Future plan. Keyboard, although a popular medium is not very convenient as it requires a certain amount of skill for effective usage. A mouse on the other hand requires a good hand-eye co-ordination. It is also cumbersome for entering non-trivial amount of text data and hence requires use of an additional media such as keyboard. Physically challenged people find computers difficult to use. Partially blind people find reading from a monitor difficult. Current computer interfaces also assume a certain level of literacy from the user. It also expects the user to have certain level of proficiency in English. In our country where literacy level is as low as 50% in some states, if information technology has to reach the grass root level; these constraints have to be eliminated. Speech interface can help us to tackle these problems.

CHAPTER-I

INTRODUCTION

Definition:

Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words. The recognized words can be the final results, as for applications such as commands control, data entry, and document preparation. They can also serve as the input to further linguistic processing in order to achieve speech understanding, a subject covered in section. In simple terms speech recognition is the process of converting spoken input to text. Speech recognition is thus sometimes referred to as speech-to-text. Speech recognition allows you to provide input to an application with your voice. Just like clicking with your mouse, typing on your keyboard, or pressing a key on the phone keypad provides input to an application, speech recognition allows you to provide input by talking. In the desktop world, you need a microphone to be able to do this. In the Voice XML world, all you need is a telephone.

Outline:

Speech recognition systems can be characterized by many parameters, some of the more important of which are shown in Figure below. An isolated-word speech recognition system requires that the speaker pause briefly between words, whereas a continuous speech recognition system does not. Spontaneous, or extemporaneously generated, speech contains dis-fluencies, and is much more difficult to recognize than speech read from script. Some systems require speaker enrollment - a user must provide samples of his or her speech before using them, whereas other systems are said to be speaker-independent, in that no enrollment is necessary. Some of the other parameters depend on the specific task. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words. When speech is produced in a sequence of words, language models or artificial grammars are used to restrict the combination of words.

PARAMETERS	RANGE
Speaking mode	Isolated words to continuous speech
Speaking style	Read speech to spontaneous speech
Enrollment	Speaker-dependent to speaker-independent
Vocabulary	Small (< 20 words) to large (> 20,000 words)
Language model	Finite-state to content-sensitive
Perplexity	Small (<10) to large (> 100)
SNR	High (> 30dB) to low (< 10 dB)
Transducer	Voice-canceling microphone to telephone

Fig. 1 Table showing various parameters and range of speech

CHAPTER-II

PROCEDURE

CEPSTRUM METHOD (METHOD – I)

Cepstrum:

The cepstrum is a common transform used to gain information from a person's speech signal. It can be used to separate the excitation signal (which contains the words and the pitch) and the transfer function (which contains the voice quality).

It is convenient to assume that the signal consists of a discrete time sequence, so that the spectrum consists of a z-transform evaluated on the unit circle. Let us consider a speech example, with X referring to the spectrum of the observed speech signal, E to the excitation component (for instance, the glottal pulse train), and V to the vocal tract shaping of the excitation spectrum. We begin with a multiplicative model of the two spectrums (the excitation and the vocal tract). Thus the spectral of the speech signal can be written as

$$X(w) = E(w)\, V(w) \qquad \ldots(i)$$

Taking the logarithm of above equation yields

$$\log(X(w)) = \log(E(w)) + \log(V(w)) \qquad \ldots(ii)$$

Particularly for voiced sounds, it can be observed that the E term corresponds to an event that is relatively extended in time (e.g. a pulse train with pulses every 10 ms), and thus it yields a spectrum that should be characterized by a relatively, rapidly varying function of w; in comparison, because of the relatively short impulse response of the vocal tract, the V term varies more slowly with w. With the use of this knowledge, the left hand side of equation (ii) can be separated into the two right hand side components by a kind of a filter that separates the log spectral components that vary rapidly with w (the so-called high time components) from those vary slowly with w (the low time components). Such an operation would essentially be performing de-convolution.

Equation (ii) has transformed the multiplicative equation (i) into a linear operation and thus can be subjected to the linear operations such as filtering. Since the variable is frequency rather than time, notations must be changed. Thus, for example, rather than filtering (for time), we have liftering (for frequency); instead of a frequency response we have a quefrency response; and the DFT (or Z-transform or Fourier transform) of the log(X(w)) is called the cepstrum. The cepstrum is computed by taking the inverse z-transform of equation (ii) on the unit circle yielding

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(X(w)) e^{jwn} dw$$

Where c(n) is called the *n*th complex cepstral coefficient.

After calculating complex cepstral coefficient, next step is vector quantization.

What is Quantization ?

Quantization, the process of approximating continuous-amplitude signals by digital (discrete amplitude) signals, is an important aspect of data compression or coding, the field concerned with the reduction of the number of bits necessary to transmit or store analog data, subject to a distortion or fidelity criterion. The independent quantization of each signal value or parameter is termed scalar quantization, while the joint quantization of a block of parameters is termed block or vector quantization.

Vector Quantization:

A vector quantizer maps k-dimensional vectors in the vector space R^k into a finite set of vectors Y = {y_i: i = 1, 2 ...N}. Each vector y_i is called a code vector or a codeword and the set of all the code words is called a codebook. Associated with each codeword, y_i is a nearest neighbor region called Voronoi region, and it is defined by:

$$V_i = \{x \in R^k : \| x - y_i \| \leq \| x - y_i \|, \text{ for all } j \neq i\}$$

The set of Voronoi regions partition the entire space R^k such that:

$$\bigcup_{i=1}^{N} V_i = R^k$$

$$\bigcap_{i=1}^{N} V_i = \phi \qquad \text{for all } i \neq j$$

As an example we take vectors in the two dimensional case without loss of generality. Figure below shows some vectors in space. Associated with each cluster of vectors is a representative codeword. Each codeword resides in its own Voronoi region. These regions are separated with imaginary lines in figure for illustration. Given an input vector, the codeword that is chosen to represent it is the one in the same Voronoi region.

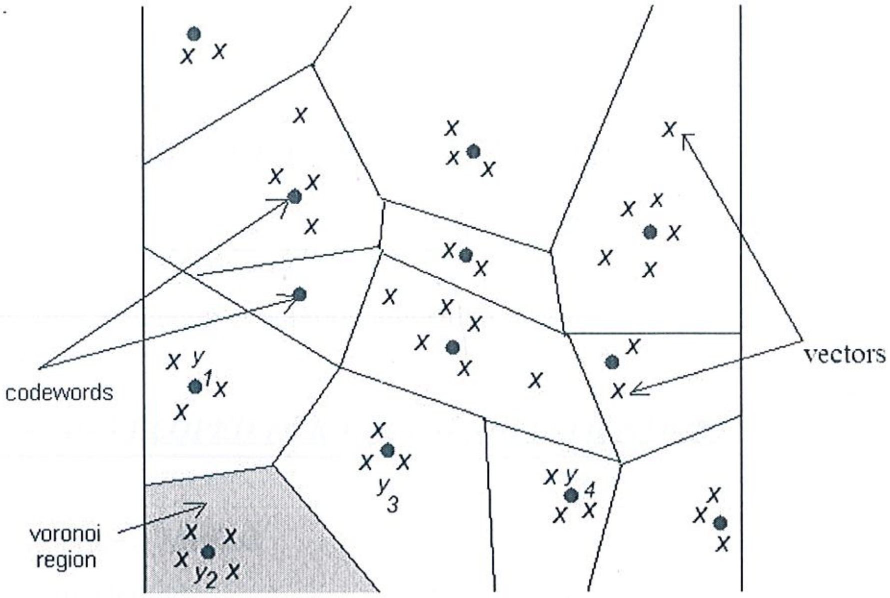


Fig. 2 Figure showing vornoi regions, vectors and code words.

