CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Artificial speech has been a dream of the humankind for centuries. Researchers are working to fulfill this dream from last few decades [20]. This effort has made tremendous development in the field of developing speech synthesis system, with high accuracy and intelligibility. But still it lacks the ability to produce sound with naturalness. A basic Text to Speech (TTS) Synthesis system consists of two phases. The first category is called the text analysis, in which the input text is transcribed into a phonetic or other linguistic representation, and the second category is called the generation of speech waveforms, in which the acoustic output is generated from this prosodic and phonetic information. These two phases are generally called as high- and low-level synthesis. The basic setup of speech Synthesis System of human is as shown in Figure (2.1)

![Block Diagram of a Basic Speech Synthesis System](image)

Figure 2.1: Block Diagram of a Basic Speech Synthesis System

Many years ago, von Kempelen[7] demonstrated that the speech-production system of the human being could be modeled. Speaker identification research continues today under the realm of the field of digital signal processing. Synthesis of speech and automatic generation of speech waveform is under study for the
last few decades[23]. Although there have been tremendous development had happen and many has produced synthesizers with a high degree of accuracy and intelligibility, but still it lacks to produce better quality sound and naturalness. As far as TTS synthesis process concerned, it has two main basic phases. The first phase, in TTS processing is called High Level Synthesis that is associated with the linguistic representation of input text, which is known as Phonetic Representation. The second phase is called Low Level Synthesis, which is associated with the production of acoustic output based on the phonetic and prosodic information collected or produced in the first phase i.e. in the High level Synthesis. The basic setup of Phonetic Representation is as given in figure (2.2).

![Figure 2.2: Phonetic Representation](image)

One of the simplest ways to produce synthetic speech is to play long pre-recorded samples of natural speech such as single words or sentences[32]. This method produces high quality and naturalness. However, it has got the limitation of limited vocabulary and limited number of informants [61]. The method seems to be suitable for some announcing and information system. On the other hand it is very difficult to create a database of all words of a particular language in a recorded form. Even if it is possible, it will not be justified to call it as Speech Synthesis system. Thus to design a system with greater flexibility, we need to consider shorter pieces of speech signals, such as syllables, phonemes, diaphones, or even shorter segments.
2.2 Research, Advances, Contributions

In this section, the different approaches to speech recognition, which are so-far used by the different group of researchers world wide have been described. Speech recognition deals with the recognition of the specific individual speech sounds. Cepstral Method Evaluation in Speech Formant Frequencies Estimation of English language is well documented by Med Ali kammoun(2004)[35]. Analysis and Synthesis of Vowels of Romanian phoneme is documented by Alina Nica (2006)[41]. They have presented several methods which are commonly used in estimation of the main speech signal features. Voiced/unvoiced detection of speech signals using empirical mode decomposition model by Md. Khademul Islam Molla[25] in year (2007). He demonostrated the discrimination of voiced/Unvoiced. Some of the commonly used techniques speech analysis and synthesis are –LPC Cepstral Coefficients, MFCC, Hidden-Markov Model (HMM) and Artificial Neural Network (ANM) methods etc. The mel-frequency cepstral coefficients (MFCCs) introduced by Davis and Mermelstein[52] is perhaps the most popular and common feature for SR systems.

2.3 Approaches to Speech Synthesis Process

A simple way to produce a synthetic speech is to record some samples of natural speech such as word and sentences, then play them for the synthesis. This procedure produces naturalness but has got some limitations of limited vocabulary and limited number of informants. This method is only suitable for simple systems like information or announcement system. It is also difficult to record and create a database for all the words of a particular language. To design a Speech Synthesis system, that will have a greater flexibility, we must consider shorter pieces of a speech signals, such as syllable, phoneme, diaphones or if possible even more shorter segments.
Formant synthesis is the widely used another method to produce synthetic speech [27], it is completely based on the source-filter-model of speech production system. Here the excitation signal may be either voiced with fundamental frequency (F₀) or with the unvoiced noise. For voiced consonants and some aspirated sounds a mixed excitation of these two may also be used.

