Thirukkural II - A Text-to-Speech Synthesis System

P. Prathibha, A. G. Ramakrishnan, R. Muralishankar
Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, INDIA

1. Introduction

In this paper, we are reporting the development of an improved version of Thirukkural, the Tamil synthesis system presented in Tamilnet 2001. The improvements when compared with Thirukkural I [6] are limited ability to handle borrowed words and proper nouns, reading of digits, better naturalness due to incorporation of pitch modification and emulation of voices of old and young people.

2. Pronunciation of words of foreign origin in Tamil

Thirukkural I is the basic Tamil speech synthesis system which can pronounce only pure Tamil words. On the other hand, Thirukkural II handles pronunciation of words of foreign origin.

One cannot guarantee 100% performance for products such as, speech synthesis and OCR. When the question arises as to whether Speech Synthesis system is standardized, we must have an idea of what constitutes a general definition of standard language, or failing that, what constitutes standardization in a particular language? We have evidence in many languages of both conscious, planned standardization (via language academies, dictionary-writers, printers and proof-readers) and have the somewhat haphazard choice of a particular dialect of some city or ruler (Bangalore, Gajendran). ``In the strictest sense, no spoken language can ever be fully standardized." Writing and spelling are easily standardized; spoken standardization is an ``ideology", an idea, not a reality. New ways probably need to be devised to broaden the concept of standardization, to allow for variation, perhaps in register and domain, without giving up the whole notion of having a form of language of widest communication, or the utility of some kinds of agreed-upon understandings. Too often, standard grammars are in fact norms for written language, but this gets forgotten when spoken language is taught, as it is today. Computerization alone will demand various things; one needs to try our synthesis (which incorporates grammar) and see if one agrees with the kinds of decisions it makes about one's usage. Though spoken Tamil may not be completely standardized, i.e. there are areas of variability, it is in a position where standardization could in fact be brought about. That is, the potential for standardizing the language is there, and if certain conditions were met, the process could be complete.

Due to the non-standardization of the language, Tamil Speech synthesis system cannot perform adequately well, if it strictly follows the Tamil pronunciation rules. We need to come out with some strategies to code the characters (more exactly, phonemes) of other languages, which are likely to be used frequently. This shall facilitate synthesis of proper nouns, such as, names of people, rivers, mountains, places, common words of English and other languages, such as sanskrit, that we use. For example, based on the proper application of Tamil phonetic rules, one cannot generate words such as, Ganga or Sita. These are synthesized as Kanga and Seedha, respectively.

A newer, more open way for synthesizing words of foreign origin is to be agreed upon, allowing for flexibility and opening the system to synthesize foreign words, rather than only words of Tamil origin. Accordingly, we proposed the use of certain modifier symbols [1], which can modify any hard consonant into its soft version. We have used '~' as the modifier symbol in our current implementation. This solves problems in synthesizing words such as, Dhanya, Gajendran, and Bangalore by writing them as ~Thanya, ~K ajendran and ~Pankalore.

For example:

(1) தந்ய ~ தந்ய (Dhanya)
(2) கஜேந்த்ரந் ~ கஜேந்த்ரந் (Gajendran)
(3) பெங்கலுர ~ பெங்கலுர் (Bangalore)

However, this will not solve the problem of hard consonants being wrongly converted into soft consonants because of the rules. For example, Aakash is synthesized as Aagash due to the application of Tamil phonetic rule. Again we have used the same modifier symbol '~', which will nullify the standard phonetic rule of Tamil. Thus, whenever hard consonant is preceded and followed by vowels, if the modifier '~' is present, then the phonetic rule is not applied. Hence it remains as a hard consonant.

For example:

(1) ஆகாஷ் ஆ~காஷ் (Aakash)
(2) சீதா சீ~தா (Siitha)
(3) கோபால ~கோ~பால (Goopala)

Hence, by the use of elegant modifier symbol, Thirukkural II can synthesize words of foreign origin. We suggest effective standardization of the pronunciation of proper nouns and if that is used universally, it shall definitely help in standardization of spoken Tamil.

3. Text Normalisation

At first sight, the process of converting text into speech looks straightforward. However, when we analyze how complicated speakers read a text aloud, this simplest view quickly falls apart. The ultimate goal of simulating speech understanding is highly challenging because humans depend on common sense reasoning about the world and text's relation to it and knowledge of the language itself in all its richness and variability and so on. While computational power is steadily increasing, there remains a substantial gap that must be closed before fully human-sounding simulated voices can be created. The text analysis component is typically responsible for conversion of non-orthographic symbols and parsing of input text. The processes related to text analysis include text normalization and parsing of input text after application of grammar rules. Thirukkural II does text normalization during text analysis that can detect the non-orthographic symbol.

