SPEECH RECOGNITION APPROACHES


**CHAPTER 5**

Chapter - 5. Speech Recognition approaches

Table of Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Linear Predictive Coding</td>
<td>47</td>
</tr>
<tr>
<td>5.2.1 Fundamentals of Linear Predictive Coding</td>
<td>47</td>
</tr>
<tr>
<td>5.2.2 LPC Model</td>
<td>52</td>
</tr>
<tr>
<td>5.2.3 Generalized LPC Algorithm</td>
<td>53</td>
</tr>
<tr>
<td>5.2.4 Formants and Smooth spectra</td>
<td>59</td>
</tr>
<tr>
<td>5.2.5 Basics of LP Analysis</td>
<td>61</td>
</tr>
<tr>
<td>5.2.6 Implementation</td>
<td>64</td>
</tr>
<tr>
<td>5.2.7 Pre Emphasis</td>
<td>66</td>
</tr>
<tr>
<td>5.2.8 Frame Blocking and Windowing</td>
<td>67</td>
</tr>
<tr>
<td>5.2.9 LPC Analysis</td>
<td>68</td>
</tr>
<tr>
<td>5.1.10 Pitch Calculation</td>
<td>69</td>
</tr>
<tr>
<td>5.2.11 LPC Advantages</td>
<td>71</td>
</tr>
<tr>
<td>5.2.12 LPC Disadvantages</td>
<td>71</td>
</tr>
<tr>
<td>5.2.13 Simulation and Results of LPC</td>
<td>72</td>
</tr>
<tr>
<td>5.3 Mel Frequency Cepstral Coefficients</td>
<td>73</td>
</tr>
<tr>
<td>5.3.1 Speech Recognition Using MFCC</td>
<td>74</td>
</tr>
<tr>
<td>5.3.2 MFCC and its Calculation</td>
<td>76</td>
</tr>
<tr>
<td>5.3.3 Mel-Scale: from Auditory Modeling</td>
<td>76</td>
</tr>
<tr>
<td>5.3.4 Cepstral Analysis</td>
<td>78</td>
</tr>
<tr>
<td>5.3.5 Filter-Bank Interpretation</td>
<td>81</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>5.3.6</td>
<td>Implementation</td>
</tr>
<tr>
<td>5.3.7</td>
<td>Pre-Emphasis</td>
</tr>
<tr>
<td>5.3.8</td>
<td>Framing</td>
</tr>
<tr>
<td>5.3.9</td>
<td>Hamming Windowing</td>
</tr>
<tr>
<td>5.3.10</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>5.3.11</td>
<td>Mel Filter Bank Processing</td>
</tr>
<tr>
<td>5.3.12</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>5.3.13</td>
<td>Delta Energy and Delta Spectrum</td>
</tr>
<tr>
<td>5.3.14</td>
<td>MFFC Merits</td>
</tr>
<tr>
<td>5.3.15</td>
<td>Demerits of MFCC</td>
</tr>
<tr>
<td>5.4</td>
<td>Introduction to Wavelets</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Wavelet Analysis</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Wavelet Transform</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Wavelet Family</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Types of Wavelets</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Haar Wavelet</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Daubechies-N Wavelet Family</td>
</tr>
<tr>
<td>5.4.7</td>
<td>Discrete Wavelet Transform</td>
</tr>
<tr>
<td>5.4.8</td>
<td>Advantages of Wavelet analysis over STFT</td>
</tr>
<tr>
<td>5.4.9</td>
<td>Multilevel Decomposition of Signal</td>
</tr>
<tr>
<td>5.4.10</td>
<td>Wavelet Decomposition</td>
</tr>
<tr>
<td>5.4.11</td>
<td>Choice of Wavelet</td>
</tr>
<tr>
<td>5.3.12</td>
<td>Implementation</td>
</tr>
<tr>
<td>5.4.13</td>
<td>Preprocessing</td>
</tr>
<tr>
<td>5.4.14</td>
<td>Preemphasis</td>
</tr>
<tr>
<td>5.4.15</td>
<td>Frame Blocking &amp; Windowing</td>
</tr>
</tbody>
</table>
5.4.16 Frame Blocking 110
5.4.17 Windowing 111
5.4.18 Feature Extraction 112
5.5 Dynamic Time Warping 115
   5.5.1 DTW Algorithm 117
   5.5.2 DP-Matching Principle 120
   5.5.3 Discussions on Weighting Coefficient 123
5.6 Hidden Markov Model 125
   5.6.1 Recognition 127
   5.6.2 Optimal State Sequence 127
   5.6.3 Adjustment 127
   5.6.4 HMM – The Training of a Model of a Word – The Re-Estimation Problem 128
   5.6.5 Principal Cases of HMM 130
   5.6.6 Signal – The Utterance 131
   5.6.7 MFCC – Mel Frequency Cepstrum Coefficients 131
   5.6.8 Initialization 133
   5.6.9 Multiple Utterance Iteration 134
   5.6.10 Forward Algorithm 136
   5.6.11 Backward Algorithm 140
   5.6.12 Observation Sequence Probability Storing 143
   5.6.13 Simulation results of HMM 144
   5.6.14 Recognition Accuracy 145
5. SPEECH RECOGNITION APPROACHES

5.1 INTRODUCTION

In this proposed work the author implemented Linear Predictive Coding and Mel Frequency Cepstral Coefficients for extracting features of speech signals. In addition to these wavelet analysis is also used. After extracting the features, speech signals are applied to Speech Recognizer. Here author implemented two techniques like Dynamic time warping and Hidden Markov Model for recognizing telugu words.

5.2 LINEAR PREDICTIVE CODING

This method is an analytical one which is introduced during 1960’s and is used for knowing the present sample of the speech signal based on the set of past samples. This method is one of the best ways for synthesizing the accurate speech signal. It can be considered as one of the quick analysis algorithm which takes very less bandwidth for the encoded speech signals.

5.2.1 FUNDAMENTALS OF LINEAR PREDICTIVE CODING

Predominantly there exists two different ways for estimating the sound spectrum envelope. First way is by Fast Fourier Transform (FFT) method which gives good representation of the spectrum by sampling at equal intervals of amplitude values at equally spaced frequency points. Linear Predictive Coding, (LPC) is the second
method in which the linear image of the sound spectrum is formed by measuring the envelope of the overall spectrum. Each has its own benefits. Since, speech signal has rapid variations in time domain LPC gives good estimation of spectrum rather FFT. Usually LPC models the spectrum whereas FFT estimates the spectrum. LPC is considered\[25\] to be as the best way of obtaining spectrum of a low bit rate good quality speech signal. It is the most efficient method for estimating the speech parameters accurately. In LPC the nth sample of the speech signal is represented as the weighted sum of past samples.

\[
\hat{S} = \sum_{k=1}^{P} a_k * s[n-k] \tag{5.1}
\]

The order of the LPC is indicated by P. We can predict the exact value of nth sample when P approaches infinity.

To get accurate results p is taken in the range of 10-20. The error signal or LPC residual is given in equation 5.2 which is,

\[
e[n] = S[n] - \hat{S}[n] = S[n] - \sum_{k=1}^{P} a_k * S[n-k] \tag{5.2}
\]

Applying Z transform to equation (5.2) we have

\[
E[n] = S[z] - \sum_{k=1}^{P} a_k * S[z] * Z^{-k} \tag{5.3}
\]

\[
E[z] = S[z] * \left[ 1 - \sum_{k=1}^{P} a_k \right] \tag{5.4}
\]

\[
E[z] = S[z] * A[z] \tag{5.5}
\]
Thence, from the product of transfer function $A[Z]$ and the speech signal $S[Z]$ we can calculate the error signal. The error signal is framed as an all-zero digital filters, in which the coefficients $a_k$ represents zeros in the Z-plane of the filter. The product of the error signal $E[Z]$ and transfer function $1/A[Z]$ gives the original speech signal $S[Z]$ which can be shown as.

$$S[z] = E[z] \frac{1}{A(z)} \quad (5.6)$$

$1/A[Z]$ is the transfer function which represents an all pole digital filter, in which the coefficients $a_k$ were taken as the Z-plane poles. The transfer function $A[z]$ roots must lie inside the unit circle for the system to be stable.

For voiced and unvoiced signals there exist variations in the spectrum. Vocal cord vibrations produce voiced sounds, whereas there exist high frequency and less energy content in the unvoiced sound signals. Moreover voiced sound spectrum has the inherent behavior of having repetitive nature with some fundamental frequency which does not exists in unvoiced signals. The fundamental frequency of the voiced signal is called pitch.

By using a single or multi level sampling system, LPC digitally encodes the analog signals by estimating the each signal sample as a linear function of quantized values of past samples. Since LPC and adaptive predictive coding (APC) use adaptive predictors both are
considered to be as alike. To get benefit of the low bit rate, more number of prediction coefficients were used by LPC compared to APC. Because of this reason complex processor is used in LPC. Usually, speech signal takes 8 bits after sampling at 8 KHz. Therefore, the data processing rate would be 64000 bits/second. LPC algorithm uses compression algorithm to reduce the data rate to 2400 bits/second. LPC does the speech is broken into segments and also routed as voiced or unvoiced data, the frequency period, and the coefficients with the filter which represents the particular vocal tract for each and every segment.

At 2400 bits/sec of bit-rate, the speech incorporates a distinctive man-made sound along with noticeable loss of quality of compressed sound. But still we can listen the speech in an understandable way. Since some information loss takes place in linear predictive coding it is called lossy compression. The speech signal analyzed by LPC by estimating the formats, and eliminates their changes from the speech signal, and evaluating the intensity and frequency of the remaining buzz. Formants frequencies are the frequencies at which resonant peaks occur. The number that describes the formants and the remaining can be stored or transmitted somewhere else.

When the predictor filter may be adjusted to predict the input at best it can do so from the immediate preceding samples. This difference involving the input speech as well as the predictor result
(known as residual) will have roughly flat spectrum. The spectral peaks caused by the resonance of speech production will have to be removed. For the same reason, the total filtering process might possibly be termed as inverse filtering. That is the speech is synthesized by LPC by reversing the process. LPC uses the residue to create a source signal and used the formats to create an all-pole filter and runs the source via the filter resulting in speech. Since the speech signal differ with time, this operation is done on short channels of the speech signal known as frames, as the speech signal varies with time. Usually 30-50 frame per second gives intelligible speech with good compression. The figure 5.1 explains how the Predictor Filter is used in LPC systems.

![Diagram](image)

**Fig. 5.1: Predictor Filter used in Analyzer of LPC Systems**
5.2.2 LPC MODEL

Fig. 5.3: LPC Model

Controlled by transmitter’s Predictor coefficients

Excitation signal, generally residual signal

Speech output

Analysis / encoding (done by transmitter)

Synthesis/decoding (done by receiver)
5.2.3: GENERALIZED LPC ALGORITHM

In LPC, the input signal is sampled, broken into segments/blocks/frames, analyzed and then transmitted to the receiver.

Fig. 5.4: Generalized LPC Algorithm
A process is designed to increase, within a range of frequencies, the magnitude of some frequencies with respect to the dimensions of other (lower) frequencies so as to boost the overall signal-to-noise ratio by decreasing the address effects of such phenomenon as attenuation distortion or saturation recording media in consequent parts of the system.

It is truly computationally intensive to determine the pitch interval for proved segment involving speech throughout speech examination. There are several algorithms to compute this. When K is equivalent to the pitch function, r(k) will have a maximum value.

From equation (5.2),

\[ e[n] = S[n] - \hat{S}[n] = S[n] - \sum_{k=1}^{p} a_k * S[n-k] \]  

The sum the squared error being minimized will be expressed as:

\[ E = \sum_{n} e^2[n] = \sum_{n} \left( S[n] - \sum_{k=1}^{p} a_k * S[n-k] \right)^2 \]  

(5.8)

After calculating the derivation of E with respect to ai by making use of chain rule and equating it to zero, one can obtain

\[ a_1 * \sum_{n} S[n-k] * S(n-1) + a_2 * \sum_{n} S[n-k] * S(n-2) + \ldots \ldots + a_m * \sum_{n} S[n-k] * S(n-m) \]

\[ = \sum_{n} (S[n] * S[n-k]) \text{ for } k = 1, 2, 3, 4, 5, \ldots, M \]  

(5.9)
Equation (5.9) can be easily expressed in matrix form as:

\[
\begin{bmatrix}
  r(0) & r(1) & \cdots & r(m-2) & r(m-1) \\
  r(1) & r(0) & \cdots & r(m-3) & r(n-2) \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  r(m-2) & r(m-3) & \cdots & r(0) & r(1) \\
  r(m-1) & r(m-2) & \cdots & r(1) & r(0)
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a(m-1) \\
  a_m
\end{bmatrix}
= 
\begin{bmatrix}
  r(1) \\
  r(2) \\
  \vdots \\
  r(m-1) \\
  r(m)
\end{bmatrix}
\]

where,

\[r(k) = \sum_{n=0}^{N-k} S(n) \ast S(n+k)\]  \hspace{1cm} (5.10)

\[a = R^{-1} \ast r\]  \hspace{1cm} (5.11)

To solve equations (5.10) and (5.11), any efficient matrix solving methods like Decomposition, Gauss Elimination, Levinson-Durbin recursive method, Cholesky Decomposition etc. can be used.