In theory, human speech production system directly is to model to get the most accurate method to generate artificial speech[70]. This method is called articulatory synthesis, which is basically involves the models of the human articulators and vocal cords. The vocal cord model is generally used to create an appropriate excitation signal, which may be the example of a two-mass model with two vertically moving masses. From the above concept we can categorized the methods for speech synthesis into three main groups-

- **Articulatory synthesis**, which attempts to model the human speech production system directly.
- **Formant synthesis**, which models the pole frequencies of speech signal or transfer function of vocal tract based on source-filter-model.
- **Concatenative synthesis**, which uses different length pre-recorded samples derived from natural speech.

All the above mentioned methods have their own merits and demerits, so it will be difficult to select one as appropriate. Concatenative and Formant synthesis process shows a very promising result, but Articulatory synthesis is also rising as a potential method for the future.

### 2.4 History of Development of Speech Synthesis

Attempts to developed artificial speech synthesis process began in the early 18th century[64]. This dream of producing artificial speech has motivated
the researchers and scientists towards a remarkable development in the speech synthesis process over the last few decades [79].

2.4.1 Early attempts in the Speech Synthesis Process

Russian Professor Christian Kratzenstein[80] was the first person to explained physiological differences between five long vowels (/a/, /e/, /i/, /o/, and /u/) and in 1979 he made an apparatus to produce them artificially. After a few years of Christian Kratzenstein, Wolfgang von Kempelen[82](1791) introduced his "Acoustic-Mechanical Speech Machine", through which he was able to produce some single sounds and sound combinations.

Charles Wheatstone[19] constructed a well known version of von Kempelen's speaking machine in the mid of 1800 century. After that Alexander Graham Bell[85] and his father, inspired by Wheatstone's speaking machine and produced a same kind of speaking machine. Stewart[69] was the first person to introduce the first full electrical synthesis device in 1922.

In 1939 Homer Dudley introduced VODER (Voice Operating Demonstrator) which is considered to be the first device as a speech synthesizer. It consisted of wrist bar for selecting a voicing or noise source and has a foot pedal to control the fundamental frequency of speech. The potential for producing artificial speech were well demonstrated in this method but the speech quality and intelligibility were far from good.

In 1952, at Haskins Laboratories, Franklin Cooper[89] and his associates developed a Pattern Playback synthesizer. After that in 1953, Walter Lawrence[60] introduced the first formant synthesizer, PAT (Parametric Artificial Talker). In 1958, George Rosen[66] introduced the First articulatory synthesizer at the Massachusetts Institute of Technology, M.I.T.
2.4.2 Development during 1960's

Linear Predictive Coding (LPC) came into trend in the mid of 1960's. Noriko Umeda[110] and his companions in 1968 was produced the first full text-to-speech system for English in the Electrotechnical Laboratory, Japan.

The MITalk laboratory text-to-speech system developed at M.I.T in 1979 was demonstrated by Allen, Hunnicutt, and Klatt[88]. Later this system was used also in Telesensory Systems Inc(TSI). After two years of MITalk system, Dennis Klatt[31] introduced his new well known Klattalk system. In today's speech synthesis system has a great influenced of the technology used in MITalk and Klattalk.

2.4.3 Development during 1970's and 1980's

The commercial text to speech and speech synthesizer were developed in the late 1970's. In this context Votrax chip was considered as the first integrated circuit for the speech synthesis. An inexpensive Votrax-based Type-n-Talk system was developed in 1978 by Richard Gagnon[12]. After two years of 1980, Texas Instruments developed the Linear Prediction Coding (LPC)-based Speak-n-Spell synthesizer that was based on low-cost Linear Prediction Synthesis chip (TMS-5100), which was used for an electronic reading aid for children. Street Electronics developed Echo low-cost diphone synthesizer (track 29) in 1982, which was based on same chip as in Speak-n-Spell (TMS-5220). At the same time Speech Plus Inc. introduced the Prose-2000 text-to-speech system. Famous DECTalk and Infovox SA-101 synthesizer were developed after 1982.
2.4.4 Developments in the remaining years

Now a day, speech synthesis technologies are more complicated and sophisticated. Hidden Markov model (HMM) is the most famous method developed in recent technology [47] and it was applied to speech recognition from late of 1970's.