Figure: Code words in 2-dimensional space. Input vectors are marked with an x, code words are marked with red circles and the Voronoi regions are separated with boundary lines.
The representative codeword is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2}$$

Where x_j is the j^th component of the input vector, and y_{ij} is the j^th is component of the codeword y_i.

<u>How is the codebook designed ?</u>

So far we have talked about the way VQ works, but we haven't talked about how to generate the codebook. What code words best represent a given set of input vectors? How many should be chosen?

It requires an exhaustive search for the best possible code words in space, and the search increases exponentially as the number of code words increases. We therefore resort to suboptimal codebook design schemes, and the first one that comes to mind is the simplest. This can be named as splitting method.

THE ALGORITHMS

ALGORITHMS STUDIED FOR VECTOR QUANTIZATION:

1. K – means clustering algorithm
2. Binary split codebook generation algorithm

ALGORITHMS DEVELOPED FOR VECTOR QUANTIZATION:

I. Region Defining Method

II. V - Q Split Method

1. K-MEANS CLUSTERING ALGORITHM

Initialization: Arbitrarily choose M vectors (initially out of the training set of L vectors) as the initial set of code words in the codebook.

Nearest-Neighbour Search: For each training vector, find the code word in the current codebook i.e., Closest (in terms of spectral distance) and assign that vector to the corresponding cell (associated with the closest codeword).

Centroid update: Update the codeword in each cell using the centroid of the training vectors to that cell.

Iteration: Repeat steps 2 and 3 until the average distance falls below a preset threshold.

2. BINARY SPLIT CODEBOOK GENERATION ALGORITHM

Design a 1 vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).

Double the size of the codebook by splitting each current codebook Y_n according to the rule.

$$y_{n+} = y_n(1+e)$$

$$y_{n-} = y_n(1-e)$$

Where n varies from 1 to the current size of the codebook and e is a splitting parameter (typically e is chosen in the range $0.01 \leq e \leq 0.05$).

Use the K-means iterative algorithm to get the best set of centroids for the split codebook (i.e. the codebook of twice the size).

Iterate steps 2 and 3 until a codebook of size M is designed.

I. REGION DEFINING METHOD

Find the maximum distance by comparing between every two code vectors.

Now plot two semicircles by assuming both the code vectors as centers which is at the maximum distance as found in step I.

The common region formed by the two plots will now be divided into smaller regions by using different available methods.

Advantage: It will be easier to visualize the whole region and manipulate the processing.

Disadvantage: By this method we get unequal distribution of samples in every region, so it is not very accurate and do not give a precise result.

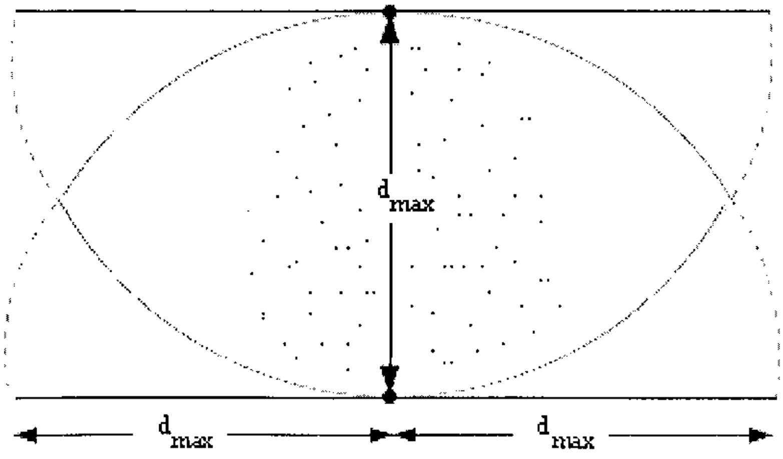


Fig. 3 Figure showing the common region formed by two semi-circles

II. ***V-Q SPLIT METHOD***

Find the centroid of the entire set of code vectors.

Draw a line parallel to y-axis which passes through this centroid, which will divide whole space of code vectors into two regions.

If the number of code vectors in both the regions is not equal, then rotate the line by an angle theta.

Repeat step III until we get equal number of code vectors in both the regions.

Now, again split the new small regions into two smaller regions by checking the number of points on both the sides of the division-line.

Repeat step V until we get the desired number of regions.

Advantage: By this method we can get an accurate result and more proper representation of the samples.

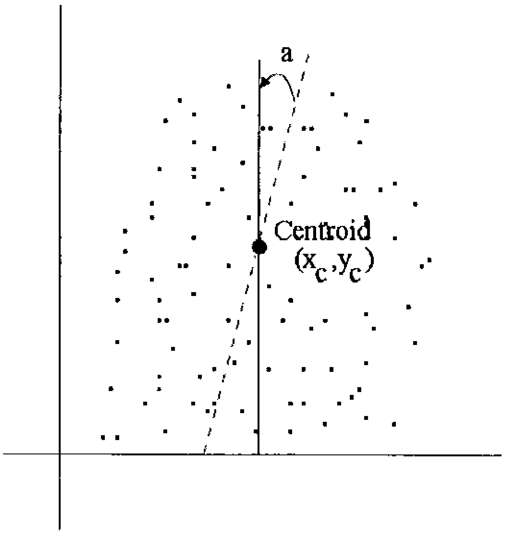


Fig. 4 Figure showing the code vectors, centroid and the line passing through the centroid

Matching:

This method was not find to be so useful because the coefficient obtained after vector quantization was not enough distinct to differentiate letter on that basis.