Of the above methods for solving matrix, the Levinson-Durbin method is very efficient because it just needs M2 multiplications to assist compute that linear prediction coefficients. From equation (5.8), the sum of squared error can be expressed as:

\[E = \sum_{n} e^2[n] = \sum_{n} \left( S[n] - \sum_{k=1}^{p} ak \ast S[n-k] \right)^2\]  \hspace{1cm} (5.12)

The above equation can be re-written for M\textsuperscript{th} order precision,

\[E_m = \sum_{n} S[n] \ast E[n] - \sum_{n} \left( \sum_{k=1}^{p} ak \ast S[n-k] \right) \ast E[n]\]  \hspace{1cm} (5.13)
Solving above equation we get:

\[ E_m = r(0) - a_1 * r(1) - a_2 * r(2) - \ldots - a_{m-1} = r(0) \sum_{k=1}^{M} a_k * r(k) \]  \hspace{1cm} (5.14)

- When \( m = 0 \), \( E_0 = r(0) \)
- When \( m = 1 \), \( E_1 = r(0) - a_{11} * r(1) \)
- So, \( a_{11} = r(1) / r(0) = K_1 \) where \( K_1 \) is termed as reflection coefficient.

Now \( |K_1| < 1 \) and as \( r(1) < r(0) \) \( \rightarrow E_1 = r(0) [1 - K_1^2] \)

From above equation, it can be concluded that the prediction error \( E_1 \) is always less than \( E_0 \).

- When \( m = 2 \), the following recursion is performed

  (i) \( q_m = r(m) - \sum_{i=1}^{m-1} a_i (m-1) * r(m-i) \)

  (ii) \( q_m = \frac{q_m}{E_{m-1}} \)

  (iii) \( a_{nm} = K_m \)

  (iv) \( a_{im} = a_{i(m-1)} - K_m * a_{(m-1)(m-1)} \) \hspace{0.5cm} \text{for} \ i = 1, \ldots, m-1

  (v) \( E_m = E_{m-1} * \left[1 - K_m^2\right] \)
(vi) If $m < M$, then increase $m$ to $m+1$ and go to (i). If $m = M$, then stop.

**Pitch Estimation:** Pitch can be estimated by auto-correlation method, average magnitude difference method and cepstrum. I have used auto correlation method in my code.

The autocorrelation of a stationary sequence $x(n)$ is defined as:

$$R_s(\tau) = [x(n)^* \cdot x(n + \tau)] = \frac{1}{N} \sum_{n=0}^{N-1} x(n)^* \cdot x(n + \tau)$$

here $\tau$ is termed as lag. An autocorrelation is the average correlation between two samples from one signal and its $\tau$ samples delayed signal. In MATLAB, inbuilt function “xcorr” can be used either for cross correlation between two signals or auto correlation between a signal with itself.

The speech signal is examined by the analysis/encoding of LPC by breaking it down into segments or blocks. Each segment is then examined further to find

1. voiced / unvoiced segment

2. pitch information important for this particular segment

By obtaining the linear conjecture coefficients and residual mistake sequence one can easily reconstruct the speech signal by making use of synthesis filtration system. LPC synthesis is conducted
by the actual receiver when using the answers received to develop a filtering. The original speech signal is reproduced once the correct insight source is usually provided, LPC functionality tries to help imitate people speech production.

LPC is comparable to cepstral analysis but uses quite different methods. LPC is employed in separating out the particular impact involving source in addition to filter from a speech indication. LPC is often a coding technique a method of encoding the information in some sort of speech signal in a smaller place for broadcasting over a prescribed channel. LPC encodes a signal by recognizing a set of weights about past indication values that could expects the following signal importance.


When the values for a \{1,...,3\} are found such that e[n] is very short for a period of speech, we can easily send only the signal values within the window. The speech frame can be rebuilt at the other end by predicting subsequent values from past samples and e[n] signal. By this we can find out the values of a[1,...,k] but there are couple of algorithms which can do this. The result of LPC analysis is a set of coefficients a [1,...,k] and problem signal e[n], the problem signal is usually a very small and symbolizes the difference between the expected signal and also the original signal. There is a certain
analogous among the LPC equation understanding that of a recursive filtration system \( y^* a = x \):

\[
\]

The linear predictive coefficients from \([1 \ldots k]\) were related to the recursive filter whereas \( e[n] \) is the error signal which relates to the source. The reduction in the error of LPC results in malfunction signal which has smooth spectrum. Hence the error signal is considered to be close to both an impulse train and a white noise signal. This is an exact match to source filter model of speech origination where we modify a vocal tract filter with a voiced signal or a noise source. LPC analysis can find the coefficient of a filter which converts noise or an impulse train into original frame of speech.

The result is not perfect. The filter coefficients which are derived by LPC analysis contain information of the glottal source filter, the lip radiation or pre emphasis filter and the vocal tract. we will discover fewer variable compared to the vocal region filter we are able to factor these in practice .

5.2.4 FORMANTS AND SMOOTH SPECTRA

We should familiar with Z-transforms to cover LPC analysis as LPC is really a way in to some more attractive signal analysis methods.
LPC coefficients were used to model the original speech that is produced by the vocal tract. The spectrum obtained from these LPC coefficients gives the characteristics of the actual vocal tract without altering the source spectrum. An example of taking the actual spectrum with filter is by replacing with exp (iwk) for Z inside the Z transform with signal or perhaps by sending an impulse via the actual filter in addition to taking the DFT. The output can be quite useful in signal analysis by either of the ways.

We can clearly see the format peaks they tend to be much better defined than in a cepstrally smoothed spectrum by looking about an LPC smoothed variety of voiced speech. By making use of Z-transform notation to search for the places of such format peaks to get a given number of LPC coefficients analogous on the points where A (Z) is actually zero. This is the key to self activating format tracking of speech signals which derives the LPC coefficient, solve the Z-transform equation and note the resulting format positions. Deplorably since the LPC model isn’t a correct fit to real speech production this method will derive spurious formants.

LPC is a powerful signal modeling technique and important in speech recognition and speech analysis and LPC coefficients are used to produce cepstral coefficients and area functions.

Linear prediction is one of the widely used powerful signal analysis method. LPC models and estimates the basic parameters of
speech efficiently. In LPC group of parameters or predictor coefficients is
decided by minimizing the sum of squared differences between the actual speech samples and predicted values.

These kinds of coefficients form the cornerstone for LPC of speech. The analysis offers the capability regarding computing this linear prediction type of speech with time. Cepstral coefficients which are considered to be as the powerful set is obtained from the predictor coefficients of LPC.

5.2.5 BASICS OF LP ANALYSIS

In the particular LP research the redundancy within the speech signal is used. The expected sample may be represented as \( s(n) \) will be the windowed talk sequence in addition to \( a_k \) will be the linear prediction coefficients which can be produced simply by multiplying limited time period speech frame which has a hamming or perhaps similar kind of window which can be given simply by \( 'n' \) will be the windowing sequence. The prediction of real sample as being a linear association of beyond \( p \) samples form the basis of linear conjecture analysis exactly where \( p \) will be the order of prediction. The expected sample may be represented \( \hat{s}(n) \) the following

\[
\hat{s}(n) = - \sum_{k=1}^{p} a_k \cdot s(n-k)
\]  

(5.15)
where \( a_k \) will be the linear conjecture coefficients in addition to \( s(n) \) will be the windowed talk order made by multiplying limited time speech frame with a hamming or similar sort of window which can be given simply by,

\[
s(n) = x(n) \cdot \omega(n)
\]  

(5.16)

here \( \omega(n) \) will be the windowing order. The conjecture error \( e(n) \) can be calculated with the difference between actual small sample \( s(n) \) and the particular predicted small sample \( s^\wedge(n) \) which can be given simply by,

\[
e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{p} a_k \cdot s(n-k)
\]  

(5.17)

\[
e(n) = s(n) - \hat{s}(n)
\]  

(5.18)

\[
e(n) = s(n) + \sum_{k=1}^{p} a_k \cdot s(n-k)
\]  

(5.19)

The key aim associated with LP analysis should be to calculate the LP coefficients which in turn minimizes the prediction problem \( e(n) \). Least squares auto correlation method would be the popular method employed for calculating the LP coefficients. This is acquired by simply reducing the entire prediction problem. The complete prediction error might be represented as follows,

\[
E = \sum_{n=-\infty}^{\infty} e^2(n)
\]  

(5.20)
This is often expanded with the equation (5.20) as follows.

\[
E = \sum_{n=\infty}^{\infty} \left[ s(n) + \sum_{k=1}^{p} a_k \cdot s(n-k) \right]^2
\]  

(5.21)

The particular values regarding \( a_k \) which in turn minimize the total prediction malfunction \( E \) may be computed through \( \frac{\partial E}{\partial a_k} \) and also equating to zero pertaining to \( k=0, 1, 2, \ldots p \). \( \frac{\partial E}{\partial a_k} = 0 \) For every single \( a_k \) provide \( p \) linear equations having \( p \) unknowns. Caused by which increases the LP coefficients. This can be represented as following.

\[
\frac{\partial E}{\partial a_k} = \frac{\partial}{\partial a_k} \cdot \sum_{n=\infty}^{\infty} \left[ s(n) + \sum_{k=1}^{p} a_k \cdot s(n-k) \right]^2 = 0
\]  

(5.22)

The actual differentiated expression might be written as

\[
\sum_{n=\infty}^{\infty} s(n-i) \cdot s(n) = \sum_{k=1}^{p} a_k \cdot \sum_{n=\infty}^{\infty} s(n-i) \cdot s(n-k)
\]  

(5.23)

Where \( i=1, 2, 3, \ldots p \). The equation (5.23) could be written in terms of autocorrelation sequence \( R(i) \) as follows,

\[
\sum_{k=1}^{p} a_k R(i-k) = R(i)
\]  

(5.24)

Intended for \( i=1,2, 3, \ldots p \). Where the particular autocorrelation sequencer used in equation (5.23) may be written as follows

\[
R(i) = \sum_{n=\infty}^{N-i} s(n) s(n-i)
\]  

(5.25)
Regarding \( i = 1, 2, 3 \ldots p \) and \( N \) is the length of the routine

This is mostly shown in the matrix form as follows

\[
R \cdot A = -r
\]

Where by \( R \) may be the \( p \times p \) symmetric matrix which includes elements. \( R(i, k) = R(|i-k|), \ (1 \leq i, k \leq p) \).
\( r \) is usually a column vector employing elements \( (R(1), R(2), \ldots, R(P)) \) and then \( A \) could be the column vector which includes LPC coefficients. \( (a(1), a(2), \ldots, a(p)) \). It may possibly be shown that \( R \) is commonly toeplitz matrix which could be denoted as

\[
R = \begin{bmatrix}
R(1), R(2), R(3), \ldots, R(P) \\
R(2), R(1), R(2), \ldots, R(P-1) \\
R(3), R(2), R(1), \ldots, R(P-2) \\
\vdots \\
R(P), R(P-1), R(P-2), \ldots, R(1)
\end{bmatrix}
\]

(5.26)

Linear Predictive coefficients might be computed as shown.

\[
A = -R^{-1} \cdot r
\]

where \( R^{-1} \) could be the inverse of the matrix \( R \).

**5.2.6 IMPLEMENTATION**

The basic steps of LPC processor include the following:

**LPC Algorithm**

- Windowing and pre-emphasis filtering are the main blocks of the algorithm.
- Computing the autocorrelation
For processing the speech using LPC a block with 240 samples size is used for the speech which has the sampling frequency of 8KHz. The blocks that are processed will gets overlapped with 80 samples. Hence it is needed to extract the characterizing parameters of 240 samples with 20msec each using LPC algorithm. These segments of 240 samples were passed through the high pass filter with Hamming window in the segmentation processing block. The resultant samples are then passed through the blocks of pitch detection and autocorrelation. In processing unit the silent speech blocks were also detected. For 11 different offsets the autocorrelation processing block computes the autocorrelation for 240 samples. After that, to solve the set of 10 linear equations the Levinson-Durbin algorithm is used. These linear equations depend on the sequence autocorrelation and the solution is the group of feedback coefficients $a_l's$. By using Levinson-Durbin algorithm one can get the transfer function gain $G$. The Levinson-Durbin algorithm is one of the recursive algorithms that involve manipulation of various numerical particularly during the calculation of division. To obtain the characteristics of voiced and unvoiced speech signals these 240 windowed samples are processed. These are first sent to LPF with 25 taps of FIR filter. Then using 3-level center clipping the signal is clipped. For 60 different sequence offsets the autocorrelation of the clipped signal is computed. Finally
the voice or unvoiced parameters of the speech signal are extracted from these autocorrelation results.

5.2.7 PRE EMPHASIS

Usually between vowels and consonants there will be a large variation in the spectrum of a phoneme or word. Since vocal tract is configured to give rise to low frequency resonances for vowels, it has higher amplitudes. On the other hand, consonants are just sudden bursts of air from the mouth or similar to the turbulent airflows, that invokes to have noise like flat spectrum with high-frequency content.

In general there exists low frequencies with large amplitudes and high frequency contents with small amplitudes in conversational speech. By making use of first order pre-emphasis FIR filter the difference in amplitudes is reduced and the spectrum is flattened. The transfer function for such a filter is given as

\[ H(z) = 1 - az^{-1}, \quad a = 0.97 \]  

(5.27)

this is equivalent to the difference equation in the discrete-time domain which is given as,

\[ y(n) = x(n) - ax(n-1) \]  

(5.28)

Where \( x(n) \) is considered as the original speech signal and \( y(n) \) is pre-emphasized signal.
Two benefits exist of this filter.

- Firstly, the negative spectral slope of different voiced sections is boosted which intern improves the speech analysis efficiency.

- Secondly, it amplifies the sensitive high frequency content which is of high sensitive in hearing. This leads to model an important perceptual aspect of the auditory system.