A Hidden Markov Model (HMM) is a collection of states connected by transitions with two sets of probabilities: one is the transition probability which provides the probability for taking this transition, and other is an output Probability Density Function (PDF) which defines the conditional probability of emitting each output symbol from a finite alphabet. The Artificial Neural Networks (ANN) are also been applied in the field of speech synthesis and speech recognition in 1977. SYNTE2[85], was developed as speech synthesizer, which is considered to be the first speech synthesizer. About five years later SYNTE3 synthesizer was developed and it was a market leader in Finland for many years. After that many speech synthesizer were developed for example, Amertronics, Brother Caiku, Eke, Humanica, Seppo, and Task, which all were based on the Votrax speech synthesis chip.

2.5 Methods of Analysis and Review of Measurement Techniques

2.5.1 Linear Predictive Coding (LPC)

Linear Predictive Coding (LPC)[16] is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the previous value of the signal. Linear prediction is a method for signal source modeling dominant in speech signal processing and having wide application in other areas. Linear Predictive Coding (LPC) is the most powerful and famous
speech analysis techniques. The glottis (the space between the vocal cords) produces the sound, which is characterized by its intensity and frequency. The basic flow chart of LPC is as given below in figure (2.3).

![LPC Flowchart](image)

Figure (2.3): A simple LPC flowchart

The basic problem of the LPC system is to determine the formants from the speech signal. The solution of this problem is a difference equation, which expresses each sample of the signal as a linear combination of previous samples. Such an equation is called a linear predictor i.e. Linear Predictive Coding. The coefficients of the difference equation (the prediction coefficients) characterize the formants. Therefore, the LPC system needs to estimate these coefficients. The estimation is made by minimizing the mean square error between the predicted signal and the actual signal.
The basic idea behind the LPC model is that a given speech sample \( s(n) \) at discrete time \( n \), can be approximated as a linear combination of the past \( p \) speech samples [45] such that

\[
s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) \ldots (2.1)
\]

Where the coefficients are \( a_1, a_2, \ldots, a_p \) assumed to be constants over the speech analysis frame. The equation (2.1) can be converted to an equality by including an excitation term \( G u(n) \),

\[
s(n) = G u(n) + \sum_{i=1}^{p} a_i s(n-1) \ldots (2.2)
\]

Where, \( u(n) \) is normalized excitation and \( G \) is the gain of excitation. Expressing equation (2.2) in Z domain we get the relation:

\[
S(z) = G u + \sum_{i=1}^{p} a_i z^{-i} s(z) \ldots (2.3)
\]

Leading to the transfer function:

\[
H(z) = \frac{s(z)}{G u(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \ldots (2.4)
\]

Based on our knowledge that the actual excitation function for speech is essentially either voiced speech sounds or an unvoiced sound.

2.5.2 LPC Analysis

The relation between \( s_n \) and \( u_n \) is defined (based on the speech production model).
We consider the linear combination of past speech samples as the estimate \( \hat{s}(n) \), defined as,

\[
\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.6}
\]

The predictor error, \( e(n) \), is defined as,

\[
e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.7}
\]

And the error transfer function is,

\[
A(z) = \frac{\hat{e}}{e} = 1 - \sum_{k=1}^{p} a_k z^{-1} \tag{2.8}
\]

The basic problem of linear prediction analysis is to determine the set of predictor coefficients \( a_k \), directly from the speech signal so that the speech properties of the digital filter match those of the speech waveform within the analysis window.

To set up the equations that must be solved to determine the set of predictor coefficients, we define the short-term speech and error segments at time \( n \) as,

\[
s_{s}(m) = s(n + m) \tag{2.9}
\]

\[
e_{e}(m) = e(n + m) \tag{2.10}
\]

and tried to minimize the mean square error signal at time \( n \),

\[
E_{\kappa} = e_{\kappa}^2(m) \tag{2.11}
\]
Using equation (5.9) & (5.10) we can write

\[ E_n = \left[ \sum_m s_n(m) - \sum_{i=1}^p a_k s_n(m-k) \right]^2 \] ... (2.12)

To solve the equation (5.12) we put

\[ \frac{\partial E_n}{\partial a_k} = 0, \quad k = 1,2,3 \ldots p \] ... (2.13)

Where \( p \) is the number of speech sample.