CHAPTER-III

L.P.C. CEPSTRAL METHOD (METHOD – II)

Block Diagram of Steps Involved In Speech Recognition:

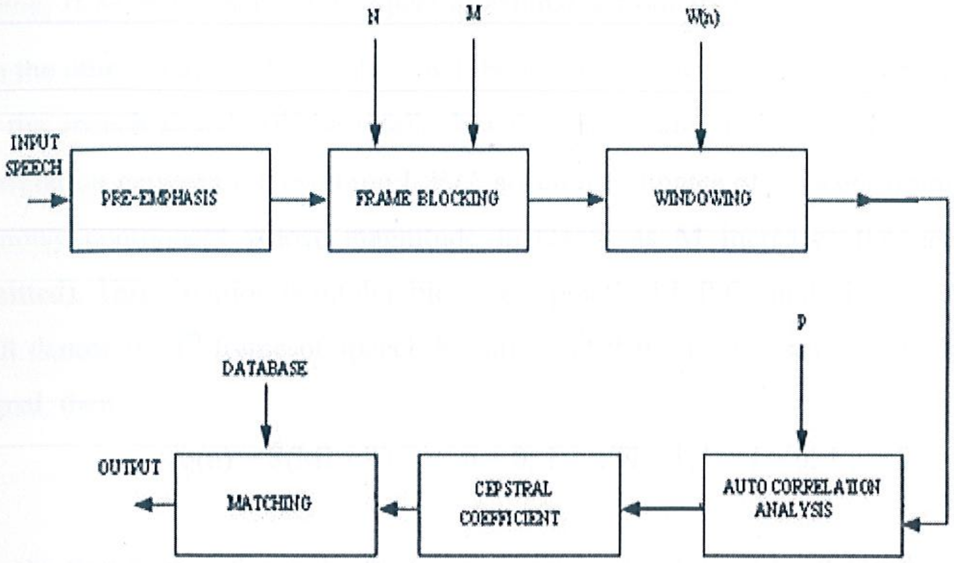


Fig. 5 Block Diagram of Steps Involved In Speech Recognition

Pre-emphasis:

The digitized speech signal s(n), is put through a low order digital system (typically a first order FIR filter) to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The digital system used in the pre-emphasizer is either fixed or slowly adaptive (e.g. - to average transmission conditions, noise backgrounds, or even to average signal spectrum) perhaps the most widely used pre-emphasis network is the first order system.

$$H(z) = 1 - \bar{a}z-1, 0.9 \leq a \leq 1.0$$

This case, the output of the pre-emphasis network $\hat{s}$ (n), is related to the input to the network s(n) by the difference equation:

$$\hat{s}(n) = s(n) - \bar{a}\, s(n - 1)$$

The most common value for $\bar{a}$ is around 0.95. (For fixed point implementations a value of $\bar{a}$ = 15/16 = 0.9375 is often used.

Frame Blocking:

In this step the pre-emphasized speech signal, ŝ(n) is blocked into frames of N samples, with adjacent frames being separated by M samples. It is easy to say if $M \leq N$, then adjacent frames overlap and the resulting L.P.C. spectral estimates will be correlated from frame to frame; If $M \ll N$, then L.P.C. spectral estimates from frame to frame will be quite smooth. On the other hand, if $M \leq N$, there will be no overlap between adjacent frames; in face, some of the speech signal will be totally lost (i.e. never appear in any analysis frame), and the correlation between the resulting L.P.C. spectral estimates of adjacent frames will be contain a noisy component whose magnitude increases as M increases (i.e. as more speech is omitted). This situation is intolerable in any practical L.P.C. analysis for speech recognition. If it denote the 1^{th} frame of speech by $x_l(n)$ and there are L frames within the entire speech signal, then

$$x_l(n) = \check{S}(Ml + n), \qquad n = 0, 1 \ldots N-1, \qquad l = 0, 1 \ldots L-1$$

i.e. the first frame of speech x0(n) encompasses speech samples Š (0), Š (1), . . . , Š (N−1), the second frame of speech x1(n) encompasses speech samples Š(M), Š(M+1), . . . , Š(M+N−1) and the 1^{th} frame of speech $x_{l-1}(n)$ encompasses speech samples Š(M(L−1)), Š(M(L−1)+1), . . . , Š(M(L−1)+N−1) typical values for N and M are 300 and 100 when the sampling of the speech is 6.67 kHz.

Windowing:

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and the end of each frame, the concept here is to use window to taper the signal to 0 at the beginning and at the end of each frame. If we define the window as w(n), $0 \leq n \leq N-1$, then the result of windowing is the signal

$$X'_l(n) = x_l(n)\, w(n), \qquad 0 \leq n \leq N-1$$

A typical window used for the auto-correlation method for the L.P.C. (the method most widely used for recognition systems) is the Hamming window, which has the form

$$w(n) = 0.54 - 0.46 \text{ Cos } (2\pi n/N - 1), \qquad 0 \leq n \leq N-1$$

Auto-correlation Analysis:

Each frame of windowed signal is next auto-correlated to give

$$r_l(m) = \sum_{n=0}^{N-1-m} x_l(n)x_l(n+m), \qquad \text{m = 0, 1 ... p}$$

Where the highest auto-correlation value p, is the order of L.P.C. analysis. Typically values of p from 8 to 16 have been used, with p=8, been the value used for the most systems to be described in this book. A side benefit of the auto-correlation analysis is that the zeroth auto-correlation, R_{l0}, is the energy of the lth frame. The frame energy is an important parameter for the speech detection systems.

L.P.C. Analysis:

The next processing step is the L.P.C. analysis, which converts each frame of p + 1 auto-correlation into L.P.C. coefficients.

Cepstral Coefficients:

The cepstrum is a homomorphic space which allows de-convolution of the signal from the vocal tract shape.

$$C(n) = \frac{1}{N_s}\sum_{k=0}^{N_s} \log|S_{avg}(k)|^{j\frac{2\pi}{N_s}kn} \qquad 0 \le n \le N_s - 1$$

Where $S_{avg}(k)$ are the filter bank amplitudes. Cepstral coefficients can be derived directly from the L.P.C. coefficients set.

$$c_o = \ln \sigma^2$$

$$c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \qquad 1 \le m \le p$$

$$c_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \qquad m > p$$

Where σ^2 is gain term in the L.P.C. model. The cepstral coefficients which are the coefficients of Fourier Transform representation of the Log Magnitude Spectrum, have been shown to be a more robust, reliable feature set for speech recognition than the L.P.C.

coefficients, the PARCOR coefficients, or the log area ratio coefficients. Generally, a Cepstral representation with Q > p coefficients is used, where Q ≃ (3/2)p.

Matching:

For matching we will use pattern recognition.

Make templates of cepstral coefficients for a particular letter.

$$r^j_1 = \{c^j_1, c^j_2, c^j_3 \ldots c^j_l\},$$

Where r^j_1 denotes the first template of j^{th} letter and each c_i is the n-dimensional cepstral vector of the input speech of i^{th} frame and l is the total number of frames of speech. In a similar manner we define a set of k templates for j^{th} letter. And by calculating average of each corresponding cepstral frame of all k templates we will obtain the average template R^j for j^{th} letter.

In a similar manner we define a set of reference patterns, $\{R^1, R^2 \ldots R^v\}$

In a similar manner we define template of cepstral coefficients for testing data

$$T = \{c_1, c_2, c_3 \ldots c_l\}$$

The goal of the pattern-comparison stage is to determine the dissimilarity or distance of T to each of the R^j, $1 \leq j \leq V$, in order to indentify the pattern that has the minimum dissimilarity, and to associate the spoken input with this pattern.

The distortion for j^{th} letter is measured as

$$D_j = \sum_{i=1}^{l}\sum_{n=1}^{q} | c_i^j(n) - c_i(n) |$$

The distortion calculated for different letters was not enough distinct to differentiate between two different letters.

