Chapter 6
Final Working Model for Text to Speech Synthesis

In this chapter, the final working model for text to speech synthesis in Indian languages is presented by incorporating the proposed quality enhancement techniques discussed in Chapter 4 and Chapter 5 with the base model presented in Chapter 3. A comparative study with the existing syllable based technique is also presented to show the effectiveness of the proposed model in producing synthetic speech segments out of the text utterances.

6.1 Overview of the Proposed Model

The overview of the final model is shown in Figure 6.1. The basic 35 speech units recorded at 8000Hz mono (male voice) are used as speech data to produce the synthetic speech. The respective fraction duration proportions (in %) are evaluated by the proposed onset identification technique discussed in Chapter 5. At an initial step of implementation the model is trained with the Unicode equivalent of the three Indian languages, Odia, Hindi, and Bengali. For all other languages the transliterated text in Odia language is used.

The input to the model is text (words or sentences) entered at the runtime or from a text document. For each of the entered or read sentences from the text document, the text processing module performs a word level processing of the text units. i.e. the sentence is broken down to a number of words and individual words are considered and processed to produce the output speech. Even though, the details of the phases involved are discussed with examples in Chapter 3 for the base model, the
steps involved in text to speech conversion for the final model may be summarized as follows:

**Text Tokenization:** The input to the model is text in the considered languages and text tokens are created from the words using the stack based reverse text processing technique discussed in Chapter 4.

**Speech Unit Identification:** The respective speech units (vowels/consonants) for the identified tokens are identified.

**Database Mapping:** This step provides the respective speech sample in the database considered for the waveform concatenation process.

**Smooth Concatenation of Speech Units:** Depending on the type of text tokens obtained, as dependent or independent a smooth waveform concatenation process is performed to produce the desired output speech. While, for the independent type units the whole waveforms are concatenated for the dependent type of units the appropriate fraction duration proportions (in %) obtained for the consonant units (discussed in Chapter 5) are used to produce the C-M or C-F or C-H-C type units. To smooth the concatenation points and to avoid audible glitches, transition frames are created for the CM type units by using the proposed centered moving average technique discussed in Chapter 5 and concatenated between the two considered waveforms.
6.2 Algorithm

Algorithm 6.1, presents the fraction based waveform concatenation process to produce the synthetic speech segments from the entered text. Upon receiving the text inputs a word is considered at a time and the stack based reverse text processing technique is used to identify the text tokens involved in the pronunciation. The algorithm then identifies the speech units involved for each identified tokens and performs the fraction-based or whole waveform concatenation to produce the output speech segments. However, for the entered numerals, first the context of numeral pronunciation is identified based on the set of rules and the pronunciation model presented in Chapter 4. The identified pronounceable units are mapped to their respective character pronunciations and the same fraction based waveform concatenation algorithm is used to produce the synthetic speech segments.

<table>
<thead>
<tr>
<th>Algorithm 6.1</th>
<th>Fraction-based Waveform Concatenation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
<td>Unicode text</td>
</tr>
<tr>
<td><strong>Output:</strong></td>
<td>Speech file</td>
</tr>
<tr>
<td><strong>Steps:</strong></td>
<td></td>
</tr>
<tr>
<td>1:</td>
<td>Create text tokens for the input word.</td>
</tr>
<tr>
<td>2:</td>
<td>For each obtained text token identify the respective speech unit in the database.</td>
</tr>
<tr>
<td>3:</td>
<td>If considered token is an independent type of unit, perform whole waveform concatenation.</td>
</tr>
<tr>
<td>4:</td>
<td>If considered token is of dependent type, perform fraction-based waveform concatenation on the speech data considering the respective fraction duration proportion and add the transition frame at the concatenation point.</td>
</tr>
<tr>
<td>5:</td>
<td>Repeat step 2 to 4 for each text token.</td>
</tr>
</tbody>
</table>

6.3 Illustration

This section presents an example of how the input text in an Indian language is converted to the respective speech signal by the proposed methodology through different phases. Figure 6.2 shows the phases of text to speech conversion for an Odia
word “sabda”. The input word is first divided into text tokens by the text analysis (text tokenization) phase and the grapheme to phoneme conversion rules included for mapping the text units to the respective sound units are used to identify the desired speech unit in the speech database (speech unit identification). The respective speech units available in the database are identified for the text (database mapping) and the fraction based waveform concatenation technique (Algorithm 6.1) is applied on the speech samples to produce the output speech.