Giving

\[ \sum_m s_n(m-i)s_n(m) = \sum_{i=1}^p a_k \sum_m s_n(m-i)s_n(m-k) \] ... (2.14)

This term \( \sum_m s_n(m-i)s_n(m-k) \) are related to the short term covariance of \( s_n(m) \) i.e.,

\[ \varphi(i,k) = \sum_m s_n(m-i)s_n(m-k) \] ... (2.15)

which can be expressed in compact notation as,

\[ \varphi_n(i,0) = \sum_{i=1}^p a_k \varphi_n(i,k) \] ... (2.16)

Which describe a set of \( p \) equations. It is readily shown that the minimum mean-square error, \( E_n \), can be expressed as:

\[ E_n = \sum_m s_n^2(m) - \sum_{i=1}^p a_k \sum_m s_n(m)s_n(m-k) \] ... (2.17)

Thus the minimum mean-squared error consists of a fixed term \( \varphi_n(i,0) \) and is depend on the predictor coefficients. To solve Equation (2.16) for the optimum coefficients \( a_k \), we have to compute \( \varphi_n(i,k) \), for \( 1 \leq i \leq p \) and \( 0 \leq k \leq p \), and
then solve the resulting set of p simultaneous equations. A method to solve these equations and compute the coefficients is the autocorrelation method.

2.5.3 The Autocorrelation Method

In this method, the segment $s_n(m) = 0$ are considered outside the interval $1 \leq m \leq N$, and $s_n$ is described by the equation

$$s_n(m) = s_n(m + n)w(m)$$

Where, $w(m)$, is identically zero outside the range $0 \leq m \leq N$, thus the speech sample for minimization can be expressed as

$$s_n(m) = \begin{cases} s(m + n) & 0 \leq m \leq N - 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.18)$$

Thus, if $s_n(m)$ differs from zero to interval $1 \leq m \leq N$, then the corresponding prediction error $e_n(m)$, for a linear predictor of order $p$, will be different from zero in the interval $1 \leq m \leq N + p$, which can be expressed as

$$E_n = \sum_{m=0}^{N-1+p} e_n^2(m) \quad (2.19)$$

and $\varnothing_n(i, k)$ can be expressed as,

$$\varnothing_n(i, k) = \sum_{m=0}^{N-1+p} s_n(m - i)s_n(m - k) \quad (2.20)$$

where $1 \leq p$ and $0 \leq k \leq p$

Or

$$\varnothing_n(i, k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m + i - k) \quad (2.21)$$
Since equation (2.21) is only a function of \( i - k \), rather than the two independent variables \( i \) and \( k \), the covariance function, \( \Phi_n(i,k) \), reduces to the simple autocorrelation function,

\[
\Phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{\infty} s_n(m)s_n(m+i-k)
\]

(2.22)

Since the autocorrelation function is symmetric, i.e. \( r_n(-k) = r_n(k) \), the LPC equations can be expressed as

\[
\sum_{k=1}^{p} r(|i-k|) \hat{a}_k = r_n(i), \text{ for } 1 \leq i \leq p
\]

(2.23)

and can be expressed in the matrix form, known as the Yule-Walker equations as follows:

\[
\begin{bmatrix}
    r_n(0) & r_n(1) & r_n(p-1) \\
    r_n(1) & r_n(0) & r_n(p-2) \\
    \vdots & \vdots & \vdots \\
    r_n(p-1) & r_n(p-2) & r_n(0)
\end{bmatrix}
\begin{bmatrix}
    \hat{a}_1 \\
    \hat{a}_2 \\
    \vdots \\
    \hat{a}_p
\end{bmatrix}

= 
\begin{bmatrix}
    r_n(1) \\
    r_n(2) \\
    \vdots \\
    r_n(p)
\end{bmatrix}
\]

(2.24)

The \((p \times p)\) matrix of autocorrelation value is a Toeplitz matrix (all diagonal elements are equal) and hence can be solved efficiently through several methods. One of the numerical methods is called the Levinson-Durbin algorithm.