L.P.C. EXCITATION METHOD (METHOD-III)

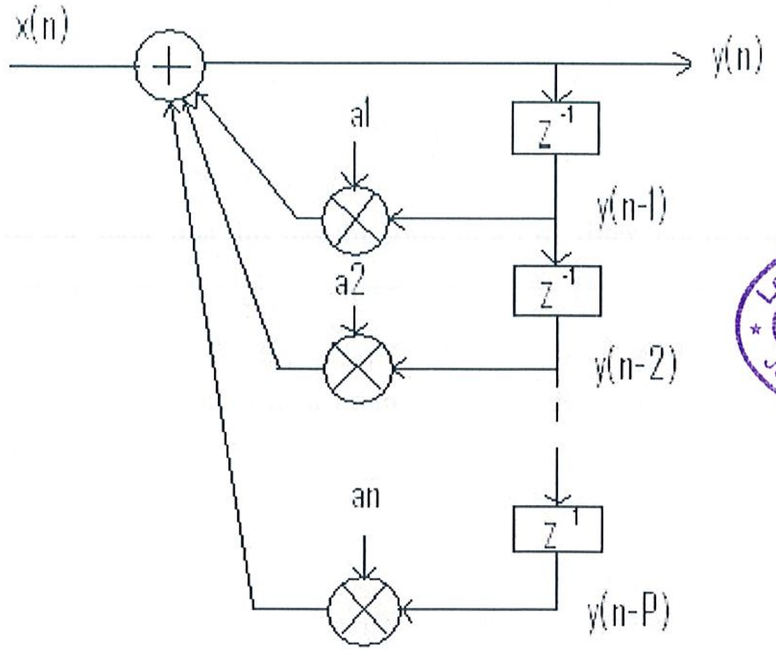


Fig. 6: All pole model for the generation of a discrete-time sequence

A direct form implementation of the spectral model:

$$H(z) = \frac{1}{1 - \sum_{j=1}^{P} a_j z^{-j}}$$

Where P is twice the number of second -order sections going into the product (P=12 in the example above),and the a coefficients are the resulting Pth-order polynomial.

Figure 1 is a diagram of the complete model. Such a system will be used as a starting point to describe linear predictive approaches to speech synthesis, but in the current context it will be used as a model to represent the signal spectrum. Thus, the short-term spectrum of a speech signal can be represented by a filter that can be specified by

P=2*(BW+1) coefficients, where BW is the speech bandwidth in kHz. Note that since the driving-signal spectrum is folded into the filter, the model excitations are considered to be white. For the system shown in fig.1, the discrete-time response y(n) to an excitation signal x(n) would be

$$y(n) = x(n) + \sum_{j=1}^{P} a_j y(n-j)$$

The coefficients for the second term of this expression are generally computed to give an approximation to the original sequence, which will yield a spectrum for H(z) that is an approximation to the original speech spectrum. Thus, we attempt to predict the speech signal by a weighted sum of its previous values. That is

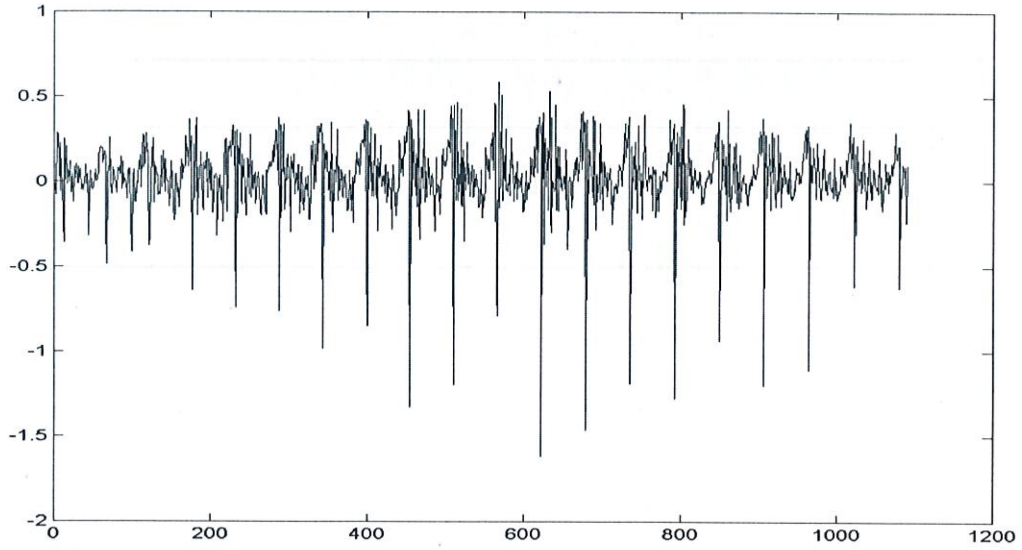
$$y'(n) = \sum_{j=1}^{P} a_j y(n-j)$$

is the linear predictor. Note that this has the form of a FIR filter, but that when it is included in the model of fig.1 the resulting production model is IIR. The coefficients that yield the best approximation y'(n) to y(n) (usually in the mean squared sense) are called the linear prediction coefficients. In the statistical literature, the overall model is sometimes called an autoregressive model.

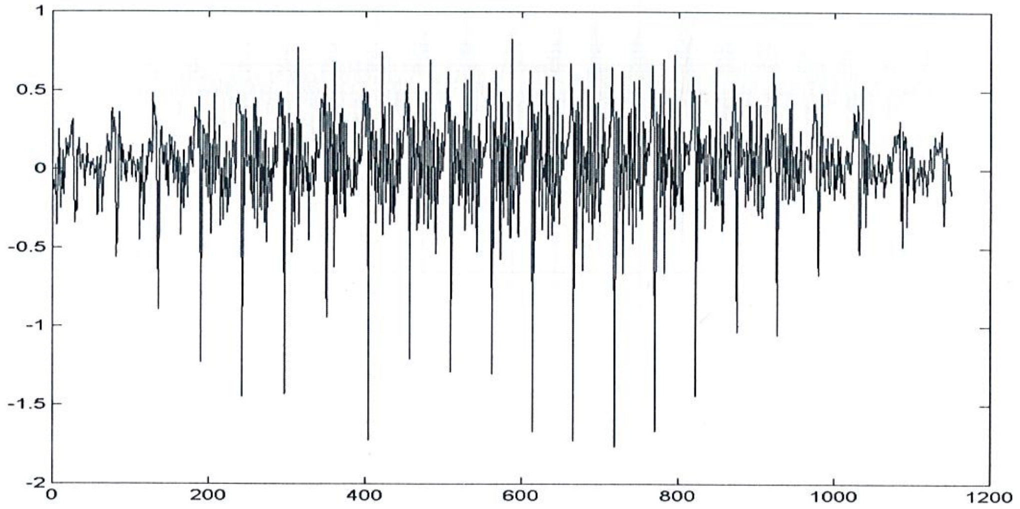
The difference between the predictor and the original signal is referred to as the error signal, also sometimes called residual error, the LPC residual, or the prediction error. When the coefficients are chosen to minimize this signal energy, the resulting error signal can be viewed as an approximation to the excitation function. The residual signal e(n) = y(n) - $y'(n)$ consists of the components of y(n) that are not linearly predictable from its own previous samples, which is the case for a periodic excitation in this model, assuming that the number of samples, which is the case for a periodic excitation in this model, assuming that the no. of samples between excitation pulses is much larger than the order of the filter

