3. DEVELOPMENT AND IMPLEMENTATION OF SYSTEM

3.1 Problem Definition

To Develop Text-to-Speech Synthesizer for Marathi language Using Concatenative synthesis strategies.

Propose objectives of the system are as:

- More Natural Sounding Output
- Intelligibility of Voice Output

To achieve these objectives:

- Data base creation is Maximum number of words from Marathi Barakhadi (Varnamala) and less number of syllables & Phonemes.
- Development & implementation TTS Code & Front end is in Visual Basic 6.0 (Software)

To analyze the performance of ‘MATTS” Test to carry out as:

- Subjective Test
- Spectrogram test

The most important qualities of a speech synthesis system should naturalness and intelligibility.

- Naturalness describes how closely the output sounds like human speech, while
- Intelligibility is the ease with which the output is understood.

The ideal speech synthesizer is both natural and intelligible. Proposed work had tried to maximize the Speech synthesis systems with both the characteristics naturalness and intelligibility.

3.2 Speech Synthesis System

A Text-to-Speech (TTS) Synthesizer is a computer based system that should be able to read any text and digit in Marathi, whether it was introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system. Here words are submitted through key a board which was converted in to Marathi fonts. Text to speech system is to convert given text into a spoken waveform. Main components of text to speech synthesizer system are:
3.3 Text processing front end

Following are the preparations to develop the front end for Marathi TTS.

3.3.1 Indian Language Scripts

The basic units of the writing system in Indian languages are Aksharas, which are an orthographic representation of speech sounds. An Aksharas in Indian language scripts is close to a syllable and can be typically in the form: C, V, CV, CCV, VC and CVC where C is a Consonant and V is a vowel, CV structure is usable to identify the words, syllables and phonemes.

3.3.2 Input Text Format

The scripts of Marathi language are stored in the form words, syllables & phonemes in a specific fonts, then link with code through MS-ACCESS.

3.3.3 Mapping: Non Standard words to standard words

In practice, an input text such as news article, books consists of standard words (whose entries could be found in the dictionary) and non-standard words such as initials, digits, symbols and abbreviations. Mapping of non-standard words to a set of standard words depends on the context, and it is a non-trivial problem of TTS conversion [25].

3.3.4 Standard word to Phoneme

Generation of sequence of phoneme units for a given standard word is referred to as letter to sound rules. The complexity of these rules and their derivation depends on the nature of the language. Understanding of this phenomenon is important to build a good text processing module [41].
3.4 Speech Generation Component

After accessing the sequence of phonemes, syllables and words. The objective of the speech generation component is to synthesize the acoustic waveform. Speech generation has been attempted by concatenating the recorded speech segments.

Current state-of-art speech synthesis generates natural sounding speech by using large number of speech units. Storage of large number of units and their retrieval in real time is feasible due to availability of cheap memory and computation power. The approach of using an inventory of speech units is referred to as unit selection approach. It can also be referred to as data-driven approach or example based approach for speech synthesis [16].

The issues related to the unit selection speech synthesis system are:

1) Choice of unit size,
2) Generation of speech database,
3) Criteria for selection of a unit.

The criterion for selection of a unit depends on how well it matches with the input specification and other units in the sequence. Cost is associated for mismatch with the input specification and other units in sequence, and is referred to as target and concatenation cost respectively. A unit which is responsible to minimizes the cost of the target and effect of the concatenation [20].

3.4.1 Choice of unit size

An inventory of larger size of units such as sentences, phrases and words could constitute an ideal speech database for speech generation. However, if the size of the units is large, the coverage of all possible words, phrases, proper nouns, and other foreign words may not be ensured. Sub word units make it easier to cover the space of acoustic units but at the cost of more joints. The choice of sub word unit is also related to the language itself.

3.4.2 Generation of unit selection data base

There are two issues about the generation of unit selection databases.

1) Selection of utterances which has the coverage of all possible units,
2) Recording of these utterances by a good voice talent. Selection of utterances is linked with the choice of the unit size. The larger the size of the unit the larger would be the number of utterances for the coverage of the units [26].

3.5 Synthesis strategies

There are two primary technologies for generating synthetic speech waveforms

(i) Formant Synthesis &
(ii) Concatenative Synthesis

Proposed work uses the concatenative synthesis of unit selection method. One more strategy to implement TTS is combination of formant & concatenative synthesis (Hybrid synthesis). It depend on the specific application, where memory size is constraint. It is embedded TTS, portable TTS where the naturalness and coverage i.e. intelligibility is not the major issue.