### 5.2.8 FRAME BLOCKING AND WINDOWING

Speech is considered as non-stationary signal because of the variations in phoneme’s spectral features, variations in prosody, and random variations in the vocal tract. But, in short time interval in the order of 10 to 20msec duration it can be conveniently assumed as stationary. Hence it can be analyzed over these shot-time windows. So in frame blocking the speech signal is divided into frames of \( N \) samples that overlap with adjacent frames for \( M \) samples.

While blocking each frame of speech signal each frame is multiplied with hamming window to minimize the spectral distortions. The Hamming window is of the form

\[
    w(n) = 0.54 - 0.64 \cos \left( \frac{2\pi n}{N-1} \right), \quad 0 \leq n \leq N-1
\]  

(5.29)

Where \( N \) is the duration (in samples) of the speech frame. The output of the signal after windowing is \( y(n) \) which is given as
\[ y(n) = x(n) \ w(n) \] (5.30)

This particular windowing purpose acts as a low pass filter, enhancing the particular signal on the window.

Autocorrelation examination is the next step. To auto correlate every frame connected with windowed signal we get

\[ r_i(m) = \sum_{n=0}^{N-1-m} \tilde{x}_i(n) \tilde{x}_i(n + m); \quad M = 0, 1, \ldots, p \] (5.31)

Where p is taken as the order of LPC analysis.

### 5.2.9 LPC ANALYSIS

The next computation step could be the LPC analysis, which makes over each frame of \( p + 1 \) autocorrelations directly into LPC parameter set by employing Durbin’s technique. This may be shown in the following criteria

\[ E^{(0)} = r(0) \]

\[ r(i) = \sum_{j=1}^{i-1} \alpha_j^{(i)} r(|i - j|) \]

\[ k_i = \frac{\sum_{j=1}^{i-1} \alpha_j^{(i)} r(|i - j|)}{E^{i-1}} \quad 1 \leq i \leq p \] (5.32)

\[ \alpha_i^{(i)} = k_i \]

\[ \alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad 1 \leq j \leq i - 1 \] (5.33)

\[ E^{(i)} = (1 - k_i^2)E^{i-1} \] (5.34)
Solving these recursively through i=1, 2, ..., p the LPC coefficient, a, is

\[ a_m = a_m^{(p)} \]  

(5.35)

### 5.2.10 PITCH CALCULATION

One of major limitations of the autocorrelation representation is that in a sense it retains too much of the information in the speech signal. Avoiding this problem it can be again useful to process the actual speech signal to be able to make the actual periodicity more prominent while suppressing various other distracting options that come with the signal. This was the approach followed to permit the use of very simple pitch detector. Techniques which perform this type of operation on signal are sometimes called “spectral flatteners” since their objective is to remove the effects of the vocal tract transfer function, thereby bringing each harmonic to the same amplitude level as in the case of a periodic impulse train. There are numerous spectrum flattening techniques however, a technique called “center clipping” is the technique we used.

In the scheme, the center clipped speech signal is obtained by a nonlinear transformation

\[ Y(n) = C[x(n)] \]  

(5.36)
The operation of the center clipper is depicted in fig. (5.5) for this segment, the maximum amplitude, $A_{max}'$, is found and clipping level, $C_L$, is set equal to a fixed percentage of $A_{max}'$ (we used 30%). From the figure (5.5), it can be seen that for samples above $C_L$ the output of the center clipper is equal to the input minus the clipping level. However, first let us examine the effect of the clipping level. Clearly, for high clipping levels, fewer peaks will exceed the clipping level and thus fewer pulses appear in the output, and therefore, fewer extraneous peaks will appear in the autocorrelation function. Clearly, as the clipping level is decreased, more peaks pass through the clipper and thus the autocorrelation function becomes more complex.
5.2.11 LPC ADVANTAGES

- LPC gives better speech signal modeling
- Using low dimension feature vectors LPC gives the spectral envelope
- Linear characteristics were given by LPC
- Acceptable source-vocal tract separation is obtained by LPC
- It is analytically traceable model
- LPC is easy to implement in both software and hardware

5.2.12 LPC DISADVANTAGES

- Since human perception has variable frequency perception the speech model using LPC which assumes constant weighing for the entire spectrum results in the obscured results.
- Many models of LPC are highly correlated but, uncorrelated models are highly accurate invited in the speech analysis.
- Speech specific apriority information is not considered in the modeling using LPC.
5.2.13 SIMULATION RESULTS OF LPC

a) Analysis of LPC Before coefficients Applied To Recognition System

The following are the results obtained before applying to the speech recognition system these are the LPC coefficient graphs for voiced and unvoiced part of the input signal.

Fig. 5.6: Plot of signal and its predictor coefficients
b) Results below is after applying LPC coefficients to the system

Matlab simulation results of Recognition of Telugu digits using LPC, If we increase the LPC order the execution time of the system is less.

### 5.3 MEL FREQUENCY CEPSTRAL COEFFICIENTS

The specific LPC is approximately such as sufficient statistics of haphazard samples throughout statics. The LPC coefficients are actually the very least squares estimators from the regression coefficients, if the minimum variance linear estimators from the regression coefficients. The large data of the frame usually are well-represented from the LPC coefficients with the exception of LPC coefficients usually are too modest, i.e., the estimates from the regression coefficients are certainly not significant because weighed in
opposition to noise. But this concern is solved using MFCC right here more volume of coefficients are offered and large data frames is effectively represented.

5.3.1 SPEECH RECOGNITION USING MFCC

Usually, Automatic Speech Recognition (ASR) technique could be prepared throughout two blocks the particular element extraction plus the modeling period. Employed, the particular modeling period is subdivided throughout acoustical and as well language modeling, equally dependant upon HMMs as explained in Figure 5.10.

Fig. 5.8: Automatic Speech Recognition System

The features of speech signal extracted using MFCC is a non-invertible and has losses. Since MFCC actually has filter banks, its transformation doesn't ensure accurate reconstruction of speech signal. i.e., if we are provide with only the actual features, reconstruction of the original speech signal used to build these features is not possible (restoration). The leading speech we cannot able to get back.
The main reason for loosing of the information is the complex computations and not having good robustness. By defining more number of parameters one can improve the accuracy but by paying price of complexity. Even in such cases the robustness difficulties persists. The higher the amount of boundaries in a every model. Due to the ability to robustness concerns greater training sequence.

Since speech has segments in the range of 20 to 30msec for each frame, for window analysis 10msec of window is chosen which gives 12 MFCC’s\cite{25}. In addition to these MFCC’s normalized power parameters were also calculated. From these 39 parameters are estimated by calculating the first and second differentiation, in addition to the energy values. The module drives which are used to get the attributes of the speech provide 39 values for each 10msec with 8KHz sample rate of the speech.

The operations that are going to takes place here are similar to non-overlapped 80 speech samples that are symbolized with 39 numbers. The fact exists in considering the speech sample which is represented with 1 byte to 4 bytes which also contains the information of its attributes.

This is the parametric representation, and is very clear that the number of bytes has increased in representing the speech signal. If 16 kHz sampling rate is considered then it takes 160 samples to represent 39 parameters. If the sampling frequency is increased more than 16KHz it is very difficult to reconstruct the speech samples back.
Since our motive is not the dialog compression but for obtaining the accurate features that represents the speech signal.

5.3.2 MFCC AND ITS CALCULATION

The block diagram regarding calculating MFCCs is usually given underneath

![Fig. 5.9: MFCC block diagram](image)

There exists two ways for obtaining the MFCC coefficients. In the first method filter-banks were used to process and get the speech features. The second is the modified form of the existing deconvolution method of homomorphic processing called cepstral method. The obtained points are very intuitive and give good understanding in obtaining MFCC’s. The following sections give a detailed explanation.

5.3.3 MEL-SCALE FROM AUDITORY MODELING

Mel-scale is used for auditory modeling. Two famous tests that generated the bark and Mel scale machines were taken from the literature and its outcomes are reported here to characterize human auditory system.
Table 5.1: Characteristics of human Auditory system

<table>
<thead>
<tr>
<th>Filter</th>
<th>Frequency (KHz)</th>
<th>BW (KHz)</th>
<th>Frequency (KHz)</th>
<th>BW (KHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.050</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
</tr>
<tr>
<td>2</td>
<td>0.150</td>
<td>0.100</td>
<td>0.200</td>
<td>0.100</td>
</tr>
<tr>
<td>3</td>
<td>0.250</td>
<td>0.100</td>
<td>0.300</td>
<td>0.100</td>
</tr>
<tr>
<td>4</td>
<td>0.350</td>
<td>0.100</td>
<td>0.400</td>
<td>0.100</td>
</tr>
<tr>
<td>5</td>
<td>0.450</td>
<td>0.110</td>
<td>0.500</td>
<td>0.100</td>
</tr>
<tr>
<td>6</td>
<td>0.570</td>
<td>0.120</td>
<td>0.600</td>
<td>0.100</td>
</tr>
<tr>
<td>7</td>
<td>0.700</td>
<td>0.140</td>
<td>0.700</td>
<td>0.100</td>
</tr>
<tr>
<td>8</td>
<td>0.840</td>
<td>0.150</td>
<td>0.800</td>
<td>0.100</td>
</tr>
<tr>
<td>9</td>
<td>1.000</td>
<td>0.160</td>
<td>0.900</td>
<td>0.100</td>
</tr>
<tr>
<td>10</td>
<td>1.170</td>
<td>0.190</td>
<td>1.000</td>
<td>0.124</td>
</tr>
<tr>
<td>11</td>
<td>1.370</td>
<td>0.210</td>
<td>1.149</td>
<td>0.160</td>
</tr>
<tr>
<td>12</td>
<td>1.600</td>
<td>0.240</td>
<td>1.320</td>
<td>0.184</td>
</tr>
<tr>
<td>13</td>
<td>1.850</td>
<td>0.280</td>
<td>1.516</td>
<td>0.211</td>
</tr>
<tr>
<td>14</td>
<td>2.150</td>
<td>0.320</td>
<td>1.741</td>
<td>0.242</td>
</tr>
<tr>
<td>15</td>
<td>2.500</td>
<td>0.380</td>
<td>2.000</td>
<td>0.278</td>
</tr>
<tr>
<td>16</td>
<td>2.900</td>
<td>0.450</td>
<td>2.297</td>
<td>0.320</td>
</tr>
<tr>
<td>17</td>
<td>3.400</td>
<td>0.550</td>
<td>2.639</td>
<td>0.367</td>
</tr>
<tr>
<td>18</td>
<td>4.000</td>
<td>0.700</td>
<td>3.031</td>
<td>0.422</td>
</tr>
<tr>
<td>19</td>
<td>4.800</td>
<td>0.900</td>
<td>3.482</td>
<td>0.484</td>
</tr>
<tr>
<td>20</td>
<td>5.800</td>
<td>1.100</td>
<td>4.000</td>
<td>0.556</td>
</tr>
<tr>
<td>21</td>
<td>7.000</td>
<td>1.300</td>
<td>4.595</td>
<td>0.639</td>
</tr>
<tr>
<td>22</td>
<td>8.500</td>
<td>1.800</td>
<td>5.278</td>
<td>0.734</td>
</tr>
<tr>
<td>23</td>
<td>10.500</td>
<td>2.500</td>
<td>6.063</td>
<td>0.843</td>
</tr>
</tbody>
</table>
5.3.4 CEPSTRAL ANALYSIS

The method of Homomorphism processing is well given in literature. Because of the advantage of deconvolution Cepstral homomorphism processing is frequently used. To get the clear understanding it should be remembered that inside speech processing, a speech production modeling is used in the human speech production.

**Source**: It relates to the flow of air that comes from lungs during speech generation. For unvoiced sounds like in “s” as well as “f”, vocal cords are relaxed and the glottis is opened. For voiced sounds like, “a”, “e”, the vocal cord actually vibrates which in turn gives the pitch value.

**Filter**: The shaping of the spectrum is done by using filter. This helps in creating different sounds and relates towards vocal tract organs.

**Roughly speaking**: A good parametric representation to get a speech recognition system tries to reduce the influence on the source (the system must provide same “answer” to get a high pitch woman’s voice and for to secure a low pitch males voice), and specify the filter. the problem in source e(n) along with filter impulse answer h(n) are convoluted. Then we wish deconvolution in speech recognition applications.
Mathematically in time domain, convolution:

\[ e(n) * h(n) = x(n) \]  \hspace{1cm} (5.37)

In the frequency area multiplication: source x filter = speech.

\[ E(z) \cdot H(z) = X(z) \]  \hspace{1cm} (5.38)

employed in the frequency sector, use the logarithm so that we can transform the multiplication right into a summation (obs: \( \log ab = \log a + \log b \)) isn’t easy to separate (to filter) stuffs that are multiplied as in (5.38), but you can actually design filters to separate things that are shown below:

\[ C(z) = \log X(z) = \log E(z) + \log H(z) \]  \hspace{1cm} (5.39)

Develop of which \( H(z) \) will be primarily constructed by lowered frequencies and also \( E(z) \) has the majority of its power throughout increased frequencies, in a fashion that a uncomplicated low-pass filtering can distinct \( H(z) \) via \( E(z) \) in the fact if we are dealing \( E(z) + H(z) \). In fact, let suppose in the interests of simplicity we have, rather than (5.39), the following equation

\[ C_d(z) = E(z) + H(z) \]  \hspace{1cm} (5.40)

We’re able to use any linear filter to eliminate \( E(z) \) then calculate the actual \( Z \)-inverse transform to secure a time-sequence \( c_0(z) \).
Observe that in that instant, co(z) might have dimension of period (seconds, intended pertaining to example).