Plots for the excitation signals:

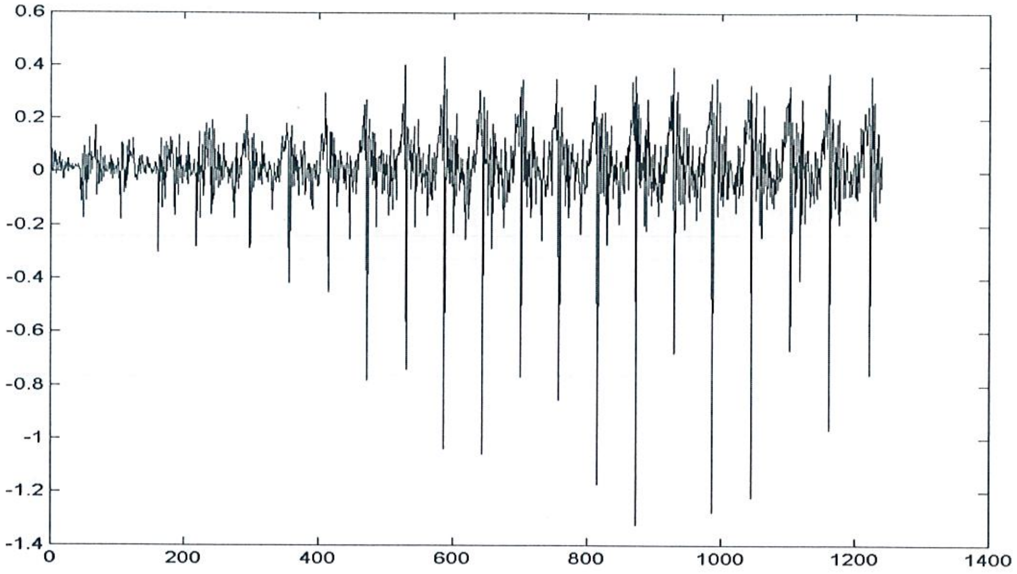
1. Excitation signal for a1



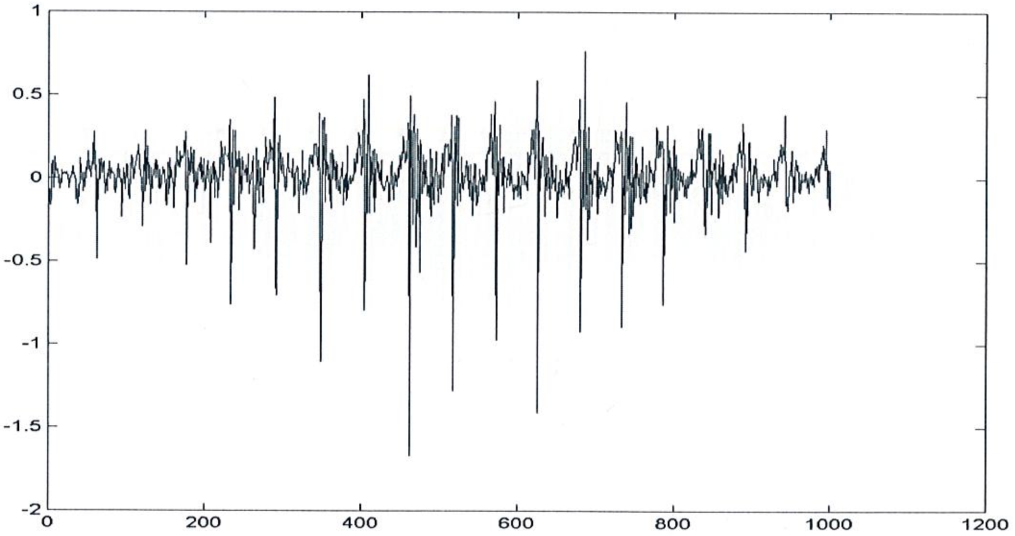
2. Excitation signal for a2



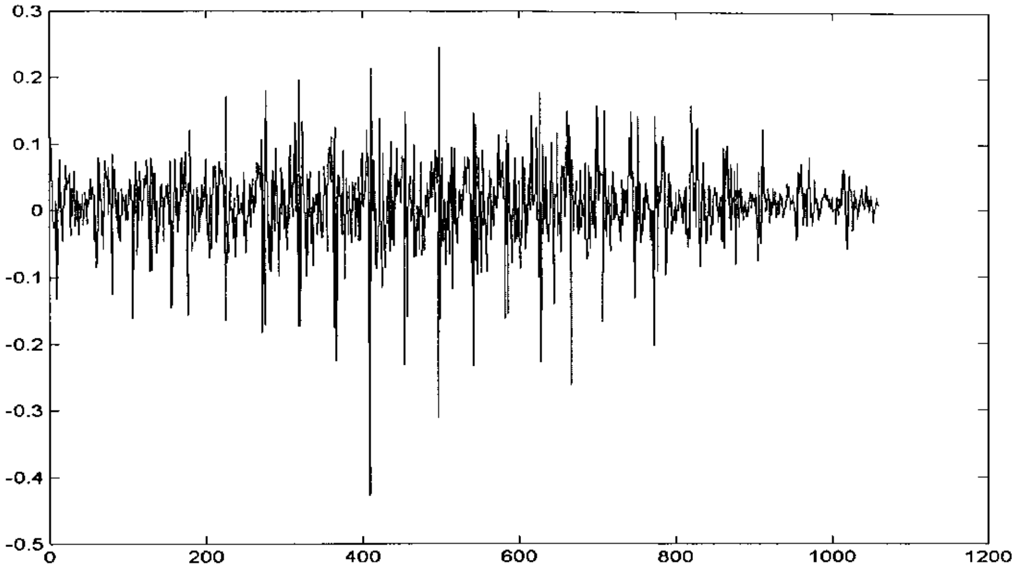
3. Excitation signal for a3



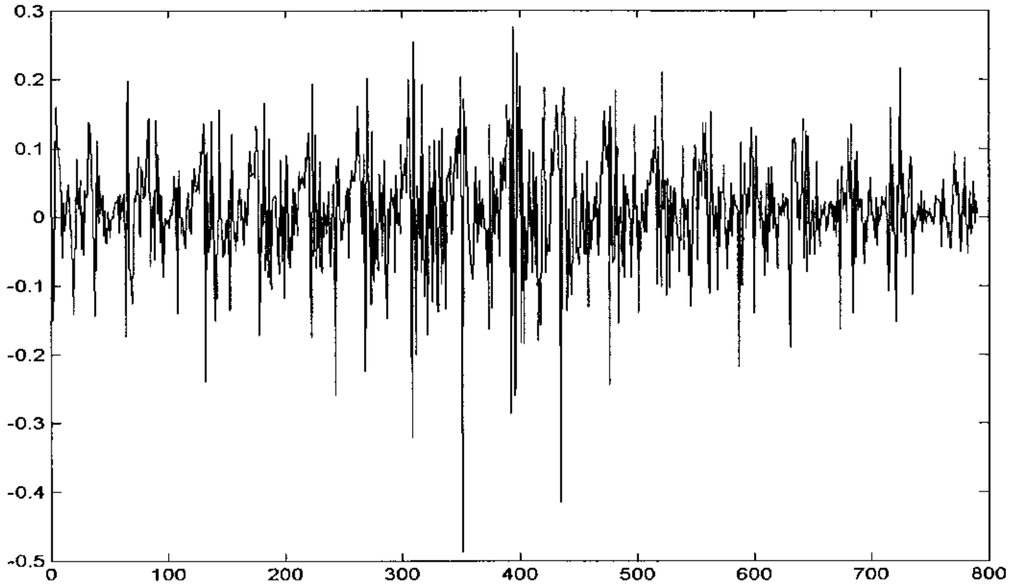
4. Excitation signal for a4



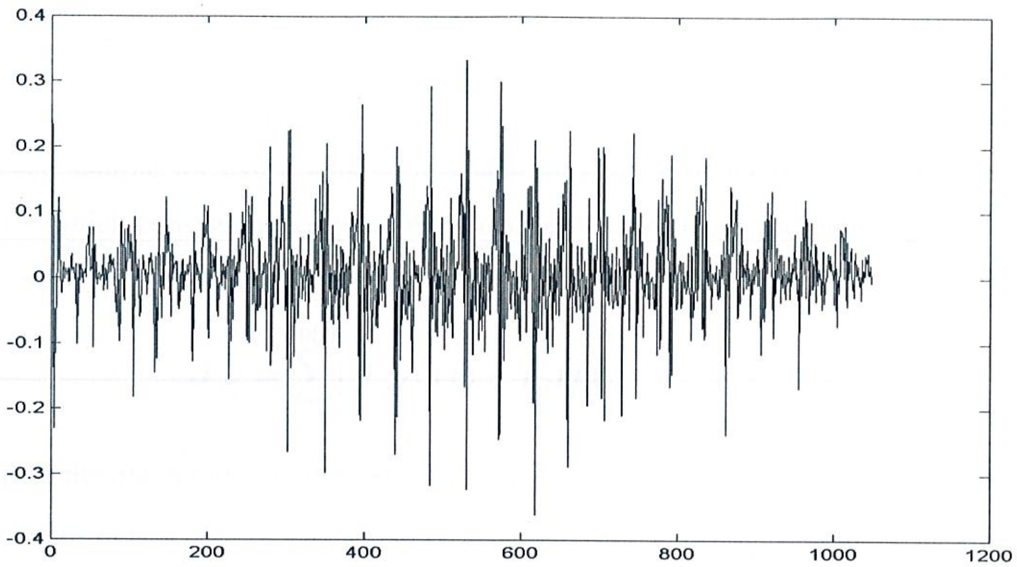
5. Excitation signal for e1



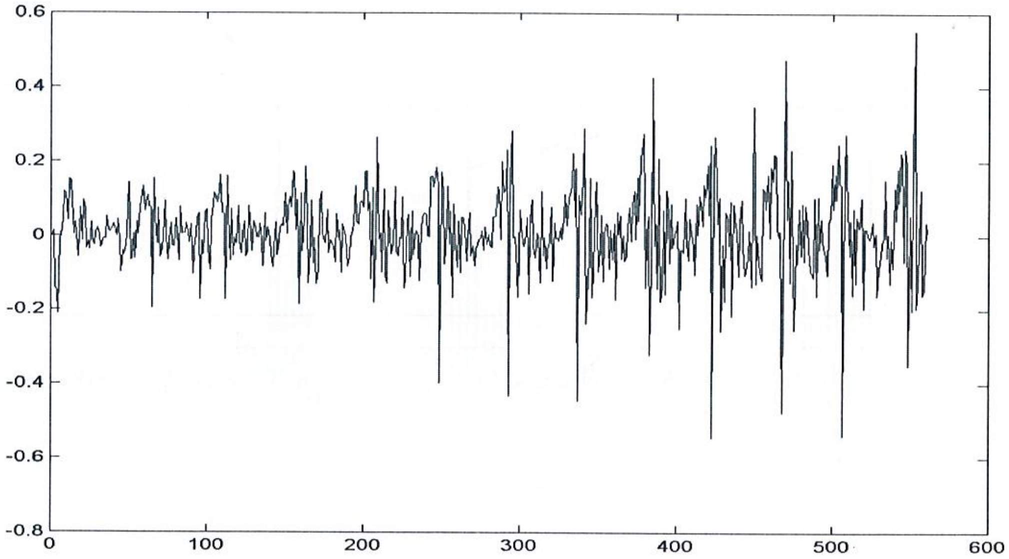
6. Excitation signal for e2



7. Excitation signal for e3



8. Excitation signal for e4



All the above plots of the excitation signals show that there is no characteristic excitation signal for particular letter. Few plots of "*a*" are similar to plots of "*e*" and all the plots of "*a*" are also not similar. Therefore, we prefer not to use this method for speech recognition.