2.5.4. The Levinson-Durbin Algorithm

A numerical solution of \( p \) equations in \( p \) unknowns, as needed in solving the Yule Walker equations, requires about \( p^3 \) multiply-add operations.
Levinson-Durbin algorithm [56] solves the Yule-Walker equations [46] in approximately \( p^2 \) multiply-add operations. The algorithm is as follows:

1. Defines \( r(o) \)

2. For \( i = 0,1,2,\ldots,p-1 \) then

\[
a_i r_{i+1} = \frac{r(i+1) + \sum_{k=0}^{j} a_k r(i+1-k)}{s_i}
\]

\[
b_i(l + 1) = s_i(l) - p_i(l + 1) \times 2
\]

\[
c_{i+1} = \gamma_{i+1} = a_{i+1} - p_{i+1}, 1 \leq k \leq l
\]

\[
d_{i+1} = -p_{i+1}
\]

The coefficients \( a_i, i \geq 1 \) recursively computed by this algorithm, correspond to the LPC coefficients, \( a_k \), in the equation (4.5).

2.5.5 The LPC- Cepstral Coefficient

In the present study, LPC-based Cepstral coefficients and phonetically important parameters are used as feature vectors. Cepstral weighted [50][26] feature vector is obtained for each frame by block processing of continuous speech signals. The analog speech waveform is then sampled and quantized analog-to-digital converter. To spectrally flatten the signal, the speech signal has been subjected to the pre-emphasis procedure through a first order digital filter whose transfer function has been given by

\[
H(z) = 1 - az^{-1}, \quad \text{for} \quad 0 \leq a \leq 1.0 \quad \ldots (2.25)
\]

Consecutive speech signal are taken as a single frame. To reduce the undesired effect of Gibbs phenomenon, the frames are multiplied by a windows function (Hamming window), which is given by [40],

44
\[ w(n) = 0.54 - 0.46 \cos \left( \frac{2\pi n}{N-1} \right), \text{ for } 0 \leq n \leq N \quad (2.26) \]

Where \( N \) is the number of sample in a block. Each frame of the windowed signal is next auto correlated to give

\[ r_f(m) = \sum_{n=0}^{N-m} x_f(n)x_f(n+m), \quad m=0, 1, 2, \ldots, p \quad (2.27) \]

Where \( p \), the highest auto correlated value and is the order of the LPC analysis.

2.5.6 LPC Parameter Conversion to Cepstral Coefficients

The LPC Cepstral coefficients, which are a set of values that have been found to be more robust, reliable feature set for speech recognition than the LPC coefficients\[76\]. These coefficients are obtained recursively as follows,

\[ c = \ln (\sigma^2) \text{ where } \sigma^2 \text{ is the gain term in the LPC model} \]

\[ c_m = a_m + \sum_{k=1}^{p} \left( \frac{k}{m} \right) c_{m-k} a_k, \quad 1 \leq m \leq p \quad (2.28) \]

\[ c_m = \sum_{k=1}^{p} \left( \frac{k}{m} \right) c_{m-k} a_k, \quad m > p \quad (2.29) \]

Equation (2.30) shows the computation of cepstral coefficients \( C_{p+1}, C_{p+2}, \ldots, C_p \).
2.5.7. Formant estimation for vowel recognition

The formant estimation is based on digital resonator technique the entire frequency range divided into fixed number of segment, (S = set of segments) each segment representing a formant frequency. Here for differentiation we have used the five point differentiation equation given in Equation (2.30). The interval should be chosen such that it is smaller than the value of the RR interval. The corresponding value of the differentiated wave at this point is calculated and a conditional loop is initiated to find out whether it is negative or positive. This step is the very important as the rest of the analysis depends on it. Depending upon what the value comes out to be, the lowest point to the left or to the right of the peak is calculated. This point is the R point.

\[
 f'(x) \approx -\frac{f(x + h) + 8f(x + h) - 8(x - h) + f(x - 2h)}{12h} \tag{2.30}
\]

The highest point to the left of the R value in the absolute differentiated wave is stored in the memory as the temporary point. From this point as a reference the lowest point to the left of this point is again calculated, to get the point. For the detection of the S point on the wave the procedure is the same as
that for the wave detection. Here too the proper detection of the R peak is very important. After getting the R peak point, the highest point to the right of the R value in the absolute differentiated wave is found out as temporary S point. From this point as a reference the lowest point to the right of this point is again calculated as the S point.