3.5.1 Concatenative over Formant Synthesis

The objective of this research work is to achieve greatest naturalness, which is possible using unit selection method of concatenative synthesis. This methodology has the advantage in its simplicity, i.e. there is no mathematical model involved. Speech is produced as natural like human speech. Concatenative synthesis is based on the concatenation (or stringing together) of segments of recorded speech. It is concatenation of words, syllables & phonemes. Generally concatenative synthesis produces the most natural-sounding synthesized speech. Many systems based on formant synthesis technology generate artificial, robotic-sounding speech, which would never be mistaken as a human speech [24].

3.6 Process Flow of Marathi TTS - MATTS

The text input is either non-standard words or standard words. If the input text is a number then it is handled by a digit processor. If input text is word then it searched in the word database. If the word does not exist in the database then it is cut into syllables and syllables are searched in the syllable database. If the corresponding syllable does not exist in the database then word is formed by concatenating phonemes in the phoneme database and played after the proper concatenation. Fig.3.1 shows the flow chart of the MATTS.
3.7 Block Diagram

The block representation of Marathi TTS is shown in fig 3.2, it accept the Marathi text, perform the text processing and searching through various blocks of TTS System i.e. words,
syllables and phonemes. It normalized the text according to digits, numbers, words & etc. After the process concatenation, concern audio file (.wave file) will be sounded.

Two databases will be maintained viz. audio database that stores the audio files and textual database that stores the text files corresponding to audio files in the audio database. The textual database is required to search the index of the required word in the audio database. TTS code is implemented for:

- Text input Processing
- Digit Processor
- Database Search
- Concatenation Process & play

3.7.1 Text input

The design for text processing front-end is done in Visual Basic 6.0 such that text can be entered either through on-screen keyboard or hardware keyboard. The text window has been
programmed to accept around five hundred words; however number of words can be easily extended by changing the coding. However database inventory is created up to 6000 words with phonemes & syllables. 6000 words are so chosen where in, words can cover all possible sentences of Marathi language newspaper. Linguistic dictionary is prepared to cater all possible words, if not then for the concatenation of words, syllable & phonemes database helps in little extent. This TTS system is able to read any written text, even if it contains numbers, dates, addresses, telephone numbers and bank account numbers. Implementation of text normalization, pre-processing and tokenization is achieved through algorithm develop in VB 6.0 The “font” property of the text box has been selected to as “KIRAN”, so all words typed in the text box appears in KIRAN font.

The objects shown on the front end is coded in Visual Basic 6.0.
For e.g. PLAY, PAUSE, LOUDSPEAKER, FORWARD, START.
The text window shown is a multi-line text box control that has been programmed to accept around five hundred words; however number of words can be easily extended by changing the coding.
3.7.2 Keyboard Design

The format for KIRAN font on standard hardware keyboard is as follows:

Press \texttt{Alt + nnnn} on (Numeric Keypad) To Type Following Characters

<table>
<thead>
<tr>
<th>0200</th>
<th>0201</th>
<th>0202</th>
<th>0203</th>
<th>0204</th>
<th>0205</th>
</tr>
</thead>
<tbody>
<tr>
<td>उ</td>
<td>च</td>
<td>ल</td>
<td>च</td>
<td>ड</td>
<td>ल</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0210</th>
<th>0211</th>
<th>0212</th>
<th>0213</th>
<th>0214</th>
<th>0215</th>
</tr>
</thead>
<tbody>
<tr>
<td>ह</td>
<td>ह</td>
<td>घ</td>
<td>घ</td>
<td>ढ</td>
<td>ढ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0222</th>
<th>0223</th>
<th>0224</th>
<th>0225</th>
<th>0226</th>
<th>0227</th>
</tr>
</thead>
<tbody>
<tr>
<td>र</td>
<td>+</td>
<td>-</td>
<td>x</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0232</th>
<th>0233</th>
<th>0234</th>
<th>0235</th>
<th>0236</th>
<th>0237</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
<td>}</td>
<td>[</td>
<td>]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0242</th>
<th>0243</th>
<th>0244</th>
<th>0245</th>
<th>0246</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>/</td>
<td>\</td>
<td>%</td>
<td>ण</td>
</tr>
</tbody>
</table>

Fig 3.4: Keyboard Layout for Marathi TTS – Kiran Font

Following are characters of Marathi \textit{i.e.} Barakhadi for Marathi Language

| A | Aa | [ | [- | { | } | e | eo | Aao | AaO | AM | A: | P |
Fig 3.5: Vowels & Consonants of Marathi language

Special Instructions

- Type ' x ' after क फ to add space.