Figure 6.2: Test to speech conversion process for input text “sabda” in Odia language
6.4 Comparative Study with Syllable-based Technique

In this section, we made a comparison of the proposed model with the existing syllable-based speech synthesis technique with respect to the considered efficiency parameters: storage requirement, computation overhead, accuracy and overall performance.

6.4.1 Storage Requirement

As, the storage required for the TTS systems depends mostly on the speech database size, the size of the speech units and their number affects the total storage requirement. While, the unit selections technique needs to store thousands of recorded speech samples requiring a memory size in gigabytes, the diphone synthesis technique storing around 1000-2000 speech samples in a language requires a memory of more than 2 megabytes. The use of syllable units in the database of concatenative technique needs to store around 700-1000 speech samples in any language requiring a memory size of around 1-2 megabytes. In general, storing thousands of speech samples in the database directly affect the total storage required for the TTS system making it difficult to use for the small hand held devices with limited storage resources.

The syllable based technique achieves highly natural speech segments, however to minimize the storage requirement it may use .gsm compressed speech samples (Global System for Mobile audio files) in the speech database where the speech signal can be stored in a highly compressed (i.e. coded) form to use a large voice database even under tight memory limitations. Even though the compressed speech samples reduces the memory requirement to around half the total storage, an extra decompression step is required while producing the output speech. However, our approach requires very less number of speech samples in the database, 35 in total requiring a memory of 235 kilobytes(approximately) that does not require any compression and hence decompression. The total space/storage requirement ($S$) for the speech database may be determined using the standard relation [136] presented in equation 6.1, where $S_r$, $N_c$, $B_s$, and $t$ are sampling rate, number of channels, bits per sample and total time duration of the speech samples respectively, and $m$ being the total number of speech units stored in the database and $(S_i)$ be the storage consumption by the $i^{th}$
audio file in the database.

\[ S = \sum_{i=1}^{m} S_i \]  

(6.1)

Where \( S_i = S_r \times N_c \times (B_s/8) \times t_i \)

Therefore, for \( m \) speech units in the database, there may be two possible patterns \( T = \{t_1, t_2, t_3, \ldots t_m\} \) for the time instances and \( S = \{s_1, s_2, s_3, \ldots s_m\} \) for the size requirement with the relationship shown in equation 6.1. And for each obtained \( S_i \) the total storage consumption needed for the speech database may be obtained by finding the sum of all obtained \( S_i \). The speech samples stored in our database are sampled at a rate of 8000 Hz using 16 bits encoding (mono). The time instance in seconds represent the total time of the speech sample in seconds over the time axis in the respective wave representations. Considering the total storage and number of units in the speech database of syllable based technique to be 100%, the proposed model achieves approximately 96% reduction in speech samples with a memory gain of 81% as shown in Figure 6.3. The small database size may make it useful for applications in small hand held devices with limited storage resources.

![Figure 6.3: proportion of total number of units (left) and storage space consumed (right) by proposed model and existing technique](image)

### 6.4.2 Computation Overhead

To analyze the performance of the proposed technique in terms of execution time, different text files containing words of \( D_c \) and \( I_c \) type units are prepared where the unit type considered are CC, CV, VC, VV, CM, CF, CH-CM and CHCF/CH-CHC. The average execution time for different experiments is obtained by generating output speech with increasing number of words. The same set of test data are generated by the syllable based technique with increasing number of words.
Figure 6.4 and Figure 6.5 show the average execution times for the $D_c$ and $I_c$ type units in the three considered languages by both techniques in milliseconds respectively with respect to increasing number of words in the text documents. Considering the average execution time of syllable based technique to be 100%, the proposed model achieves 92.5% reduction in computation overhead in an average for all the performed experiments.

![Graphs showing average execution time](image)

Figure 6.4: Average execution time of three languages for syllable based technique and proposed technique with respect to increasing number of words for $D_c$ type of units

The syllable-based technique uses the .gsm compressed speech samples in its database to minimize the storage requirements. Therefore, an extra decompression step is needed before each unit concatenation along with the speech unit identification step. Considering an input word $W$ with $N$ number of syllable units and assuming that the best decoding method is used for decompressing the speech samples, the amount of time taken to decompress $N$ units is $O(N\log N)$ in the syllable based technique. Also, the concatenation process for $N$ syllable units takes $O(N^3)$ time (due to the text to phonetic representation conversion, appropriate speech unit identification and decompression needed before concatenation). Therefore, the execution time taken by the syllable-based technique for producing the desired output speech is $O(N^4\log N)$. i.e. quadratic time of 4th order with increasing number of syllable units in the words.
Chapter 6: Final Working Model for Text to Speech Synthesis