CHAPTER-V

F.F.T. METHOD (METHOD – IV)

We have considered ten samples of each letter in our database. First of all, calculate thousand points FFT of each sample for every letter. Then take magnitude of them. This forms the reference database which consists of ten patterns of each particular letter for matching. Then compare the distortion with each ten patterns for every letter. Distortion is given by

$$d_{jS} = \sum_{i=1}^{1000} | f_j j_S(i) - f_d(i) |$$

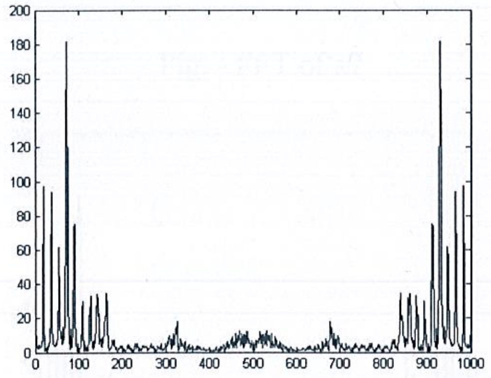
where, f_{jS} is the magnitude of 1000 points FFT of S^{th} sample of j^{th} letter,

f_d is the magnitude of 1000 point FFT of data,

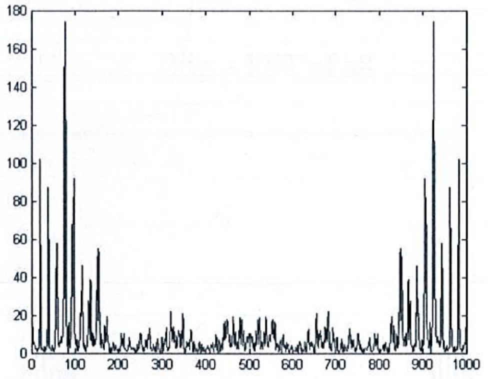
and i denotes the frequency.

The letter with the minimum distortion matches with the tested data and displayed as output.