In Devnagari, only क फ these characters have a bar in the center. In order to get Matra, Ardhchandra, Anuswar or Ukar properly at the centre for these characters, for these letters, after typing the letter itself, type ' x ' after Matra, Ardhchandra, Anuswar or Ukar [77].
3.8 Forms

The four forms were developed to process TTS system. This form provides login the system, flashing the product information which contains version, relative information for operation, specification. This form can be modify by changing the information in the code. Main form is usable to run the projects i.e. project 1, project 2, inter linking can be possible between the projects for more number of sentences as there is limitation on one window i.e. in document
form. Entering of the text data will be performed through document form. Capacity of submitting the words can be modify in this form. Different forms designed in VB6.0 are as follows:

- Form login
- Form splash
- Form document
- Form main

1. **Form Login**

   **Purpose of form login:**
   1. The form accepts user name and password.
   2. The form is modified by double clicking on it.
   3. To change the passwords, change the value in quotes txt Password. Text = "****".

![Form login screenshot](image)

**Fig.3.7: Snap shot of Form login**

2. **Form Splash**

   **Purpose of form splash:**
   1. The form shows the information about the TTS system.
   2. The form gives process of operation.
   3. Form can be modify according to desired information.
3. **Form Main**

   Purpose of form main:
   1. The form provides option for various menus like File, Edit, View, Language etc
   2. The form also provides a tool bar for all above menus & execution of projects.

4. **Form Document**

   Purpose of form Document:
   1. The text is entered into the text box through either hardware or on screen keyboard.
   2. On press of play button ‘image2’ the actual processing starts.
   3. The inputted text is checked to be a number so as to be sent to digit processor.
4. For a word, the word is binary searched in “char_name” column of database.mdb
5. If the word is found then .wave file corresponding to the word in the column of “Char_voice”, after concern voice files is played.

![Image](image.jpg)

Fig.3.10: Snap shot of form document

### 3.9 System Processing Algorithm

1. Enter the text in Text window.
2. Press the play button.
3. The play button is named as image2.
4. On clicking the play button, the image2_click event is activated.
5. All variables for programming are initialized.
6. Two string variables are required.
   (a) str- for single word. (b) str1- stores array of strings
7. The “Split function” is assigned with entire text box contents.
8. The split function separates the paragraph into words/strings depending upon the space entered.
9. All these split words are stored in “str1”.

10. A loop is executed with maximum count of ‘500’, this value indicates the maximum
    length of single string that can be entered in text box.

11. Maximum count size can be changed depending upon requirement.

12. The physical dimension of text box is immaterial because of the presence of scroll bar.

14. Loop is executed as per the following schedule:
    (i) Take the first string from str1 into the variable str.
    (ii) Apply the comparison algorithm thereby comparing ‘str’ with every element in
         column ‘char_name’ of database .mdb.
    (iii) If the comparison fails the go to (vi) else
    (iv) Borrow the corresponding entire row into temporary variables.
    (v) Play the .wav file whose path is mentioned in the Char_voice element of the row.
    (vi) If comparison fails, break the word into phonemes, apply steps (i) to (v) for each
         phonemes separately.
    (vii) Loop continues till entire str1 array variable is empty of words.

**3.9.1 Digit Processing Algorithm**

1. Check if the number is greater than 100000. If yes then continue otherwise
   follow step 12.

2. Get the quotient of no/100000 in variable quo. Store the remainder in variable
   rem for further processing.

3. Get the wave file corresponding to the no. in the quo by querying the digit in the
   textual database and play.

4. Also gets the .wave file corresponding to the word लाख and play.
5. Get the quotient for rem/1000 in quo. Store the new remainder in variable rem for further processing.

6. Get the wave file corresponding to the no. in quo by querying the digit in the textual database and play.

7. Also gets the .wave file corresponding to the word छज्जा and play.

8. Get the quotient for rem/100 in quo. Again store the new remainder in variable rem for further processing.

9. Get the wave file corresponding to the no. in quo by querying the digit in the textual database and play.

10. Also gets the wave file corresponding to the word सैं and play.

11. Finally gets the .wave file corresponding to the number stored in the rem variable by querying the digit in the textual database and play.

12. If the number is less than 100000 and greater then 1000 then follow steps 5 to 11

13. If the number is less than 1000 and greater than 100 then follow steps 8 to 11.

14. If the number is less than 100 then get the .wave file corresponding to the number by querying the digit in the textual database and play.