Figure 6.5: Average execution time of three languages for syllable based technique and proposed technique with respect to increasing number of words for $I_c$ type of units

However, as the speech samples in our model are stored without any compressions, therefore our approach does not require any further decompressions reducing the execution time by $O(n \log n)$, where $n$ is the number of segments present in the considered word ($W$). Also, the use of Stack data structure for text processing which requires constant execution time reduces the overall execution time of the proposed model. The amount of time required by the proposed model to produce the desired output speech is $O(n^2)$. i.e. quadratic time of 2nd order, where the simple centered average technique used for smoothening concatenation points requires linear time.

Even though, $n \geq N$, the proposed algorithm takes less than half the execution time of the syllable-based technique to produce the desired speech output, because of the simple adapted techniques requiring less computation time. Where as the existing syllable-based technique needing the compressed speech samples to minimize the storage requires extra decompression steps at each unit concatenation along with extra mapping functions to identify the actual unit type available in the database containing different combination of units which increases the overall computation overhead of the technique to produce the synthetic speech segments.
6.5 Overall Performance Analysis

To evaluate the overall performance of the proposed model, the overall accuracy, adaptation in a new language and adaptation in a new speech corpus issues are considered.

6.5.1 Accuracy

To obtain the overall accuracy of the model in producing synthetic speech segments, the model is tested on different type of text documents in the three considered languages. The details of the text unit types considered are presented in Table 6.1. A set of example words for testing in the three considered languages is presented in the appendix section. In all the experiments performed, the model is capable of producing uninterrupted speech segments for different text unit types in the documents.

<table>
<thead>
<tr>
<th>Type of documents</th>
<th>Story books, novels, poetry books, mythological documents, newspaper text, study books</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit types</td>
<td>C-C, V-V, C-V, C-M, C-F, C-H-C, C-H-C-M, C-H-C-F, C-H-C-H-C</td>
</tr>
<tr>
<td>Languages considered</td>
<td>Hindi, Odia, Bengali</td>
</tr>
<tr>
<td>Number of words per test</td>
<td>100</td>
</tr>
</tbody>
</table>

6.5.2 Adaptation in a New Language

To test the effectiveness of the proposed technique in producing speech units for a different language, a set of text documents are prepared containing text for 100 different words in three languages, Telugu, Panjabi and Indian English in transliterated format of Odia language. i.e. the pronunciation of the words are written in transliterated format of Odia language. The proposed technique is then applied to all the transliterated text iteratively language wise. In all the iterations, the proposed TTS technique correctly produces all the required speech samples for the set of considered text units. The details of different parameters considered for the tests are presented in Table 6.2. The output speech for 10 random words in the considered languages
are selected and a group of 5 native speakers in the respective languages are asked to give their feedback on the output speech quality with respect to understandability (language accent is not considered) in a 5 point scale (1: very low, 2: low, 3: Average, 4: high, 5: very high). The model achieves a MOS score of 3.9 out of 5 in an average in all the tests performed showing the efficiency of the proposed technique in producing understandable speech in a new language. And adapting a new language does not require addition of new speech samples in that language to the speech database.

Table 6.2: Test details for adaptation in other languages

<table>
<thead>
<tr>
<th>Languages considered</th>
<th>Telugu, Panjabi, Indian English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of words per test</td>
<td>100</td>
</tr>
<tr>
<td>Total number of words</td>
<td>1000 per language</td>
</tr>
<tr>
<td>Transliteration Language</td>
<td>Odia</td>
</tr>
<tr>
<td>Type of documents</td>
<td>mythological documents, story books, newspaper text</td>
</tr>
<tr>
<td>Speech quality parameter</td>
<td>Understandability</td>
</tr>
<tr>
<td>Language ascent considered</td>
<td>No</td>
</tr>
</tbody>
</table>

6.5.3 Adaptation in Different Speech Corpus

To show the efficiency of the technique in producing different types of speech units by the fraction-based concatenation process, the fraction duration based technique is tested on three different speech corpuses. The speech data in the database are prepared by segmenting the required 35 basic speech units from a set of recorded audio files in both adult male and female voice. All the speech samples are converted to a generic 8000Hz, wave audio format (.wav) files without adapting any compression technique.