The particular log operation in (5.39) Log is a non-linear operation but it can "create" completely new frequencies. As an example, expanding this log of a cosine inside Taylor series ensures that harmonics were created. So, even if E(z) along with H(z) usually are well separated inside frequency domain, log E(z) along with log H(z) could sometime have significant overlap. Fortunately, that is incorrect in practice for conversation processing. The opposite point can be that, with the log operation, the Z-inverse involving C(z) are yet to the dimension of energy as in (5. 40). We call cepstrum this Z-inverse of C(z) and it is dimension can be frequency (a time domain parameter).

There are 2 basic varieties of cepstrum: complicated cepstrum and real cepstrum. In addition to, there are usually two ways of calculating the actual cepstrum (used within speech finalizing because phase seriously isn't important).LPC cepstrum as well as FFT cepstrum.

LPC cepstrum: The cepstral coefficients are made from coefficients cepstrum. The recommended parametric portrayal for conversation recognition may be the FFT cepstrum derived driven by a Mel scale.
5.3.5 FILTER-BANK INTERPRETATION

We go to frequency domain plus disregard phase, working only while using power array. Then, we work with log considering our ears works in decibels. In order to decrees dimensionality, we start using a filter-bank together with around 20 filtration system. The filtration system follows Mel-scale. We acquire the DCT-II since it is good.

The examples below shows the location where the MFCC did not capture the true formants design, i. e., they did not Perform wonderful job.

The specific MFCC is usually a representation determined because genuine cepstrum regarding any windowed short-time Signal in line with fast Fourier transform while using speech signal. In MFCC, a nonlinear consistency scale can be utilized, which approximates this kind of behavior while using auditory program. The discrete cosine transform while using real logarithm short-time energy spectrum expressed for this nonlinear consistency scale is named the MFCC. Alternatively, to produce a MFCC, we should discover the DFT of the frame while using huge facts and subsequent Mel independent out standard bank smooth the selection, performs this kind of inverse DFT about the logarithm from the magnitude regarding filter regular bank production. Alternatively, to create a MFCC, one has to discover the DFT of the frame from the huge facts and following Mel independent out banks even the selection, performs this kind of
inverse DFT on the logarithm from the magnitude regarding filter regular bank output.

5.3.6 IMPLEMENTATION

The particular extraction of the most effective parametric representation of acoustic signals is usually an most important task to make a good reputation performance. The efficiency of this phase is most important to following phase given it affects their behavior. MFCC is based on human experiencing perceptions which cannot comprehend frequencies earlier mentioned 1KHz. To put it differently, MFCC is based on known variation on the human ear’s vital bandwidth having frequency. MFCC includes two forms of filter which might be spaced linearly from low consistency below 1000 Hz as well as logarithmic spacing earlier mentioned 1000Hz. A fresh subjective pitch can be found on Mel Perseverance Scale in order to record important characteristic related to phonetic within talk. The complete technique on the MFCC is shown in Fig. 5.9.

5.3.7 PRE-EMPHASIS

Pre-emphasis represents a method created to increase, within a range of frequencies, the degree of many frequencies with regards to the magnitude with the others reduced frequencies to be able to improve the complete SNR. Consequently, the preceding step refers to the passing of signals by using a filter which often stresses greater frequencies.
5.3.8 Framing

The procedure of dividing the conversation samples that’s obtained by an ADC in to a short frame with all the length within the range regarding 20 to 40msec. This voice transmission is split up into supports of N samples. Neighboring frames are separated by means of M (M<N). Normal values utilized are M = 100 and also N= 256.

5.3.9 Hamming Windowing

By taking in to account the next segment with features thoughts and opinions chain and integrating all the closest regularity lines, Hamming window is utilized as window shape. The Hamming window is represented just as in the event the window is defined as.

\[ w(n), \quad 0 \leq n \leq N-1 \text{ where} \]

\[ N \text{ is the numeral of samples in each frame} \]

\[ Y[n] \text{ is the Output signal} \]

\[ X (n) \text{ is the input signal} \]

\[ W (n) \text{ is the Hamming window.} \]

then the effect of windowing signal is revealed below:

\[ Y[n] = X(n) \ast W(n) \quad (5.41) \]
5.3.10 FAST FOURIER TRANSFORM

FFT is needed to turn every frame of n samples from time period domain into rate of recurrence domain. If \( X(w) \), \( H(w) \) as well as \( Y(w) \) include the Fourier conversion of \( X(t) \), \( H(t) \) likewise \( Y(t) \) respectively, the Fourier Transform is needed to turn the convolution on the glottal beat \( U(n) \) plus the vocal pathway impulse reply \( H(n) \) from the time domain.

\[
Y(w) = FFT[h(t) \ast X(t)] = X(w) \ast X(w)
\]

If \( X(w) \), \( H(w) \) in addition to \( Y(w) \) would be the Fourier Enhance of \( X(t) \), \( H(t) \) and \( Y(t) \) respectively

5.3.11 MEL FILTER BANK PROCESSING

This frequency varies of FFT spectrum can also be so extensive and voice signal doesn’t fallow the linear degree. Every filtration system magnitude frequency response is usually triangular with nature and is also unity with the mid frequency and lower linearly to help zero in mid frequency of a pair of filters. The output of every filter is the sum its refined spectral factors. The next equation is utilized to compute the Mel for any given frequency f in hz.

\[
F(Mel) = \left[ 2595 \times \log_{10} \left( 1 + \frac{f}{700} \right) \right]
\]
5.3.12 DISCRETE COSINE TRANSFORM

By utilizing DCT here is the technique to transform the actual log Mel range into time domain. Mel coefficient is the effect of the transform. The group of coefficient is named acoustic vectors. Hence, each and every feedback utterance is transformed into a collection of acoustic vector.

5.3.13 DELTA ENERGY AND DELTA SPECTRUM

The frames along with the voice signal changes, like the shape connected with format with its transitions. There is often a need to incorporate features related to the alter in cepstral features. 13 delta as well as velocity capabilities (12 cepstral capabilities plus energy), and 39 features comes with a double delta feature are additional. The strength in a frame for just a signal x in a very window coming from time t1 to time t2, is usually represented shown under.

\[ \text{Energy} = \sum X[t] \]

Where \( X[t] \) = signal

All of the 13 delta capabilities represent the change between frames respectively or Cepstral attribute and every one of the features represent the transform between frames from the corresponding delta features.
5.3.14 MFFC MERITS

- The low cepstral coefficients were concentrated in defining the characteristics in the slow varying part of the speech signal.

- There is a weak correlation between the features of MFCC which leads to create a statistical acoustic model.

- MFCC have no linear characteristics.

- Mel scaling as demonstrated an ability better discrimination in between phones, which helps in recognition.

- It provides good discriminating properties. While calculating MFCC from power spectrum of the speech signal the phase spectrum is not considered.

- MFCC features are advantageous given it mimic a number of the human processing in the signal.

5.3.15 DEMERITS OF MFCC

- Compared to LPC, MFCC is more expensive in computational wise because of the usage of Fast Fourier Transform during the computation of its spectrum.

- The major drawback is MFCC coefficients are in time domain but not in frequency domain.
In noisy environments MFCC is prone to be effected. It gives an idea that, especially at low SNR, the MFCC has insufficient sound representation capability.

Two problems persists in Mel frequency Cepstral Coefficients (MFCCs) in spite of its successful nature throughout dialog recognition,

1. We cannot interpret MFCC coefficients physically.
2. Dynamic warping based speech systems use filtering which is connected with cepstral coefficients. In continuous domain it doesn’t have this effect in recognition. While representing the signal using power spectrum or phase spectrum there exists boundaries in representation

5.4. INTRODUCTION TO WAVELETS

There is no theoretical theory to support the MFCC to well represent a syllable without loss of information. In case of LPC and MFCC occurrence of noise is more in real time recognition system to avoid this we need to cascade enhancement to the system. but in case of wavelet due to filters present in it noise is removed without enhancement better recognition accuracy is possible as compare to other two techniques.

A wavelet is usually derived to be a wave-like oscillation together with amplitude that begins from zero, increases then
decreases to zero. It might be visualized very limited variations similar to one may possibly see recorded by a seismograph as well as heart monitor. Wavelet is usually added a reverse, shift, multiply and also sum method called convolution, with elements of an unknown signal to help extract information from the unknown signal. Wavelets are intentionally crafted to have exact characteristics that make them useful for signal computation.

This principle notion behind wavelets is always to analyze in line with scale. The wavelet evaluation procedure is always to adopt a wavelet prototype functionality called a good analyzing wavelet or mother wavelet. Any kind of conversation signal will be symbolized by converted and scaled versions from the mother wavelet. Wavelet analysis contains the ability of revealing facts of data other conversation signal evaluation technique such as extracted features are passed into a classifier for the recognition of isolated text. The basic principle idea driving wavelets should be to analyze in line with scale. The integral wavelet transform is defined as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$  \hspace{1cm} (5.45)

Where $a$ is positive scale and $b$ is any real number and represents the shift.

For decomposition of speech signal, we can use different techniques like Fourier analysis, STFT (Short Time Fourier Transforms), wavelet transform techniques.
5.4.1 WAVELET ANALYSIS

A wavelet is considered as waveform which has fixed time duration with mean zero. Fourier analysis is just the comparison of wavelets and sinusoidal waves. The difference between sinusoidal and wavelets are given below.

i. Sinusoidal signals are of infinite duration signals whereas wavelets are not

ii. Sinusoidal signals have smooth variation in amplitude, symmetric and can be estimated its amplitude value for a given instant of time if the frequency is known. On the other hand wavelets are asymmetric and irregular in nature.

Breaking up of signal in to sine waves of various frequencies is Fourier analysis similarly, wavelet analysis contain a signal which break up into shifted and scaled method of the original wavelet.

The process of Fourier analysis is denoted by the Fourier transform as:

\[ F(w) = \int_{-\infty}^{\infty} f(t) e^{-jwt} dt \]  \hspace{1cm} (5.46)

That is the addition over-all time on the signal f(t) multiplied by way of a complex exponential. Fourier coefficients are classified as the results on the transforms after they are multiplied by way of a
sinusoid involving frequency produce the constituent sinusoidal portion of the original signal.

The continuous wavelet transform (CWT) may be referred because the sum total time with the signal multiplied by scaled, shifted strategies of the wavelet function:

\[ C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} f(t) \phi(\text{scale}, \text{position}, t) \, dt \quad (5.47) \]

The actual output with the CWT will be many wavelet coefficients C, which are a functions connected with scale and position. Multiplying each coefficient from the concerned scaled and shifted wavelet yields the constituent wavelets with the original signal.

**5.4.2 WAVELET TRANSFORM**

A signal can be represented in two domains. (i) Time domain (ii) Frequency domain in time domain we can know about time information of the signal whereas in frequency domain we can know the spectral or frequency information of the signal. Transform of a signal means representing a signal in other form so that the hidden information which is not available in normal form can be extracted. The transform of a signal does not change the information content of the signal. For example a signal mostly represented in time domain and cannot be found in the frequency information in this domain so to find out the frequency information we have to transform the signal in frequency domain by taking its Fourier transform. A Fourier transform
of a signal gives the frequency information or spectrum nothing else. A Fourier transform of a signal gives the frequency information or spectrum nothing else. The time and frequency information can be found out simultaneously.

Wavelet Transform\cite{24} provides this information. It represents a signal in time as well as frequency domain by other words we can know the frequency information corresponding with time. Wavelet analysis of a signal is done by multiplying a wavelet function to the signal to be analyzed and then the transform is calculated for each block generated.

Transform of a signal can be found by using wavelet basis function known as mother wavelet and all other wavelet function used in transformation are derived by this basis function (mother wavelet) by translation and scaling. A continuous wavelet transform of signal can be represented by the following Equation.

$$X_{\omega \tau} (\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t).\varphi^*(\frac{t - \tau}{s}) dt$$  \hspace{1cm} (5.48)

Where \(x(t)\) is the signal to be analyzed and \(\psi(t)\) is mother wavelet or basis function. \(\tau\) and \(s\) are translation an scale parameter respectively. Scaling parameter \(s\) is related to the frequency. The parameter which gives the window location as it is shifted through the signal is called translation parameter \(\tau\). The time information is represented by this translation parameter.
Computation of CWT

1. Take a wavelet along with compare it with block of the starting with the original signal.

2. Calculate C, which shows how closely equate the actual wavelet with this section of the signal. The larger C is, the more similar it truly is. If the actual signal energy as well as the wavelet energy both are corresponding to unity, C is usually represented to be a correlation coefficient. Therefore the result is determined by the form of the wavelet.

![Correlated value of Wavelet](image)

3. Move the particular wavelet on the right aspect and repeat it again the steps 1 & 2 until covered the whole signal.
4. Stretch the wavelet and repeat it again from steps 1 to 3.

5. Repeat the above steps 1 to 4.

5.4.3 WAVELET FAMILY

Wavelet family includes the following type of wavelet. Their use is depends on the type of application. Some of the major used wavelet functions are shown in figure 5.14 Haar wavelet is the oldest one whereas Daubechies wavelets are the most popular one and used in many applications including signal processing, image processing etc. The Haar, Daubechies, Symlets and Coiflets are supported orthogonal wavelets.
5.4.4 TYPES OF WAVELETS

Different types of wavelets are Haar wavelets, Daubechies wavelets, Biorthogonal wavelets, Coiflet wavelets, Symlet wavelets, Morlet wavelets, Mexican Hat wavelets and Meyer wavelets.

Wavelets mainly used in speech recognition are discussed here.
5.4.5 Haar Wavelet

The first wavelet that is used is Haar wavelet. It is the simple wavelet and is similar to the step function and is discontinuous. Its first and simplest. It shows the same wavelet as Daubechies db1.