Text Normalisation is the process of generating normalised unambiguous representation from text containing words, numbers, punctuation and other symbols. Any text often includes digits, which, may be part number, stock number, date, time, currency or any mathematical expression. Without context analysis or prior knowledge, even a human reader would sometimes be hard pressed to give a perfect rendition of every sequence of non-alphabetical character or any abbreviation. For example,

The rate of interest is 10 percent THE RATE OF INTEREST IS TEN PERCENT

Text Analysis for TTS does the work of converting such text into the format that can be recognised by our synthesis system. Text Normalisation includes two phases: identification of type and expansion to unambiguous representation. The algorithm for text normalisation is given in Table 1. First the input text is processed to identify the type of sequence. Once the sequence is identified, it is passed on to a function that expands the sequence into representation that is easily recognised by the system. Table 2 shows the example of input with relaxed unambiguous output.

Table 1

```
Algorithm for Text Normalisation:
 1. Identification
        If a match is found go to 2     Else Go to 3
 2. Expansion
        Insert the expanded sequence corresponding to the match.
 3. Advance
         Move one character right and go to 1.     If end of text, finish
```

Table: 2

Input	Output
10	பத்து
2000	இரண்டாயிரம்
150000	ஒன்ரு லச்சத்தி ஐம்பது ஆயிரம்
10000000	ஒன்ரு கோடி

4. Improving naturalness in speech

Natural sounding speech is speech that allows the listener to attribute this voice to some pseudo-speaker and to perceive some kind of expressivity as well as some indices characterizing the speaking style and the particular situation of elocution. Naturalness is associated with many features like voice quality, prosody, intelligibility, co-articulator coherence and presence of acoustic processing artefacts.

The current generation of concatenative speech synthesis systems rely on the selection of appropriate pre-recorded speech units from a repository of sounds. This process, commonly referred to as unit selection, is a critical step in the production of natural sounding speech. However the process is only as good as the annotation and initial recording of the underlying database. These units, once selected, must be seamlessly concatenated and prosodically modified to reflect the desired rhythm and intonation. The individual speaking style of the speaker and the basic unit used in the recording both contribute significantly to the overall naturalness of the system.

For synthesis of natural-sounding speech, it is essential to control prosody, to ensure appropriate rhythm, tempo, accent, intonation and stress. Improving naturalness is achieved in two steps: Text interpretation and Prosody modification.

4.1 Text Interpretation

In normal writing, sentence boundaries are often signalled by terminal punctuation from the set: full stop, exclamation mark, question mark or comma {. ! ? ,} followed by white spaces. In reading a long sentence, speakers will normally break up the sentence into several phrases, each of which can be said to stand alone as an intonation unit. If punctuation is used liberally so that there are relatively few words between the commas, semicolons or periods, then a reasonable guess at an appropriate phrasing would be simply to break the sentence at the punctuation marks though this is not always appropriate. Hence determining the sentence break and naming the type of sentence has to be done so as to apply the prosodic rules. A simple algorithm for sentence breaking and naming the type is incorporated: The input text is scanned for the above set of punctuation and once it is found, the sentence boundary and type is labelled during parsing of input text. This process is continued till the end of the text.

4.2 Prosody modification

From the listener's point of view, prosody consists of systematic perception and recovery of speaker's intention based on (1) Pitch: Fundamental frequency (fo) as a function of time, (2) Pauses: To indicate phrases and (3) Loudness: Relative amplitude/volume

Pitch is the most expressive of the prosodic phenomena. As we speak, we systematically vary our fundamental frequency to express our feelings about what we are saying. If a paragraph is spoken on a constant, uniform pitch without pauses or uniform pauses between words, it sounds highly unnatural. Prosodic rules [3] differ for sentences like affirmative, interrogative or exclamatory. For example: 1. Rise in pitch on the last syllable of a yes-no question as shown in Fig. 1. 2. Drop in pitch on the last syllable of an affirmative sentence as shown in Fig. 2. 3. Extreme rise in pitch on the last syllable of an exclamatory sentence as shown in Fig. 3.