2.6 Mel- Frequency Cepstral Coefficients (MFCC)

In sound processing the Mel- Frequency Cepstral Coefficients (MFCC) is a representation of the short term power spectrum on a nonlinear mel scale of frequency.

Mel- Frequency Cepstral Coefficients (MFCC) are coefficients that collectively make up an MFC. The difference between the Cepstrum and the Mel frequency Cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system’s response more closely than the linearly spaced frequency warping can allow for better representation of sound, for example in audio compression.

MFCCs are commonly derived as follows [38].

1. Take the Fourier transform of a signal.

2. Map the power of the spectrum obtained above onto the mel scale, using triangular overlapping windows.

3. Take the logs of the powers at each of the mel frequencies.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum.
There can be variations of this process, for example differences in the shape or spacing of the windows used to map the scale [39]. For measurement of the formant frequency, Matlab 7.0 data acquisition Toolbar has been used in windows 7 environment.

2.7 Hidden Markov Model (HMM)

The Hidden Markov Model (HMM)[70] is a most famous statistical method for modeling a wide range of time series data. In the context of Natural Language Processing (NLP)[34], HMMs have been applied with great success to problems such as part-of-speech tagging and noun-phrase chunking. The Hidden Markov Model (HMM) is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence. HMMs have found application in many areas interested in signal processing, and in particular speech processing, but have also been applied with success to low level NLP tasks such as part-of-speech tagging, phrase chunking, and extracting target information from documents. Andrei Markov gave his name to the mathematical theory of Markov processes in the early twentieth century[3], but it was Baum and his colleagues that developed the theory of HMMs in the 1960s.

The overall flow chart to determine the HMM data is as given below
Markov Processes Diagram figure 2.5 depicts an example of a Markov process. The above figure contains the functional blocks of HTK tool used for speech recognition. Sample block is used to store all the collected samples. Feature extraction block is used to extract the features (Feature vectors like MFCC etc) and store into 6x6 vector table. Using Polynomial expansion function windowing performed. The collected 6x6 vector table is stored in speech unit table, which is further expanded using correlator to create sequence vector. Thus adding with Mean, States etc corresponding HMM is generated for each sample. In viterbi block best possibility is searched from the HMM set using backward and forward model tracing to get the corresponding output.
Figure 2.6- Markov process example[1]

The model presented describes a simple model for a stock market index. The model has three states, Bull, Bear and Even, and three index observations up, down, unchanged. The model is a finite state automaton, with probabilistic transitions between states. Given a sequence of observations, example: up-down-down we can easily verify that the state sequence that produced those observations was: Bull-Bear-Bear, and the probability of the sequence is simply the product of the transitions, in this case $0.2 \times 0.3 \times 0.3$. Hidden Markov Models figure 2.6 shows an example of how the previous model can be extended into a HMM. The new model now allows all observation symbols to be emitted from each state with a finite probability. This change makes the model much more expressive.
and able to better represent our intuition, in this case, that a bull market would have both good days and bad days, but there would be more good ones. The key difference is that now if we have the observation sequence up-down-down then we cannot say exactly what state sequence produced these observations and thus the state sequence is 'hidden'. We can however calculate the probability that the model produced the sequence, as well as which state sequence was most likely to have produced the observations. The next three sections describe the common calculations that we would like to be able to perform on a HMM. The formal definition of a HMM is as follows:

$$\lambda = (A, B, \pi)$$  \hfill (2.31)

$S$ is our state alphabet set, and $V$ is the observation alphabet set:
We define Q to be a fixed state sequence of length T, and corresponding observations O:

\[
Q = q_1, q_2, \cdots, q_T \\
O = o_1, o_2, \cdots, o_T
\]  

(2.34)  

(2.35)

A is a transition array, storing the probability of state j following state i. Note the state transition probabilities are independent of time:

\[
A = [a_{ij}] , a_{ij} = P(q_t = s_j | q_{t-1} = s_i) .
\]  