For $t \in [0, 1]$ the Haar wavelet is defined as

$$h_i(t) = \begin{cases} 1, & \text{for } t \in \left[\frac{k}{m} + 0, \frac{5}{m}\right] \\ -1, & \text{for } t \in \left[\frac{k}{m} + 0, \frac{5}{m}\right] \\ \end{cases}$$

(5.49)

Integer $m = 2^j$ ($j = 0, 1, 2, \ldots J$) indicates the level of the wavelet; $k = 0, 1, 2, \ldots m-1$ is the translation parameter. Maximal level of resolution is $J$.

5.4.6 DAUBECHIES-N WAVELET FAMILY

The Daubechies wavelets are a family regarding rectangular wavelets, which are characterized by way of maximal vanish moments and are also defined as a discrete wavelet change. There is usually a scaling functionality with every single wavelet type of this class, which produces an orthogonal multi resolution analysis. The Daubechies wavelet is just about the popular wavelets and has been useful for speech reputation.

In general the Daubechies wavelets are selected to offer the maximum number $A$ involving decaying scenarios, this will not indicate the top smoothness with regard to given width $N=2A$, and
involving the $2A-1$ probable solutions one is chosen whose scaling filter features has external phase. The wavelet transform can also be easy to place into practice using the fast wavelet transform. Daubechies wavelets are extensively used in solving an extensive range of problems, e.g self-similarity characteristics of any signal as well as fractal complications, signal discontinuities, etc.

The Daubechies wavelets properties:

- The support duration is $2N-1$ for a wavelet function $\Psi$ with the scaling function $\Phi$.

- $N$ is the variety of vanishing moments of the wavelet $\psi$.

- Symmetric property exists for most $\text{db}N$.

- Regularity increases while using the order. When $N$ becomes large $\Psi$ along with $\Phi$ go to $C^{\mu N}$ by taking the value of approximately 0.2 for $\mu$.

- Daubechies-8 wavelet is employed for decomposition connected with speech signal because it needs minimal support size to the given amount of vanishing points.

The family of Daubechies wavelets is represented as $\text{db}N$ with surname of the wavelet as db and $N$ the order of the wavelet. The particular $\text{db}1$ wavelet, as mentioned above, is equivalent to Haar. Here are the next nine family members.
5.4.7 DISCRETE WAVELET TRANSFORM

The particular scales in addition to positions were being chosen through the discrete wavelet transform (DWT) depending on power of two, so called dyadic weighing scales in addition to positions. Mother wavelet will be rescaled or even dilated through powers involving two in addition to converted through integers. The main advantage of using discrete wavelet change over continual wavelet transform is usually to enable rapid computation through computer in addition to hardware resources. In discrete wavelet change, the signal components tend to be processed with different resolution dependant on their frequencies. The signal is transferred through a number of digital filters to discover different rate of recurrence components. Multilevel electronic filter banks are used to conduct the filtering.

A function \( f(t) \in L^2(\mathbb{R}) \) (describes space regarding square integrals functions) can be shown as

\[
f(t) = \sum_{j=L}^{j=-L} \left\{ \sum_{k=-\alpha}^{k=\alpha} d(j,k) \psi(2^{-j}(t-k)) + \sum_{k=-\alpha}^{k=\alpha} a(L,K) \phi 2^{-j}t - k \right\} \tag{5.50}
\]

The function \( \psi(t) \) is referred as the mother wavelet, while \( \phi(t) \) is referred as the grading function.

The set of functions

\[
\left\{ \sqrt{2^{-j}} \phi(2^{-j}t-k), \sqrt{2^{-j}} \psi(2^{-j}t-k) \right\}, \quad [ j < L, j, k, L \in \mathbb{Z}] \tag{5.51}
\]

Wherever \( \mathbb{Z} \) would be the set connected with integers is an
orthogonal basis for $L^2(\mathbb{R})$.

The location where the letters $(L, k)$ are usually referred because estimation coefficients in scale $L$, while $d(j, k)$ are usually referred because the detail coefficients at scale $j$. The approximation along with detail coefficients are usually expressed as

$$\alpha(L, K) = \left( \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} f(t) \phi(2^{-l}t - k) dt \right)$$  \hspace{1cm} (5.52)

$$\alpha(j, K) = \left( \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} f(t) \phi(2^{-j}t - k) dt \right)$$  \hspace{1cm} (5.53)

The DWT analysis might be done having a fast, pyramidal algorithm concerned in order to multi-rate filter-banks. A multi-rate filter-bank DWT is certainly a Q filter-bank using octave spacing one of several centers of the filters. Each sub-band possesses half the samples of the surrounding higher frequency sub-band. In the pyramidal algorithm the sign is looked at at unique frequency groups with unique resolution simply by separating the actual signal into a coarse approximation along with detail information. The coarse estimation is usually additional separated while using the same wavelet decomposition. This may be accomplished simply by alternate high-pass along and low-pass filtering of that time domain signal and is defined as

$$y_{high}[n] = \sum_{k=-a}^{n} x[k] g[2n-k]$$  \hspace{1cm} (5.44)
\[ y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] h[2n-k] \] (5.45)

Fig. 5.13: Filter functions

Signal \( x[n] \) is passed through low pass and high pass filters and it is down sampled 2.

\[ y_{\text{low}}[n] = (x * g)^2 \] (5.56)
\[ y_{\text{high}}[n] = (x * h) \] (5.57)

By passing the previously estimated coefficients through HPF and LPF each level is estimated in DWT. Production of large band cross filter in addition to low band pass filter for a discrete wavelet decomposition of a signal could be represented mathematically by equation 5.56 and 5.57.

\[ Y^{\text{high}}[k] = \sum_{n} X[n]g[2k-1] \] (5.58)
\[ Y^{\text{low}}[k] = \sum_{n} X[n]h[2k-1] \] (5.59)

Exactly where \( Y \) high and \( Y \) low would be the outputs from the high pass and low pass filter systems respectively. Wavelet evaluation
performed so as to extract these features in the speech signal. A particular case from the wavelet transform provides with a succinct description of any signal time and frequency may be the discrete wavelet transform. Passing the signal through High pass and Low pass filters is the basic step in wavelet analysis. The most important thing in this is the good way of selecting the appropriate wavelet. The levels that are used for decomposition of wavelet is also important.

5.4.8 ADVANTAGES OF WAVELET ANALYSIS OVER STFT

Windowing technique with variable type is permitted with wavelet analysis. This is another logical stage of wavelets. Accurate low frequency information is obtained from these wavelets with long time periods. High frequency information is obtained with reduced regions.

Fig. 5.14: Comparison of Wavelet analysis over STFT
The time based, frequency based as well as STFT views of the signal are given regarding that involving Wavelet evaluation. One major advantage of wavelets is to be able to perform local analysis.

5.4.9 MULTILEVEL DECOMPOSITION OF SIGNAL

A signal can be decomposed using Wavelet Analysis as Shown below:

![Decomposition of DWT Co-efficient](image)

**Fig. 5.15: Decomposition of DWT Co-efficient**

![Decomposition using DWT](image)

**Fig. 5.16: Decomposition using DWT**
The successive low-pass filtering and high-pass filtering of the distinct time-domain signal calculated the DWT as shown in figure 5.16. This is known as the mallet algorithm or mallet tree decomposition.

**Fig. 5.17: Three level wavelet decomposition tree**

Figure 5.16 represents three level decomposition trees, exactly where every level comprises low and also high pass filter and also straight down sampler. HO and also Go represent high and also low pass filter systems respectively. The actual output connected with 1st level low pass filtration system has signal component having 50 percent the particular Sampling frequency. The specific consecutive higher-level blocks will once again decompose signal components in two-frequency area.

The particular high frequency filter output is referred to as detailed components and also the low frequency filter output is referred to as approximation coefficients. Every time output on low cross filter has signal components having 50 % the frequency
spectrum of input transmission at that level. Down sampling by 50% will throw out half the particular components all of which will keep each and every alternate parts. The quantity of signal components after just about any n-level decomposition is going to be \((\text{original number of signal components}) / (2^n)\).

Based on Nyquist’s solution, sampling frequency must be twice the actual signal frequency. If first signal offers frequency \(f_s\) then sampling rate must be \(2f_s\). Decrease sampling discards 50 percent of the components, so expected sampling frequency also lowers by 50 percent still just about all signal information is conserved inside decimated coefficients. Discrete wavelet alter provides beneficial time image resolution at increased frequencies as well as good frequency resolution at lower frequencies is utilized.

Wavelet transform comprises two main steps:

1) Scaling a signal by many number containing typical attributes which preserves the main harmonics of the signal that is certainly multiplying a signal by the Daubechies coefficients.

2) Splitting the main signal as well as compressed result into odd and even components so that we can preserve or even utilize all of them for subsequent level.
5.4.10 WAVELET DECOMPOSITION

The speech signal is decomposed up to sixth level using discrete wavelet transform after preprocessing of signal.

In figure 5.16 $c_A$ and $c_D$ are approximate and detail coefficients at different level respectively taking into account that the human speech frequency range between 250 – 4000 Hz.

Discrete wavelet transform analyze the signal at different frequency band with different resolution by decomposing the signal into approximation and detail coefficients. An approximation coefficient represents the lower frequency components whereas a detail coefficient represents higher frequency components of the signal. The approximation coefficients are computed by passing the signal in to low pass filter followed by down-sampling by 2 and detail coefficients are computed by passing the signal in to high pass filter followed by down-sampling by 2. $g[n]$ and $h[n]$ are the impulse response of the half band high pass and low pass filter through which the actual signal is passed through then the signal is sub sampled by 2. Therefore frequency of the signal will be half after each decomposition level.
5.4.11 CHOICE OF WAVELET

The selection of the mother-wavelet function which is used in designing speech recognition system, it is essential to choose a wavelet that has decrease assistance in both period and frequency in addition to a specific number of vanish moments for an optimum speech recognition. In selecting an optimal wavelet function different ways can be used. The aim is to minimize rebuild error variance and
increases signal to noise ratio. In general optimum wavelets can be
selected based on the energy conservation characteristics in the
estimation part of the wavelet coefficients.

Deciding on a decomposition level to the DWT usually depends
upon the sort of signal being examined or some other suitable method
such as entropy. Decomposition up to scale 6 is adequate for
processing of speech signals, with no further advantages gained in
processing.

5.4.12 IMPLEMENTATION

Extracting the important features that are sufficient enough for
the recognizer to recognize the words is the main task of this stage.
This describes how to extract information from a speech signal, which
means creating feature vectors from the speech signal. A wide range
of possibilities available for parametrically denoting a speech signal
and its content. The main steps intended for extracting details are
preprocessing, frame blocking & windowing and feature extraction.

Fig. 5.18: Main steps in Feature Extraction
5.4.13 PREPROCESSING

The basic step for generation of feature vectors is preprocessing technique where the speech signal \( x(n) \) is modified into a most suitable form to extract the feature vectors. Figure 5.19 shows different operations like noise cancelling, preemphasis and voice activation detection that will be performed under preprocessing.

\[
x(n) \xrightarrow{\text{Noise Canceling}} \hat{s}(n) \xrightarrow{\text{Preemphasis}} s_1(n) \xrightarrow{\text{Voice Activation Detection}} x_1(n)
\]

Fig. 5.19: Preprocessing

Usually the recorded speech is not accurate and clean i.e., there is a possibility for the noise to get add into the speech signal. Mathematically this can be represented as \( x(n) = s(n) + d(n) \) where \( x(n) \) is the noise corrupted signal with noise or disturbance part \( d(n) \) and \( s(n) \) the clear speech signal. This is the first issue that is to be considered. There may be variety of ways of perform disturbance reduction with a noisy presentation signal. It’s frequently utilized noise lowering algorithms in the field of speech recognition context is spectral subtraction along with adaptive noise cancellation. A decreased signal in order to noise proportion (SNR) lessens the performance of the recognizer in a real environment. Some changes
to make the speech recognizer more noise robust will be presented later. Note that the order of the operations might be reordered for some tasks. For example the noise reduction algorithm, spectral subtraction, is better placed last in the chain (it needs the voice activation detection).

5.4.14 PREEMPHASIS

There's a need regarding spectrally flatten this signal. The pre-emphasize, often represented with a first order high pass FIR filter can be used to emphasize the greater frequency elements. The second stage with feature extraction should be to enhance the energy in the high frequency contents. As it happens that in the event we think about the spectrum regarding voiced sections like vowels, there's more energy on the lower frequencies than the higher frequencies. This spectral tilt lower in vitality across frequencies (which is termed spectral tilt) is a result of the nature on the glottal pulse. Boosting this high frequency of recurrence energy can make information from these increased formants more accessible to the traditional acoustic model and improves phone detection exactness.

This pre-emphasize is employed to spectrally flatten the particular speech signal. This is usually done by way of high pass filter. The most commonly applied filter because of this step would be the FIR filter described below

\[ H (z) = 1 - 0.95 z^{-1} \] (5.60)
Fig. 5.20: Pre emphasis filter

The filter response just for this FIR filter is so shown in figure. The filter inside time domain \( h(n) = \{ 1, -0.95 \} \) are going to be and the filtering the period domain can give the pre emphasized signal \( s_1(n) \).

\[
s_1(n) = \sum_{k=0}^{M-1} h(k)s(n-k)
\] (5.61)

5.4.15 FRAME BLOCKING & WINDOWING

Speech signal is a form of unstable sign. But we are able to assume this kind of a stable signal in 10-20ms. Framing is utilized to reduce the particular long-time speech towards short-time speech signal to obtain relative stable frequency characteristics. Features acquire periodically produced. The time is actually a signal is known
for processing is actually a window as well as the data acquired within a window is called frame. Generally features are usually extracted once every 10ms, to create frame rate. The window duration can often be 20ms. Subsequently two consecutive structures have overlapping regions.