3.9.2 Time Processing Algorithm

Example: 10:30 गीत

1. Play the .wave file corresponding to the number before the symbol ‘:’

2. Play the .wave file “vajun.wav” after the symbol ‘:’

3. Play the .wave file corresponding to the numbers after the symbol “:”

4. Play the .wave file “minit.wav”
3.9.3 Date, Phone Number Processing Algorithm

- Special adjustments are done in the code for reading dates, provided to the user enters the date in following format dd | mm | yy
- For reading numbers in sentences like number of people, type “नं” preceding the number.
- For reading phone numbers or mobile numbers just type the number.

3.10 Database Creation

Two databases had been created viz. audio database that stores the audio files and textual database that stores the text files corresponding to audio files in the database [20].

3.10.1 Textual Database

Fig. 3.11.(a,b) shows the snapshot of textual database. The textual database is required to search the index of the required word in the audio data base. All words are stored in one file consecutively. The file is a Microsoft Access 2007 file named “database.mdb”. Database.mdb consists of three columns

a) Char_id: Indicates serial number
b) Char_name: the actual spelling of the word in KIRAN font.
c) Char_voice: the name of the wav file corresponding to the word.

Fig. 3.11: Snap shot of Textual Database for words
Fig. 3.11(a): Snap shot of Textual Database for syllables

Fig. 3.11(b): Snap shot of Textual Database for phoneme \textit{i.e.} Barakhadi
3.10.2 Audio Database

The audio database creation was a typical process consisting of tedious recording of frequently used words in Marathi language by a voice talent. All recorded words are stored in the folder “voice files”. Fig. 3.12 shows the snapshot of audio database.

![Fig. 3.12: Snap shot of Audio Database](image)

3.11 Word Formation

Following sections narrates the outline to form the words from syllables and phonemes. It also explains the process of cutting the words from sentences, region list for specifying the boundaries of the words. CV structure of the words [12].

3.11.1 Deriving Syllable

In order to develop a syllable database, word cut into syllables and specific syllables cut into phonemes. All the syllables and phonemes store in audio files in the audio database. Search the syllables into the database & concatenate them [44]. Cutting of the word into the syllables must be very accurate.
With the help Sound Forge & Audacity software, words are recorded and cut into syllables. This software will automatically form the regions according to the settings & each region give the syllables present in the word. The length of the syllables & their start & end location which we require for concatenation is obtained by using `view region list` option.

Fig. 3.13 shows vowels and consonants in Marathi language and CV structure of a word.

![Fig. 3.13: Vowels - consonants and CV structure](image)

![Fig. 3.14: Syllables forming from a word](image)
Fig. 3.15: Setting for auto region for words

Fig. 3.16: Effect words/sentence on sound file

Fig. 3.17: Region list of words
3.11.2 Cutting of Sound wave file

Words are recorded with good voice talent. Word from each sentences manually cut by taking the help Audacity software and region list is formed by the sound forge software. As entire process of recording and cutting the words in to syllables can be done automatically with high end software’s, but due to limited version of software available, the words are manfully cut in to syllables. For syllables in to phonemes, syllables cut for specific examples [19]. Spectrogram study of the words, syllables & phonemes were performed with the help speech synthesizer software. The steps are described as follows:

Step1: Open the required .wav file in Audacity, part highlighted to be cut.

Fig.3.18: Snap shot of cutting of the sound wave
Step 2: Open file->export selection as .wav

Fig.3.19: Snap shot of cut dot wave file

Addition of Words and Syllables in database:

1. Add the new word/syllable to Char_word column of database.mdb.
2. Add the wav file to Char_name column of database.mdb.
3. Add the wave file to the folder “voice files”.

3.12 Summary

A study of the various speech synthesis system using words, syllabic synthesis units for generating various speeches like Malay, Mandarin Chinese, Punjabi, Arabic speech, Bengali, experiment of unit selection by Kishore Pralahad, experiments on Unit Size for Unit Selection Speech Synthesis for Indian languages gives in sight. All this work enabled and help appropriate synthesis method for Marathi TTS synthesizer as explained in this chapter. All the stages of the TTS conversion process have been investigated, and rules governing the production of speech from input text, conversion of the words to syllables & phonemes, generation of the synthesis units from the phonetic string, and concatenation of the synthesis units are implemented.

Algorithms are developed for the computer handling TTS synthesis for Marathi language. When tested with sample file using the data base created, speech synthesizer gives the results as estimated and desired.