(2.36)

B is the observation array, storing the probability of observation k being produced from the state j, independent of t:

\[
B = [b_i(k)] , b_i(k) = P(x_t = v_k | q_t = s_i).
\]  

(2.37)

\( \pi \) is the initial probability array:

\[
\pi = [\pi_i] , \pi_i = P(q_1 = s_i).
\]  

(2.38)

Two assumptions are made by the model. The first, called the Markov assumption, states that the current state is dependent only on the previous state, this represents the memory of the model:

\[
P(q_t | q_{t-1}^{t-1}) = P(q_t | q_{t-1})
\]  

(2.39)

The independence assumption states that the output observation at time \( t \) is dependent only on the current state, it is independent of previous observations and states:
\[ P(o_t | o_{t-1}^t, g_1^t) = P(o_t | q_t) \] (2.40)

**Figure 2.8- A trellis algorithm**

### 2.8 The Proposed Approaches:

The main approach used in this thesis work is **acoustic phonetic approach**. In the present study, the Fast Fourier Transform (FFT) is used to obtain the frequency domain signal from input temporal time domain signal. Then FIR filter is used to filter the noise content of the speech signal before performing fast Fourier transform on the signal. Linear Predictive Coding is used to locate the format locations as LPC coefficients emphasize the location of the formants in the frequency spectrum. Cepstral coefficients of a framed signal are generated using LPC technique. The LPC coefficients would provide a good generalization of the
speaker's unique vocal characteristics. The EMD based method is used to word level characterization of Bodo and Assamese words. The Mel frequency Cesptral Co-efficient (MFCC), Hidden Markov Model (HMM) are used analyze the features of Bodo and Assamese phonemes. The sequence of Tasks performed in the present study are as follows:

1. Use of Fast Fourier Transform: This will transform the continuous time domain signal to frequency domain signal.

2. Use of FIR filter to remove noise: This will be used to filter the noise content of the voice signal recorded.

3. Use Linear Predictive coding: This will produce a vector of coefficients that represent a smooth spectral envelope of the discrete Fourier transform magnitude of an input speech signal that shows the formant locations spectrally.

4. Study and analyze the formant frequencies i.e., F1, F2 and F3 for each word and to find the distinctive differences between male and female speakers.

5. Study and analyze LPC cepstral measures to identify between male and female speakers of Bodo and Assamese. Also use the MFCC and HMM techniques to analyze the feature of Bodo and Assamese phonemes and their diversity.

6. Analyze the power spectral density of each word under consideration between male and Female speakers.

7. Analyze the fundamental frequency using five point differentiation technique to verify male and female speakers.

8. EMD based method is used to characterization of Bodo and Assamese Speakers.

9. A Comparative study of LPCC, MFCC and HMM techniques.
2.9 Organization of the Thesis

The thesis is organized in eight chapters.

Chapter 1: Gives an introduction to Speech Research. Speech Production mechanism has been discussed in this chapter supported by mathematical formulation. The chapter also discusses about some commonly used mathematical concept used in speech processing in details.

Chapter 2: Literature review in details. This chapter begins with the speech synthesis process and its types as proposed by different researchers. Different techniques which are commonly used in Speech Recognition Process have also been introduced in this chapter. The Speech Research works Advances and contribution. Discuss about the early speech production and representation technology.

Chapter 3: In this chapter the Methodology that have been adopted is illustrated.

Chapter 4: This chapter gives Introduction to Bodo and Assamese Language. The brief history of Bodo and Assamese Script and its different dialects has been elaborated in this chapter with the characteristic of Bodo and Assamese phonemes.

Chapter 5: In this chapter Formant Estimation of Different Types of Words of Bodo and Assamese languages have been made.

Chapter 6: This chapter concerned with Acoustic Analysis of Bodo and Assamese, vowels, Consonant and words.

Chapter 7: In this chapter Analysis and Synthesis of Bodo and Assamese using Hidden Markov Model have been made.

Chapter 8: This is a concluding chapter. Here analytical discussion on LPCC, MFCC and HMM techniques and Future scope of the study have been discussed.