**5.4.16 FRAME BLOCKING**

For each utterance with the word, window length of 320 samples is used for finalizing in the future stages. A shape is formed from the windowed facts with common frame length ($T_f$) of approximately 200 samples. Since the particular frame length is smaller than window duration there’s an overlap involving data as well as the percentage overlap can be given as follows

$$\% \text{Overlap} = \left( \frac{T_w - T_f}{T_w} \right) \times 100$$  \hspace{1cm} (5.62)

Every frame can be $K$ samples long, along using adjacent frames being separated by $p$ samples.

---

**Fig. 5.21: Frame blocking of a sequence**
By applying the frame blocking to de noised signal \( x(k) \), we could possibly get \( M \) vectors connected with length \( K \), which correspond to \( x(k; m) \) where \( k=0, 1...K-1 \) and \( m=0, 1....M - 1 \).

### 5.4.17 WINDOWING

To minimize the signal distortion, windowing concept is needed which is because of the window to be able to taper this signal to be able to zero from the outset and end of frame i.e., to lessen signal discontinuity at either end from the block.

The rectangular window (i.e., electronic no window) might cause problems, if we do Fourier investigation; it abruptly cuts in the signal on its boundaries. Rectangular window has the narrow main lobe with small side lobe levels, which lessens the values from the signal to zero on the window boundaries, avoiding discontinuities.

\[
w(k) = \begin{cases} 
1, & 0 < k < K \\
0, & \text{otherwise} 
\end{cases} \quad (5.63)
\]

The most commonly used window function in speech processing is hamming window defined as follows

\[
w(k) = 0.54 - 0.46 \cos \left( \frac{2\pi k}{K} - 1 \right) \quad (5.64)
\]
Hamming window function is shown in Figure 5.22 below:

![Hamming Window](image)

**Fig. 5.22: Hamming Window**

Multiplication in the signal with a window function from the time domain matches convolving the actual signal from the frequency domain. Hamming window blurs in frequency but generates much less leakage. Window produces highest accuracy but large side-lobes (ripples).

### 5.4.18 FEATURE EXTRACTION

The particular pattern vector will be reduced to a lower dimension by way of a feature extractor, which consists a lot of the beneficial data in the original vector. Here use of all extract features of the input speech transmission using Daubechies-8 wavelets regarding level 4. The estimated wavelet coefficients offer a compressed representation that shows the energy allocation with the signal in time
and frequency. To slow up the dimensions with the derived element vectors, statistics over the set with the wavelet coefficients are used.

The following specifications are used in our system:

- In each sub band the mean of the accurate value of the coefficients. These characteristics provide data concerning the frequency allocation of the speech signal.
- With each sub band the common deviation of the coefficients. These features provide data about the amount of change in the consistency distribution.
- Energy of sub-band of signal. Provides data about energy of each sub-band.
- Lopsidedness of sub-band of the signals. These characteristics are the measure of symmetry or deficit of symmetry.

After frame blocking and windowing we find different frame vectors i. e. different signals are to be loaded to get the features at a time. Hence Multi signal analysis is completed on input framework vectors using wavelets using Mat lab.

a) Results obtained before applying to system

![Input Speech Signal](image)

Fig. 5.23: Input digit sunna
The input speech signal with duration of 5 seconds with sampling frequency of 8k Hz is shown in Figure 5.23

**Pre emphasis:**

Pre emphasis output for “sunna”:

![Speech Signal after Pre-Emphasis](image)

No. of samples

**Fig. 5.24: Pre-Emphasis Input digit sunna**

The output obtained after passing the input “sunna” signal to the pre emphasis (first order high pass) filter. The output has significant spectral flatness when compared with input.

**Voice Activation & Detection**

1) Voice Activation and Detection for “Sunna”:

![Voice Activation Detection](image)

No. of samples

**Fig. 5.25: Voice Activation and Detection for Sunna**
The above plot shows the voice activated region for the digit "sunna". The output is 1 for voiced region and 0.5 for unvoiced and silence region. Hence out of the total samples, only the voice activated samples are going to be filtered out.

**De-noising:**

De-noising for digit sunna:

![De-noising signal for Sunna](image)

**Fig. 5.26:** De-noising signal for Sunna

The final denoised signal obtained after Spectral subtraction. Here the noise components present in the signal are reduced.

### 5.5 Dynamic Time Warping

It is the method where the speech signal is divided into corresponding templates thereby, whenever the speech for detection arrives for recognition, it is converted into a similar template and the distance from this template to the existing templates is calculated. The template to which the distance measured is minimum that template is recognized as the most nearest speech signal. For doing this dynamic programming is used and hence it is called Dynamic Time Warping (DTW) word recognizer[26].
Since speech is the time varying signal different time durations will exists for the same word which is spoken in different instances. Same duration speech signal with same sound will not have the same amplitudes at different times because the rate of speech differs. A time alignment must be carried out to get a global distance between two speech patterns. The best understanding of DTW is done by understanding Features and Distances. Representation of speech signal in any possible way is the part of Feature whereas Distances is a metric to match two templates. It is of two types. (i) Local Distance, (ii) Global Distance. Local distance calculates the feature related difference between two signals and Global distance calculates the similar difference but this is between the total signal and another signal with different length.

There is a possibility for the feature vectors to have multiple vectors and hence these are to be calculated. By calculating Euclidean distance metric one can easily calculate the distance between two features. The Local distance is given by

\[ d(x, y) = \sqrt{\sum (x_i - y_i)^2} \]  

(5.65)

where \( x, y \) are the feature vectors for two different signals.
5.5.1 DTW ALGORITHM

Speech can be a time-dependent method, the sound in the same word could have different duration and also the sounds associated with same phrase with similar duration will vary in the actual spoken at various portions in the words becoming spoken at various rates. A time alignment has to be carried out to have a global distance between a couple of speech patterns which can be shown as a order associated with vectors a time adjustment has to be performed.

The time-time matrix is illustrated in Figure 5.27. This matrix is used to represent some time adjustment, as with all the current time adjusting examples the reference pattern goes up the side along with the input pattern goes down the bottom side. In this kind of illustration the actual input SsPEEhH is usually a noisy style of the format SPEECH. The aim is that may ‘h’ is usually a nearest to ‘H’ in comparison to anything else in the template. The input SsPEEhH may be equal all templates inside system’s repository. The excellent equality template could be the one that there could be the shortest range path aiming the input pattern to template. A basic global range score for just a path is the sum of local range that check out make the path.
We apply specific conditions for the direction connected with propagation in order to make the algorithms in addition to reduce extreme computation. The constraints are shown below.

- It is not possible for the matching paths to move back.
- For matching paths all frames shall be used.
- To get global distances the scores of local distances shall be added

Such an algorithm is called Dynamic Programming (DP). If we apply this algorithm to template based conversion identification it is called Dynamic Time Period Warping (DTW). Dynamic Programming is needed to find the shortest distance path throughout the matrix, while reducing the amount of computation. The DP criteria operates within
a time-synchronous approach each column with the time-time matrix
is recognized as in chronological sequence similar to processing this
input frame-by-frame, for the template associated with length N, the
maximum number of paths becoming treated anytime is N. If \( D(i, j) \)
would be the global distance approximately \((i, j)\) along with all the
local length at \((i, j)\) is given by \( d(i, j) \).

\[
D(i, j) = \min[D(i-1, j-1), D(i-1, j), D(i, j-1)] + d(i, j) \tag{5.66}
\]

Considering that \( D(1, 1) = d(1, 1) \) would be the initial condition, we
contain the basis to have an capable recursive algorithm for
calculating \( D(i, j) \). The matching score on the template using the input
is due to overall distance \( D(n, N) \), the input word is usually the
acknowledged as the word concerned towards the template using the
lowest equaling score.

Dynamic time warping is an algorithm, which is used pertaining
to finding similarity between a couple of sequences which can differ in
time or speed. For case in point resembling with walking patterns will
be found, through in a single video anyone was walking and with
another he / she were jogging fast and even there was a increase or
decrees with speed during the course of one observation, DTW also
placed on audio, video clip, and graphics indeed, any data that is
converted into a linear representation is usually examined with DTW.
A crucial application has become automatic speech recognition, to
handle different communicating speeds.
DTW is often a method that produces a computer to understand a maximum match involving two given sequences together with less rules. The order is warped non-linearly inside time dimension so as to determine a measure of their similarity that’s not being based upon concerned non-linear changes inside time measurement. Incoming signal is equated to the located prototype in the recognition process. The word which has the small distance measure from the template word is recognized first. The finest shortest length measure is reliant upon dynamic programming.

5.5.2 DP-MATCHING PRINCIPLE

General Time-Normalized Distance Definition

Speech is usually expressed by appropriate function extraction as a sequence associated with feature vectors.

\[ A = a_1, a_2, a_3, \ldots, a_i, \ldots, a_I \] \hspace{1cm} (5.67)

\[ B = b_1, b_2, b_3, \ldots, b_j, \ldots, b_J \] \hspace{1cm} (5.68)

Consider the problem connected with eliminating timing distinctions between both of this speech behavior. In order to clarify the character of time-axis fluctuation or timing distinctions, let’s consider ai-j plane, shown in Figure. Where behavior A along with B tend to be developed across the i-axis along with j-axis, respectively. Where these types of speech behavior are with the same category, the
timing distinctions between them might be depicted by the sequence connected with points \( c = (i, j) \):

\[
F = c(1), c(2), \ldots c(k), \ldots c(K) \quad (5.69)
\]

In which \( c(k) = (i(k), j(k)) \).

This specific sequence could be to characterize a function which approximately realizes the mapping from the time axis involving pattern A onto in which of pattern B. It can be called the warping function. When there isn't any timing distinction between these types of patterns, the warping function coincides while using diagonal range \( j = i \). It deviates further through the diagonal line because timing distinction grows.

![Fig. 5.28: Warping function & altering window](image-url)
Being a measure from the difference between two feature vectors $a_i$ and $b_i$.

$$d(c) = d(i, j) = |a_{i, j} - b_{i, j}|$$ \hspace{1cm} (5.70)

is needed between all of them. Then, this weighted summation connected with distances with warping functionality $F$ will become.

$$E(F) = b \sum_k d(c(k)) \cdot w(k)$$ \hspace{1cm} (5.71)

Where $w(k)$ is usually a nonnegative weighting coefficient, and that is intentionally introduced to permit the $E(F)$ calculate flexible characteristic which is a reasonable measure with the goodness associated with warping function $F$. It attains its bare minimum value whenever warping function $F$ is determined in an attempt to optimally change the timing distinction. This bare minimum residual distance value could be to certainly be a distance in between patterns $A$ and $B$, remaining however after eradicating the timing dissimilarities between these, and will be naturally anticipated to be dependable against time-axis fluctuation. Depending on these factors, the time-normalized distance between a couple of speech patterns $A$ as well as $B$ means follows.

$$D(A, B) = \min_F \left[ \sum_k d(c(k)) \cdot \left[ \frac{w(k)}{\sum_{k=1}^K w(k)} \right] \right]$$ \hspace{1cm} (5.72)

Where denominator $\sum(w(k))$ must be used to compensate for the
effect regarding K (number regarding points about the warping function F). Above equation is a maximum of a basic definition regarding time-normalized distance.

5.5.3 DISCUSSIONS ON WEIGHTING COEFFICIENT

Since the criterion perform in equation 5.69 is usually a rational expression, its maximization can be an unwieldy trouble. If the actual denominator throughout Equation 5.70.

\[ N = \sum_{k} d_w(k) \]  \hspace{1cm} (5.73)

Named normalization coefficient is actually independent regarding warping perform Fit might be put beyond bracket, while simplifying the equation as follows.

\[ D(A, B) = \min_k \left[ \sum_{l=1}^{K} d(c(k) \cdot w(k)) \right] \]  \hspace{1cm} (5.74)

\[ W(k) = [i(k) - i(k-1)] + [j(k) - j(k-1)] \]  \hspace{1cm} (5.75)

Then

\[ N = I + J, \]

Where by ‘i’ as well as ‘j’ are generally lengths regarding speech patterns A as well as B, respectively.
Whether it is assumed that time axes i and j are classified as the two continuous. Subsequently, in the symmetric kind, the summation in equation 5.69 signifies an integration on the temporarily defined axis I = i + j. Because of this change, time-normalized distance is symmetric, or perhaps \( D(A, B) = D(B, A) \), inside the symmetric kind. Another a lot more important outcome, caused through the difference inside the integration axis, is usually that, as in Figure 5.29 a weighting coefficient \( w(k) \) decreases to zero inside the asymmetric kind, when the purpose in warping function steps in the direction to j-axis, or \( c(k) = c(k-1) + (0, 1) \).

This means that some feature vectors bj are generally possibly excluded from the integration inside the asymmetric type. On the actual contrary, regarding symmetric type, minimum \( w(k) \) importance is adequate to 1, and no exclusion takes place. Since discussions here derive from the assumption that many part within a speech pattern needs to be treated both equally, an different of almost any feature vectors coming from integration needs to be avoided so long as possible. It can be expected, as a result, that the actual symmetric form gives better acceptance accuracy than the asymmetric type.

However, it need to be noted how the slope limit reduces the situation where the actual in warping function steps from the j-axis route. The particular difference inside performance relating to the symmetric one and asymmetric one will slowly vanish for the reason that slope limit is intensified.
5.6 HIDDEN MARKOV MODEL

To implement the actual speech identification system, the hidden markov Style (HMM) has been used this technique is accustomed to train the model which within our case ought to represent the utterance of a word. This model is employed later on in this testing of the utterance, and also calculating the actual probability of this the model has established the sequence of vectors.