PLOTS OF MAGNITUDE OF 1000 POINT FFT OF TEN SAMPLES OF "*a*" AND DATA OF "*a*":



FFT of a1



FFT of a2

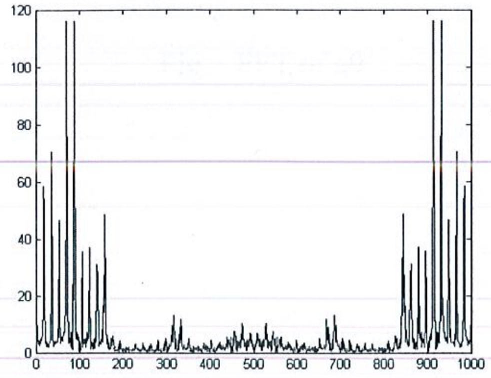


Fig. - FFT of a3

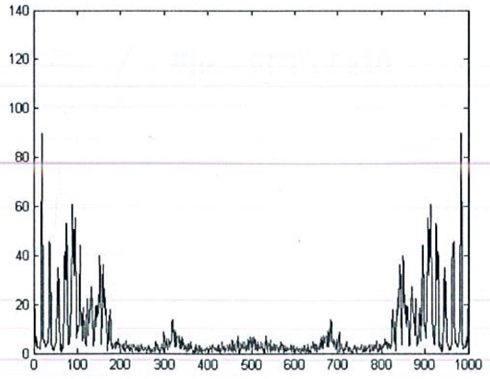


Fig. - FFT of a4

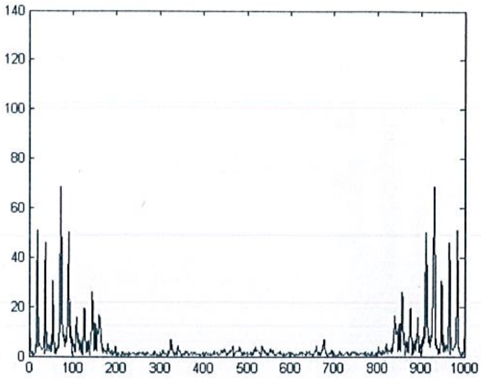


Fig. - FFT of a5

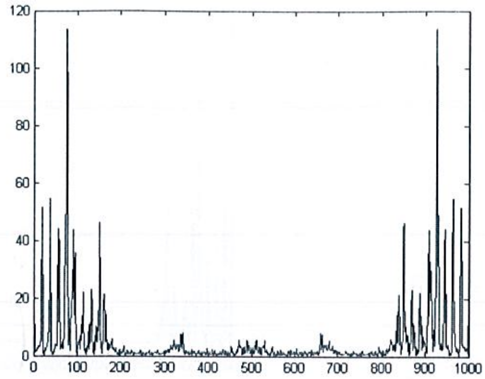


Fig. - FFT of a6

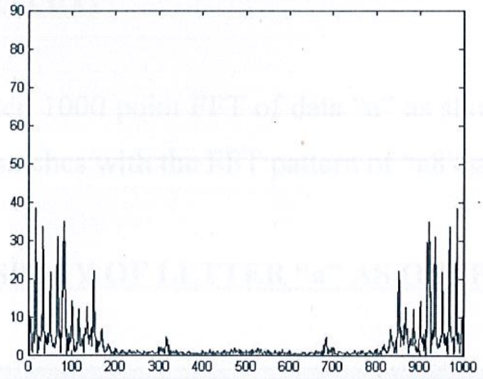


Fig. - FFT of a7

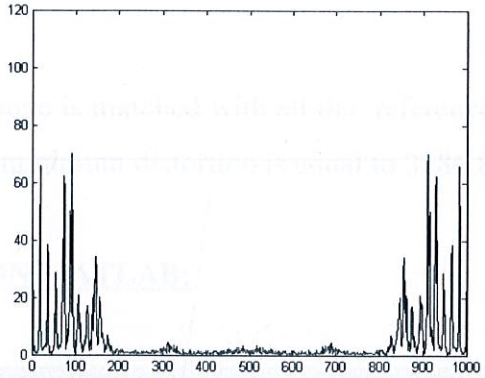


Fig. - FFT of a8

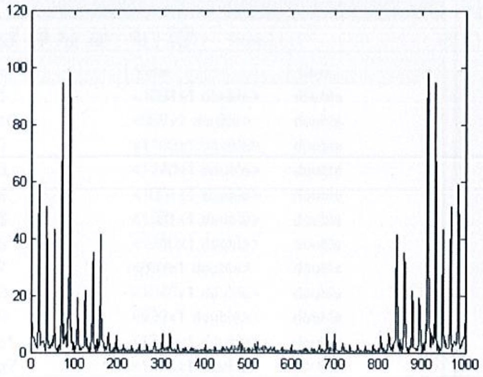


Fig. - FFT of a9

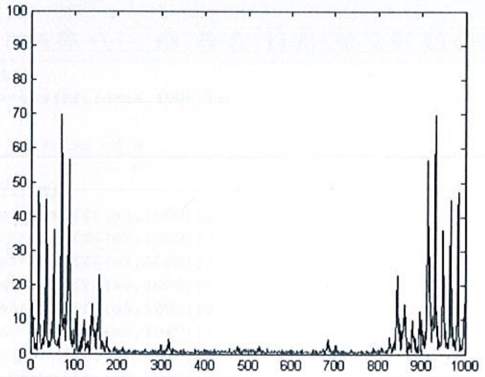


Fig. - FFT of a10

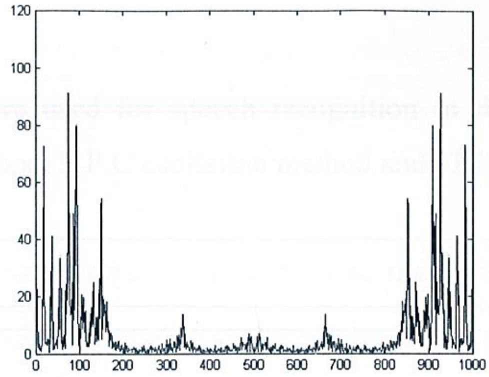
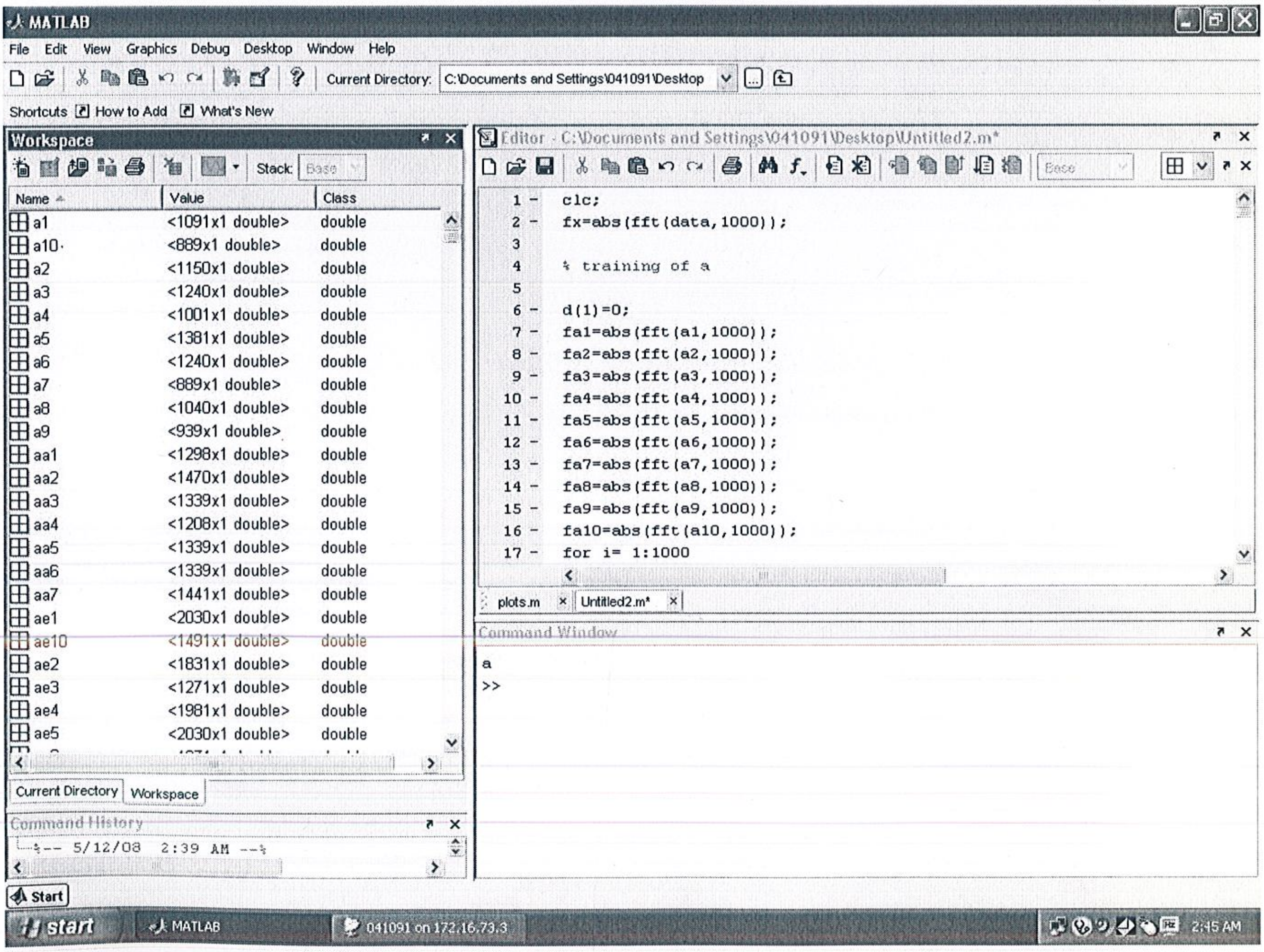


Fig. - Data of “a” to be matched

OUTPUT:

When 1000 point FFT of data “a” as shown above is matched with all the reference patterns it matches with the FFT pattern of “a8” giving minimum distortion is equal to 3586.8.

DISPLAY OF LETTER “a” AS OUTPUT ON MATLAB:



MATLAB
File Edit View Graphics Debug Desktop Window Help
Current Directory: C:\Documents and Settings\041091\Desktop
Shortcuts How to Add What's New
Workspace
Stack: Base
Editor - C:\Documents and Settings\041091\Desktop\Untitled2.m*
plots.m
Untitled2.m*
Command Window
Command History
Current Directory
Workspace
Start
start
MATLAB
041091 on 172.16.73.3
2:45 AM

Name	Value	Class
a1	<1091x1 double>	double
a10	<889x1 double>	double
a2	<1150x1 double>	double
a3	<1240x1 double>	double
a4	<1001x1 double>	double
a5	<1381x1 double>	double
a6	<1240x1 double>	double
a7	<889x1 double>	double
a8	<1040x1 double>	double
a9	<939x1 double>	double
aa1	<1298x1 double>	double
aa2	<1470x1 double>	double
aa3	<1339x1 double>	double
aa4	<1208x1 double>	double
aa5	<1339x1 double>	double
aa6	<1339x1 double>	double
aa7	<1441x1 double>	double
ae1	<2030x1 double>	double
ae10	<1491x1 double>	double
ae2	<1831x1 double>	double
ae3	<1271x1 double>	double
ae4	<1981x1 double>	double
ae5	<2030x1 double>	double

```
clc;
fx=abs(fft(data,1000));

% training of a

d(1)=0;
fa1=abs(fft(a1,1000));
fa2=abs(fft(a2,1000));
fa3=abs(fft(a3,1000));
fa4=abs(fft(a4,1000));
fa5=abs(fft(a5,1000));
fa6=abs(fft(a6,1000));
fa7=abs(fft(a7,1000));
fa8=abs(fft(a8,1000));
fa9=abs(fft(a9,1000));