A set of experiments are then performed to produce different type of units from the new speech database. The generated output speech are evaluated by a set of listeners based on the understandability score in the same 5 point scale in the three considered languages for the $D_c$ type units based on the feelings of the listeners to the output speech. For this purpose, a set of ten random words from each language are considered. The average results for different type of units for three different
speech corpus (corpus1: male-segmented, corpus2: female segmented, corpus3: female recorded) are presented in Figure 6.6 for the three considered languages. The MOS results show the effectiveness of the proposed techniques even on a new speech database with short variation in the MOS scores.

![Figure 6.6: MOS test results for different type of speech units in the three considered languages on different speech corpus](image)

### 6.6 Comparative Study with Naturally Generated Speech

In this section, we made a comparison of the artificially produced speech with the naturally spoken speech based on different speech parameters: Energy, magnitude and auto correlation. As speech is produced from a time varying vocal tract system with time varying excitation the signal is non-stationary in nature. i.e. it changes over time to time. It is because different speakers have different voices and produce different type of sounds. However, as the proposed work focuses on a waveform concatenation technique it retains the speech quality of the respective speaker. We made a comparison of the fraction based technique generated speech with the naturally generated speech for the same speech unit with respect to the variation in their different speech parameters. The relation for finding the short term energy can be derived from the total energy relation defined in signal processing [136]. The total energy ($E_T$) of an energy signal ($S_g$) with n samples is given by:

$$E_T = \sum_{n=-\infty}^{\infty} S_g^2(n)$$

(6.2)
6.6.1 Time Domain Representation and Energy Analysis

The time domain representation and the energy plot of the speech segments generated naturally and artificially are shown in Figure 6.7 and Figure 6.8 respectively for the CM type unit “kaa”. Assuming the non stationary speech signal to be stationary for a very short period of time, the similarity of energy level of the generated speech and the naturally recorded speech may be observed from the graph presented in Figure 6.9 where a 20ms and 30ms window is considered for short term energy computation. The results show the high degree of similarity at energy level of generated speech with the naturally generated speech for the considered window sizes.

![Figure 6.7: Time domain representation of naturally recorded (left) and artificially generated (right) speech for CM type unit “kaa”](image1)

![Figure 6.8: Energy plot of naturally recorded (left) and artificially generated (right) speech for CM type unit “kaa”](image2)

The short term energy and spectral similarity in word formation is also analyzed for the naturally spoken words and generated speech. Figure 6.10 shows the time domain representation of the naturally recorded speech and fraction based generated
Figure 6.9: Short term energy comparison with varying window length 20ms (left) and 30ms (right) for naturally recorded and artificially generated speech for the CM type unit “kaa” in Odia language

speech for the word “sabda” in Odia language. The energy plot of both speech segments are shown in Figure 6.11 and Figure 6.12 shows the short term energy analysis with respect to 20ms and 30ms window length showing high degree of similarity at energy level used for speech production.

Figure 6.10: Time domain representation of naturally recorded (left) and artificially generated (right) speech for the input word “sabda” in Odia language
Figure 6.11: Energy plot of naturally recorded (left) and artificially generated (right) speech for the input word “sabda” in Odia language.

Figure 6.12: Short term energy comparison with varying window length 20ms (left) and 30ms (middle) for naturally recorded and artificially generated speech for the input word “sabda” in Odia language.

6.6.2 Magnitude Analysis

While the short term time domain analysis is useful for computing the time domain features at the gross level like energy, the different frequency or spectral components that are present in the speech signal are not directly apparent in the time domain. Therefore, the frequency domain representation using Fourier representation are used. The similarity of spectral components of both the speech units are presented in the magnitude plot presented in Figure 6.13 for the CM type unit and in Figure 6.14 for the production of the Odia word “sabda”. The magnitude spectrum is for the entire speech signal and gives the gross information of the frequency components present in the generated and naturally recorded speech signal. The short term magnitude analysis for the units “kaa” and “sabda” are presented in Figure 6.15 showing high degree of similarity of the spectral values overlapping on each other.
Figure 6.13: Magnitude plot for naturally recorded (left) and artificially generated (right) speech for CM type unit “kaa”

Figure 6.14: Magnitude plot for naturally recorded (left) and artificially generated (right) speech for the input word “sabda” in Odia language

Figure 6.15: Short term energy comparison for “kaa” (left) and “sabda” (right) for naturally generated and synthesized speech with window size 20ms

6.6.3 Auto-correlation Analysis

Figure 6.16 shows the energy contours for speech signal taken for study and the correlation coefficient that shows the auto co relation among the data values for the
natural and generated speech. While the auto correlation values shows high degree of similarity, as the whole non stationary signal is considered for short term energy comparison a variation may be noticed in the presented graph. However, considering the non stationary speech signals as stationary for a period of 10-30 ms, the short term energy comparison is presented in Figure 6.17 for a window of 20 and 30ms.