Hidden Markov Model (HMM) is a state machine\(^{27}\). The particular states on the model tend to be represented since nodes plus the transition tend to be represented as edges. The difference in the event of HMM is the symbol doesn’t uniquely identify a state. The new state is dependent upon the symbol as well as the transition probabilities through the current state to some other state. Figure 5.30 indicates a diagrammatic representation of an HMM. Nodes denoted as circles are states. O1 to O5 are observations. Declaration O1 will take us to be able to states S1. It is usually observed how the
states also provide self changes. If we have been at talk about S1 together with observation O2 is discovered, we can certainly either choose to go state S2 or even in state S1. The decision is made with regards to the probability regarding observation at both the states and the transition probability.

Fig. 5.30: Diagrammatic representation of HMM

The particular difference among an observable Markov Model as well as a Hidden Markov Style is that in the Observable the actual output state seemingly determined at every time t. Within the hidden Markov Model the state at every time t should be inferred via observations. An observation is often a probabilistic function of a state. Further information about the actual difference and information on the observable Markov Style and Hidden Markov Style.

The hidden Markov Model is manifested by $\lambda = (\pi, A, B)$. 
\( \pi \) = preliminary state distribution vector.

\( A \) = State transition probability matrix.

\( B \) = continuous observation probability density function matrix.

There are three basic problems in the Hidden Markov Model design is given below.

### 5.6.1 RECOGNITION

The observation sequence \( O = (O_1, O_2, \ldots, O_T) \) along with the model \( \lambda = (\pi, A, B) \), how would be the probability from the observation routine given the actual model calculated. That is, how is \( P(O|\lambda) \) basically computed correctly.

### 5.6.2 OPTIMAL STATE SEQUENCE

Offered the observation sequence \( O = (O_1, O_2, \ldots, O_T) \) along with the model \( \lambda = (\pi, A, B) \), how is a related state sequence \( q = (q_1, q_2, \ldots, q_T) \), chosen for being optimal in a few sense.

### 5.6.3 ADJUSTMENT

How would be the probability steps, \( \lambda = (\pi, A, B) \), adjusted to improve \( P(O|\lambda) \)
5.6.4 HMM – THE TRAINING OF A MODEL OF A WORD–THE RE-ESTIMATION PROBLEM

Given N variety of observation sequences of an word ON= {O1 O2 O3 ....OT}. How may be the training of that model completed to finest representation of the word. This is completed by modifying the parameters to the model.\(\lambda=(\pi, A, B)\). The adjustment is an estimation of the parameters with the model \(\lambda=(\pi, A, B)\) which improves\((O|\lambda)\). The option for this can be the solutions with the first and third HMM problems.

The sequence to generate a HMM connected with speech utterances may be the following:
Fig. 5.31: HMM of speech utterances
5.6.5 PRINCIPAL CASES OF HMM

There are three primary cases to become dealt having to formulate a prosperous HMM. They're:

**Case 1: Evaluation**

Given:

- A design $\lambda = (A, B, \pi)$ wanting to be used.
- Testing observation sequence $I = O_1, O_2, O_3, \ldots, O_{T-1}, O_T$.
- Action: work out $P(O/\lambda)$: The probability in the observation order given in the model.

**Case 2: Decoding**

Given:

- A type $\lambda = (A, B, \pi)$ able to be applied.
- Testing or even training observation sequence $O = O_1, O_2, O_3, \ldots, O_{T-1}, O_T$.
- Action: Track the perfect state sequence $Q = q_1, q_2, q_3, \ldots, q_{T-1}, q_T$ that many likely produce the given observations, using the given type.
Case 3: Instruction

Given:

- A model $\lambda = (A, T, \pi)$ willing to be utilized.
- Training declaration sequence $O_k = O_{1k}, O_{2k}, O_{3k} \ldots \ldots \ O_{T-1k}, O_{Tk}$

Where $K$ is the number of examples regarding training model.

- Action: Tune your model parameters to maximize $P(O/\lambda)$. Case 3: Training

5.6.6 SIGNAL – THE UTTERANCE

This signal employed for training requirements are common utterances on the specific word, the word to become recognized.

$$\chi_{uit}(n) = \quad . \quad . \quad . \quad . \quad . \quad . \quad n = 1,2, \ldots, 24253$$

Fig. 5.32: Utterances of word

5.6.7 MFCC – MEL FREQUENCY CEPSTRUM COEFFICIENTS

The particular MFCC matrix can be calculated in accordance Speech Indication to Mel frequency Cepstrum Coefficients. This really also utilized when tests an utterance next to model. The testing associated with an observation, $\mu$, mean once the MFCC can be achieved, there exists a need to help normalize the many given
instruction utterance. The matrix might be divided into quite a few coefficients times’ variety of states. Then these include regarding calculating the mean in addition to variance all the matrices regarding variance calculations. The mean is computed using equation 5.76.

\[
\bar{x}_c = \frac{1}{N} \sum_{n=0}^{N-1} x_c(n), \ c = \text{column} \quad (5.76)
\]

\[
x_{\mu}(m, n) = \begin{array}{c}
\text{mean}(x^{(1,1,12)}) & \text{mean}(x^{(1,13,24)}) & \text{mean}(x^{(1,25,36)}) & \text{mean}(x^{(1,37,49)}) & \text{mean}(x^{(1,49,62)}) \\
\text{mean}(x^{(2,1,12)}) & \text{mean}(x^{(2,13,24)}) & \text{mean}(x^{(2,25,36)}) & \text{mean}(x^{(2,37,49)}) & \text{mean}(x^{(2,49,62)}) \\
\text{mean}(x^{(3,1,12)}) & \text{mean}(x^{(3,13,24)}) & \text{mean}(x^{(3,25,36)}) & \text{mean}(x^{(3,37,49)}) & \text{mean}(x^{(3,49,62)}) \\
\text{mean}(x^{(4,1,12)}) & \text{mean}(x^{(4,13,24)}) & \text{mean}(x^{(4,25,36)}) & \text{mean}(x^{(4,37,49)}) & \text{mean}(x^{(4,49,62)}) \\
\text{mean}(x^{(5,1,12)}) & \text{mean}(x^{(5,13,24)}) & \text{mean}(x^{(5,25,36)}) & \text{mean}(x^{(5,37,49)}) & \text{mean}(x^{(5,49,62)}) \\
\text{mean}(x^{(6,1,12)}) & \text{mean}(x^{(6,13,24)}) & \text{mean}(x^{(6,25,36)}) & \text{mean}(x^{(6,37,49)}) & \text{mean}(x^{(6,49,62)}) \\
\text{mean}(x^{(7,1,12)}) & \text{mean}(x^{(7,13,24)}) & \text{mean}(x^{(7,25,36)}) & \text{mean}(x^{(7,37,49)}) & \text{mean}(x^{(7,49,62)}) \\
\text{mean}(x^{(8,1,12)}) & \text{mean}(x^{(8,13,24)}) & \text{mean}(x^{(8,25,36)}) & \text{mean}(x^{(8,37,49)}) & \text{mean}(x^{(8,49,62)}) \\
\text{mean}(x^{(9,1,12)}) & \text{mean}(x^{(9,13,24)}) & \text{mean}(x^{(9,25,36)}) & \text{mean}(x^{(9,37,49)}) & \text{mean}(x^{(9,49,62)}) \\
\text{mean}(x^{(10,1,12)}) & \text{mean}(x^{(10,13,24)}) & \text{mean}(x^{(10,25,36)}) & \text{mean}(x^{(10,37,49)}) & \text{mean}(x^{(10,49,62)}) \\
\text{mean}(x^{(11,1,12)}) & \text{mean}(x^{(11,13,24)}) & \text{mean}(x^{(11,25,36)}) & \text{mean}(x^{(11,37,49)}) & \text{mean}(x^{(11,49,62)}) \\
\text{mean}(x^{(12,1,12)}) & \text{mean}(x^{(12,13,24)}) & \text{mean}(x^{(12,25,36)}) & \text{mean}(x^{(12,37,49)}) & \text{mean}(x^{(12,49,62)}) \\
\text{mean}(x^{(13,1,12)}) & \text{mean}(x^{(13,13,24)}) & \text{mean}(x^{(13,25,36)}) & \text{mean}(x^{(13,37,49)}) & \text{mean}(x^{(13,49,62)}) \\
\end{array}
\]

\[
x_{\mu}(m, n) = \begin{array}{c}
\text{mean}(x^{(1,1,12)}) & \text{mean}(x^{(1,13,24)}) & \text{mean}(x^{(1,25,36)}) & \text{mean}(x^{(1,37,49)}) & \text{mean}(x^{(1,49,62)}) \\
\text{mean}(x^{(2,1,12)}) & \text{mean}(x^{(2,13,24)}) & \text{mean}(x^{(2,25,36)}) & \text{mean}(x^{(2,37,49)}) & \text{mean}(x^{(2,49,62)}) \\
\text{mean}(x^{(3,1,12)}) & \text{mean}(x^{(3,13,24)}) & \text{mean}(x^{(3,25,36)}) & \text{mean}(x^{(3,37,49)}) & \text{mean}(x^{(3,49,62)}) \\
\text{mean}(x^{(4,1,12)}) & \text{mean}(x^{(4,13,24)}) & \text{mean}(x^{(4,25,36)}) & \text{mean}(x^{(4,37,49)}) & \text{mean}(x^{(4,49,62)}) \\
\text{mean}(x^{(5,1,12)}) & \text{mean}(x^{(5,13,24)}) & \text{mean}(x^{(5,25,36)}) & \text{mean}(x^{(5,37,49)}) & \text{mean}(x^{(5,49,62)}) \\
\text{mean}(x^{(6,1,12)}) & \text{mean}(x^{(6,13,24)}) & \text{mean}(x^{(6,25,36)}) & \text{mean}(x^{(6,37,49)}) & \text{mean}(x^{(6,49,62)}) \\
\text{mean}(x^{(7,1,12)}) & \text{mean}(x^{(7,13,24)}) & \text{mean}(x^{(7,25,36)}) & \text{mean}(x^{(7,37,49)}) & \text{mean}(x^{(7,49,62)}) \\
\text{mean}(x^{(8,1,12)}) & \text{mean}(x^{(8,13,24)}) & \text{mean}(x^{(8,25,36)}) & \text{mean}(x^{(8,37,49)}) & \text{mean}(x^{(8,49,62)}) \\
\text{mean}(x^{(9,1,12)}) & \text{mean}(x^{(9,13,24)}) & \text{mean}(x^{(9,25,36)}) & \text{mean}(x^{(9,37,49)}) & \text{mean}(x^{(9,49,62)}) \\
\text{mean}(x^{(10,1,12)}) & \text{mean}(x^{(10,13,24)}) & \text{mean}(x^{(10,25,36)}) & \text{mean}(x^{(10,37,49)}) & \text{mean}(x^{(10,49,62)}) \\
\text{mean}(x^{(11,1,12)}) & \text{mean}(x^{(11,13,24)}) & \text{mean}(x^{(11,25,36)}) & \text{mean}(x^{(11,37,49)}) & \text{mean}(x^{(11,49,62)}) \\
\text{mean}(x^{(12,1,12)}) & \text{mean}(x^{(12,13,24)}) & \text{mean}(x^{(12,25,36)}) & \text{mean}(x^{(12,37,49)}) & \text{mean}(x^{(12,49,62)}) \\
\text{mean}(x^{(13,1,12)}) & \text{mean}(x^{(13,13,24)}) & \text{mean}(x^{(13,25,36)}) & \text{mean}(x^{(13,37,49)}) & \text{mean}(x^{(13,49,62)}) \\
\end{array}
, \ m = 13, \ n = 5
\]

**Fig. 5.33: Multiple utterances of word**

Note that if multiple utterances utilized for training there exists a need of computing the mean connected with \( x_{\mu} (m, n) \) for that number of utterances.

\[
\Sigma, \text{ variance}
\]

\[
\bar{x}_c^2 = \frac{1}{N} \sum_{n=0}^{N-1} x_c^2(n), \ c = \text{column} \quad (5.77)
\]
\[ \sigma_c^2 = \bar{x}_c^2 - \left[ (x_c)_c \right]^2, \ c = \text{column} \] (5.78)

The variance is calculated using equation 5.77 and 5.78.

An even more explicit example of calculating a specific index e.g. the \( x_{\Sigma(1, 1)} \) is done using the following equation

\[
x_{\Sigma(1, 1)} = x_{\Sigma(1:12)} \sum x - \mu(1,1)^2
\] (5.79)

5.6.8 INITIALIZATION

The state transition probability matrix, with the left-to-right model. Their state transition probability matrix, \( A_{\text{new}} \) is initialized. While using the equal probability for each state.

\[
A = \begin{bmatrix}
0.5 & 0.5 & 0 & 0 & 0 \\
0 & 0.5 & 0.5 & 0 & 0 \\
0 & 0 & 0.5 & 0.5 & 0 \\
0 & 0 & 0 & 0.5 & 0.5 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

Over the experimentation with how many iterations inside the reestimation of \( A \) the last estimated values of \( A \), were shown to deviate quite a lot from a beginning estimation. A last initialization values of the \( A \), where initialized with all the following values instead, which is very likely to the re estimated values.

\[
A = \begin{bmatrix}
0.85 & 0.5 & 0 & 0 & 0 \\
0 & 0.85 & 0.5 & 0 & 0 \\
0 & 0 & 0.85 & 0.5 & 0 \\
0 & 0 & 0 & 0.85 & 0.5 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]
The actual change associated with initialization values isn’t an important event therefore the re estimation adapt the values for the correct ones using the estimation procedure. Initialize the initial state submitting vector, with the entire left-to-right model. The primary state submitting vector is actually initialized while using the probability to be state one in the beginning, which is actually assumed in speech acceptance theory. It is also assumed that is comparable to five states in cases like this.

\[ \pi_i = [i \ 0 \ 0 \ 0 \ 0], \ 1 \leq i \leq \text{variety of states}, \text{in such case } i = 5 \]

5.6.9 MULTIPLE UTTERANCE ITERATION

This Continuous observation Probability Denseness Function Matrix. HMM - Hidden Markov Design, the complication on the direct observation on the state the speech process is just not possible there’s need for a few statistic calculations. This is performed by launching the continual observation likelihood density function purpose matrix, B. The idea would be to that there is a probability of creating a particular observation from the state, the probability that this model features produced the actual observed Mel Rate of recurrence Cepstrum Coefficients. There exists a individually distinct observation probability option to use. This really is less. Difficult in calculations but it really uses some sort of vector quantization that generates some sort of quantization errors.
The advantage with steady observation probability density capabilities is how the probabilities usually are calculated direct in the MFCC with no quantization. The regular distribution accustomed to describe this observation density would be the Gaussian one particular. To stand for the steady observation possibility density functionality matrix, $B$ the mean, $\mu$ and also variance, $\Sigma$ are used.