fa10=abs(fft(a10,1000));
for i= 1:1000
```

```
a
>>
```

```
%-- 5/12/08  2:39 AM --%
```

CONCLUSION

The four methods that were used for speech recognition in this project are – Cepstrum method, L.P.C cepstral method, L.P.C excitation method and FFT method.

Cepstrum method was not found to be so useful because the coefficient obtained after vector quantization was not enough distinct to differentiate letter on that basis.

In L.P.C cepstral method, distortion calculated for different letters was not enough distinct to differentiate between two different letters.

L.P.C. excitation method requires optimization of linear prediction coefficients to minimize residual signal energy so that it approximately represents the excitation signal, which is the trickiest part in the method and it is very complicated too.

Among all four methods, FFT method is the most efficient and easiest one.
Speech recognition of Hindi letters was successfully done using FFT method.

BIBLIOGRAPHY

Research Papers

- John Makhoul, Salim Roucos and Herbert Gish - *Vector Quantization in Speech Coding*, november 1985. pages 1551-1557.
- Vance Faber - *Clustering and the Continuous k-Means Algorithm*, Number 22 1994. pages 138-143.
- Andrew W. Moore - *K-means and Hierarchical Clustering.*
- Dr. E. Gopinathan - *Residual excited linear predictive (relp) vocoder system and speaker independent isolated word recognition.* pages 20-28.
- Picone – *Continuous speech recognition using hidden markov models.*

Books

- Ben Gold and Nelson Morgan - *Speech and audio signal processing*, John Wiley & Sons, Inc. New York, NY, USA, 1999.
- Lawrence Rabiner and Biing-Hwang Juang - *Fundamentals of speech recognition*, Prentice-Hall, Inc.,1993.
- John G.Proakis and Dimitris K.Manolakis - *D.S.P. Principles, Algorithms and Applications*, Prentice Hall, 1996.
- Lawrence Rabiner - *Theory and application of D.S.P.*, Prentice Hall, 1975.
- Simon Haykins - *Communication Systems*, 4th edition, McMaster Univ.
- A.V.Oppenheim and A.S.Willsky and I.T.Young - *Signal and systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

Web Pages

- www.google.com
- www.matworks.com
- www.dsprelated.com
- www.mathcentral.com
- www.ieeexplore.com