Figure 6.16: Short term energy plot (left) and auto correlation (right) for naturally recorded and artificially generated speech for the input word “sabda” in Odia language

Figure 6.17: Short term energy plot with varying window length 20ms (left), 30ms(right) for naturally recorded and artificially generated speech for the input word “sabda” in Odia language

The results show very high degree of similarity at energy levels of the naturally recorded and generated speech for short windows compared to the whole wave data. The results of all our experiments shows very close resemblance of the generated speech with the naturally recorded speech showing the effectiveness of the proposed technique in achieving similar speech segments with the naturally recorded speech by the fraction-based waveform concatenation technique.
6.7 Subjective Evaluation for Understandability

To analyze the output speech produced by the proposed model compared to the syllable-based technique, a subjective evaluation test is performed. We use the most common approach adopted by the researchers. i.e. the Mean Opinion Score (MOS) test [30]. In these tests, the MOS for the set of test data prepared in the three considered languages are evaluated by a group of listeners. To perform these experiments, three groups of respective language native speakers (20 listeners) are considered without any knowledge on the TTS technology. The test focuses on comparing the performance of the technique for the \( D_c \) type units only where a fraction based waveform concatenation is performed.

The concatenation process of \( I_c \) type units of the form CC, CV, VC and VV shows similar results as direct waveform concatenation is considered in both the speech synthesis techniques. A set of test data in Odia, Hindi and Bengali language is prepared with CV, VC, CM, CF, CH-C, CH-CM, CH-CF, CH-CH-C units and the respective output speech is generated by the proposed technique and by the syllable based technique. The listeners of the respective languages are asked to rate the output speech in a 5 point scale (1: very low, 2: low, 3: average, 4: high, 5: extremely high) for ten different words from each type. The average MOS for different types of words by the listeners are presented in Figure 6.18. The results obtained in all the experiments shows the proposed model achieves relatively better or comparable results compared to the existing technique on the test data considered in the three Indian languages even with a very small speech database.
In this work, we proposed a fraction-based waveform concatenative technique for text to speech synthesis in Indian languages. The major objective of our work is to produce highly intelligible and natural sounding speech with less storage and computation overhead for its usability in human-computer interactive systems. The model provides a cross section between different issues in the concatenative technique and implemented some techniques to achieve the required objectives. The use of the fraction duration evaluation technique for waveform concatenation provides better quality results compared to the static fractions used for waveform concatenation.

To analyze the performance of the proposed model compared to the existing technique, a subjective evaluation test is performed by comparing the synthesized output speech with the syllable-based technique generated speech in the three Indian lan-
Table 6.3: Performance comparison of proposed technique with other techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Domain</th>
<th>Diaphone</th>
<th>Unit</th>
<th>Syllable</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency on Speech Database</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>New sound Production capability</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Need of new speech samples for a new language</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Storage Requirement</td>
<td>Very high</td>
<td>Very high</td>
<td>Very high</td>
<td>High</td>
<td>Very low</td>
</tr>
<tr>
<td>Execution Time</td>
<td>Average</td>
<td>Average</td>
<td>High</td>
<td>Average</td>
<td>Very low</td>
</tr>
</tbody>
</table>

languages, Odia, Hindi and Bengali. The subjective analysis shows the effectiveness of the proposed technique in producing intelligible speech in different Indian languages. Also, the proposed model is evaluated on the basis of storage requirement and computation overhead to show the effectiveness of the proposed waveform concatenation technique compared to the existing technique. The summary of our findings are given in Table 6.3.

Even though, the model provides remarkable results compared to the existing technique with less storage and computation overhead, the output speech still presents gap while compared to the naturally recorded speech. Therefore, the model may further be enhanced to smoothen the concatenation points by implementing some smoothening techniques to achieve much better results. Also, some prosodic modeling and intonation techniques may be incorporated in to the model to produce more natural speech segments.