Because of that the actual MFCC are usually not frequency distributed the weight coefficient is critical to use in the event the mixture of the Probability distribution Function is actually applied. This kind of weight coefficient, more how many these weights is used to design the frequency functions which cause an mixture of the probability Distribution Purpose.

$$b_j(o_t) = \sum_{k=1}^{M} c_{jk} b_{jk}(o_t), \; j = 1,2,\ldots, N$$

And $M$ is the amount of mixture weights $C_{jk}$. These are restricted due to

$$\sum_{k=1}^{M} c_{jk} = 1, \; j = 1,2,\ldots, N$$

$$c_{jk} \geq 0, \; j = 1,2,\ldots, N; \; k = 1, \; 2,\ldots, M$$

By using diagonal covariance matrices, a result of the less computation plus a faster implementation, then the following formula is employed.
\[ b_{jk}(O_t) = \frac{1}{(2\pi)^{D/2} \left( \prod_{l=1}^{D} \sigma_{jkl} \right)^{1/2}} e^{-\sum_{i=1}^{D} \frac{(O_{ti} - \mu_{jkl})^2}{2\sigma_{jkl}^2}} \] (5.80)

A single \( x_{mfcc} \) characteristic vector is within the estimation versus each and every \( \mu \)- and also \( \Sigma \) vector. i.e. Each characteristic vector can be calculated for all \( x_\mu \)- and \( x_\Sigma \) columns one by one.

The actual resulting state-dependent observation symbol possibilities matrix. The columns required observation probabilities for every state.

5.6.10 FORWARD ALGORITHM

If we consider the sequence \( O = \{O_1 O_2 O_3 \ldots \ldots O_T\} \) and a model which is given by \( \lambda = (\pi, A, B) \) and tries to obtain the probability of the sequence, we need to find the solution to difficulty one, probability assessment. The solution is about finding which from the models (assuming that they exist) that almost certainly has produced the observation sequence.

By natural means to do this is to evaluate each and every possible sequence connected with states of length \( T \) then add these collectively.

\[ P (O / \lambda) = \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} \prod_{i=2}^{T} a_{q_{i-1} q_i} b_{q_i} (O_i, ) \] (5.81)
Here the symbol $O_i$ is generated with $b_{q_i}(O_i)$ as the probability of occurrence from the state $q_1$ which has the probability of $\pi_{q_1}$ at the starting time $t=1$. The clock modifications from $t$ to $t+1$ along with a transition from $q_1$ in order to $q_2$ can occur having probability $a_{q_1q_2}$, and also the symbol $O_2$ will end up being generated with possibility $b_{q_2}(O_2)$. The process continues in this fashion until the last transition is made (at moment $T$), i.e., a new transition via $q_{T-1}$ to $q_T$ may occur with possibility $a_{q_{T-1}q_T}$, and the symbol will probably $O_T$ be generated with probability $b_{q_T}(O_T)$.

The amount of computations is extensive possesses an exponential growth being a function of sequence length $T$. This equation is $2T \times NT$ calculations. When using that equation with 5 states and 100 an observation provides about 1072 calculations. As this amount of computations is incredibly demanding it is vital to discover a method to reduce this total. This is done while using the Forward Algorithm.

The actual Forward Algorithm is founded for the forward varied $\alpha_i(i)$, defined by $\alpha_i(i) = P(o_1, o_2, \ldots, o_i, q_i = i \mid \lambda)$. The definition of $\alpha_i(i)$ is that $\alpha_i(i)$ the probability at period $t$ and throughout state $i$ granted the model, having generated the particular partial observation sequence on the first observation right up until observation number $t$, $o_1, o_2, \ldots, o_t$. The variable might be calculated inductively in accordance with Figure
6. $\alpha_{i+1}(i)$ might be calculated by summing the particular forward variable for all $N$ states on time $t$ multiplied using their corresponding state change probability and by the emission probability $b_{q_i}(o_{i+1})$. The method of calculating the particular forward variable, which is often computed at any time $t, 1 \leq t \leq T$ is shown below figure 5.34.

**Fig. 5.34: Forward probability function representation**
1. **Initialization**
   
   **Set** $t = 1$; $\alpha_i(i) = \pi_i b_i(o_i)$, $1 \leq i \leq N$

   From the initialization stage the forward variable will get its start value, which means the joint probability of being in point $1$ along with observing this symbol $o_i$. In left-to-right models only $\alpha_i(l)$ could have a nonzero value.

2. **Induction**
   
   $\alpha_{i+1}(j) = b_j(o_{i+1}) \sum_{i=1}^{N} \alpha_i(i)a_{ij}$, $1 \leq j \leq N$ Using the lattice structure in Figure 6.6.

3. **Update time**
   
   Set $t = t + 1$; Return to step 2 if $t \leq T$;

   Normally, terminate the particular algorithm.

4. **Termination**
   
   $P(O | \lambda) = \sum_{i=1}^{N} \alpha_f(i)$

As mentioned above example in line with the number of computations in the any path which gave variety of 1072 calculations combined with 5 states in addition to 100 observations.

When use in the particular forward algorithm the amount of multiplications will always be $N(N-1)(T-1)$ and $N(N+1)(T-1) + N$ improvements. With 5 states and 100 observations it’s going to give 2975 multiplications in addition to 1980 additions, to look when placed against the direct method (any path) that gave 1072 calculations.
5.6.11 BACKWARD ALGORITHM

When the recursion explained to calculate the forwards variable is completed in the slow way and the backward transforming. This variable will be defined with the following significance

\[
\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T \mid q_t = i, \lambda)
\]  

(5.82)

The definition of \( \beta_t(i) \) is that \( \beta_t(i) \) is the particular probability on time \( t \) and state \( i \) given particular model, getting produced the partial observation order from \( t+1 \) observation until observation number \( T \). The variable can be found out inductively according to Figure 5.37. The scaling factor, \( \alpha \) scaled, \( \beta \) scaled.
Fig. 5.35: Backward probability function representation

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Initialization</td>
<td>Set $t = T - 1$; $\beta_t(i) = 1, \quad 1 \leq i \leq N$</td>
<td></td>
</tr>
<tr>
<td><strong>2.</strong> Induction</td>
<td>$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(i) a_j b_j (o_{t+1}), \quad 1 \leq i \leq N$</td>
<td></td>
</tr>
<tr>
<td><strong>3.</strong> Update time</td>
<td>Set $t = t - 1$; Return to step 2 if $t \geq 0$; Otherwise, terminate the algorithm</td>
<td></td>
</tr>
</tbody>
</table>
Due to the complexity of accurate range when computing with multiplications regarding probabilities makes a scaling of each $\alpha$ and $\beta$ important. The complexity is how the probabilities are intending exponentially to absolutely nothing when $t$ expands large. The scaling aspect for scaling both forward and backward variable would depend on instances $t$ and in addition to the state $i$. The notation from the factor and it is done for every single $t$ and expresses $i$. Using the identical scale factor will be shown useful any time solving the parameter appraisal problem, where the scaling coefficients for $\alpha$ in addition to $\beta$ will cancel out each other specifically.

The subsequent procedure shows the calculation from the actual scale factor which mentioned above previously is also utilized to scale $\beta$. From the procedure the denotation may be the unscaled forward variable, denote the scaled forward variable and denote the temporary onward variable before scaling.
1. **Initialization**

Set $t = 2$;

$\alpha_i(i) = \pi_i, b_i(o_i)$, \hspace{1cm} $1 \leq i \leq N$

$\hat{\alpha}_i(i) = \alpha_i(i)$, \hspace{1cm} $1 \leq i \leq N$

$$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_i(i)}, \hspace{1cm} \hat{\alpha}_i(i) = c_1 \alpha_i(i)$$

2. **Induction**

$\hat{\alpha}_i(i) = b_i(o_i) \sum_{j=1}^{N} \hat{\alpha}_{j-1}(j) a_{ji}$, \hspace{1cm} $1 \leq i \leq N$

$$c_1 = \frac{1}{\sum_{i=1}^{N} \hat{\alpha}_i(i)}, \hspace{1cm} \hat{\alpha}_i(i) = c_1 \hat{\alpha}_i(i), \hspace{1cm} 1 \leq i \leq N$$

3. **Update time**

Set $t = t + 1$; Go back to step 2 if $t \leq T$

Otherwise, terminate the algorithm.

4. **Termination**

$$\log P(O | \lambda) = - \sum_{t=1}^{T} \log c_t$$

By using the logarithm in step 4 is used as a result of precision range in fact it is only used compared to other probability in various other models.

**5.6.12 Observation Sequence Probability Storing**

The actual $\log(P(O | \lambda))$ can be saved in the matrix to view the adjustment from the restimation sequence For each and every iteration we have a summation from the sum($\log$(scale)), overall probability. This summation is when compared to the previous summation within previous iteration. If the difference between
measured values is under a predetermined value, then the optimum might be assumed to obtain. If necessary a fixed number regarding iterations could be set to lessen calculations.

![Graph](image)

**Fig. 5.36: graph for iteration and sum(log(scale))**

### 5.6.13 SIMULATION RESULT OF HIDDEN MARKOV MODEL

- Save the Hidden Markov Model for a particular utterance

After the re estimation is performed. The model is saved to characterize that distinct observation sequences, i.e. an isolated word. This style is then used for recognition. The style is represented with the following notation $\lambda=(A, \mu, \Sigma)$. 
5.6.14 SIMULATION RESULTS & RECOGNITION ACCURACY

This chapter presents the experimental results obtained from the proposed approaches namely LPC, MFCC, Wavelet analysis, Dynamic Time Warping, HMM that was applied to the isolated Telugu digits recognition. The effectiveness of the algorithms is measured through the analysis of the results.

The below table 5.2 shows the Recognition Accuracy for all the combination proposed in the Research with stored speech sample and with real time speech sample.
From the given table we can conclude that, with comparison of all the techniques with stored speech signal is giving 100% accuracy but in case of real time due to noise, accuracy decreased even in noise. Wavelet & DTW is giving the best accuracy rate of all the above combination.

**Table 5.2: Recognition Accuracy for all the combination proposed in Research**

<table>
<thead>
<tr>
<th>Word</th>
<th>LPC+DTW</th>
<th>MFCC+DTW</th>
<th>MFCC+HMM</th>
<th>WAVELET+DTW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With stored Speech</td>
<td>Real time Speech</td>
<td>With stored Speech</td>
<td>Real time Speech</td>
</tr>
<tr>
<td>Sunna</td>
<td>100</td>
<td>70</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>Okkati</td>
<td>100</td>
<td>75</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>Rendu</td>
<td>100</td>
<td>85</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>Moodu</td>
<td>100</td>
<td>70</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>Nalugu</td>
<td>100</td>
<td>70</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>Idu</td>
<td>100</td>
<td>85</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>Aru</td>
<td>100</td>
<td>80</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>Edu</td>
<td>100</td>
<td>70</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>Enamidi</td>
<td>100</td>
<td>70</td>
<td>100</td>
<td>70</td>
</tr>
<tr>
<td>Tommidi</td>
<td>100</td>
<td>80</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>Average</td>
<td>100</td>
<td>75</td>
<td>100</td>
<td>75.5</td>
</tr>
</tbody>
</table>
- **Speech Recognition (LPC & DTW):** This combination of recognition system giving 76% in real time system but in case of stored speech the result obtain is 100%. LPC is based on order as the order increases the recognition time is decreasing but there is no change in recognition accuracy.

- **Speech Recognition (MFCC & DTW):** This combination of recognition system having more coefficients i.e. is 39 the performance of each digit is varying and also overall performance is better than LPC and also recognition time is improved.

- **Speech Recognition (MFCC & HMM):** This combination of recognition system having more coefficients i.e. is 39 the performance of each digit is varying and also overall performance is better than LPC and also recognition time is improved. The difference as compare to above combination is HMM this matching technique is time taking but accuracy of recognition is good.

- **Speech Recognition (WAVELET & DTW):** This combination of recognition system better in real time recognition system because there is filter bank in wavelets this will avoid noise to enter the system this makes system to increase the recognition accuracy.
CONCLUSION

This chapter discussed about speech recognition techniques, before recognizing the speech, feature of the speech signal was extracted using LPC, MFCC and wavelet methods. Out of these three methods wavelets feature extraction technique shown better results, and this features can be applied to recognition techniques like Dynamic time warping and Hidden markov models their implementation procedure were discussed. Different combinations of feature extraction methods and recognition techniques are implemented to recognize isolated telugu words. Out of above methods wavelet and DTW combination gives good accuracy for recognizing telugu isolated words for both real time and stored words.