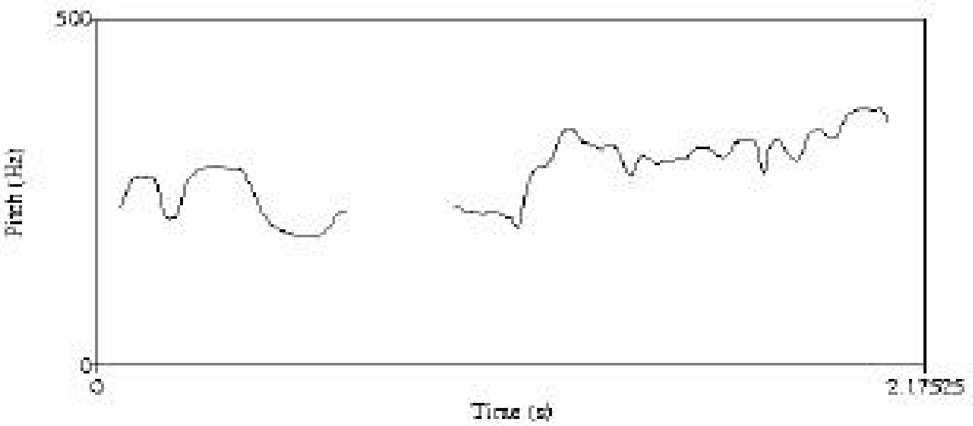

Figure (1): Pitch contour of a yes-no question
(vitiyai matiyAl vella muTiyuma?)

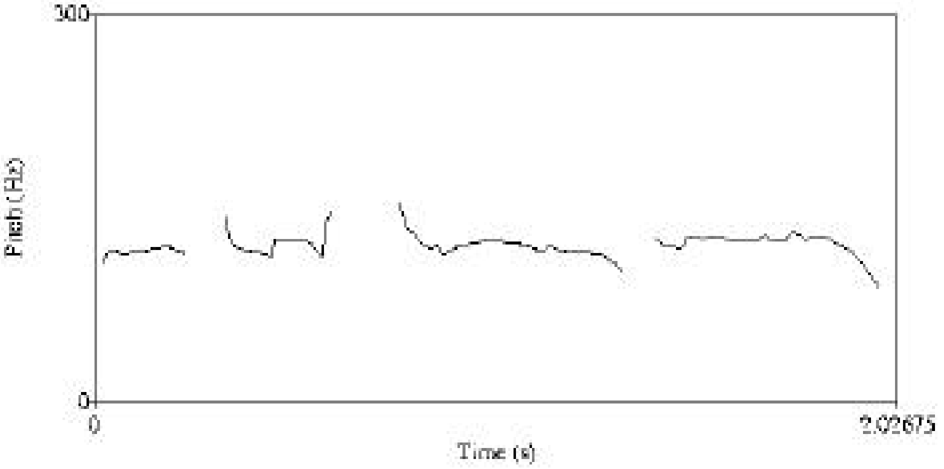

Figure 2. Pitch contour of an affirmative sentence
(AkAsh nalla paiyan)

Time varying Pitch modification-using Discrete Cosine Transform (DCT) [8] has been used here to raise or lower the pitch contour of the segments before concatenation, so as to generate different types of sentences. The linear prediction (LP) residual is obtained from pitch synchronous frames by inverse filtering the speech signal. Then the DCT of the residual frames is taken. Based on the desired factor of pitch modification, the dimension of DCT of the residual is modified by truncating or zeros padding, and then the Inverse DCT is obtained. This period- modified residual signal is then forward filtered to obtain the pitch modified speech. With the pitch-marking algorithm, we can add or subtract the number of cycles required to suit the duration of the segment. Thus our system was able to generate different types of sentences by varying the pitch according to the prosodic rules.

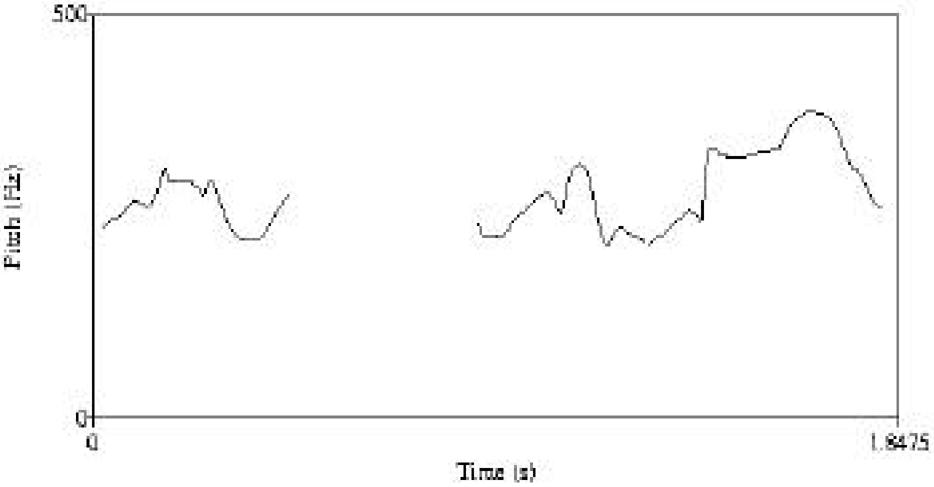

Figure (3): Pitch contour of an exclamatory sentence
(avan ingku vantAnA?)

In natural speech, speakers normally and naturally give pauses between sentences. The average duration of pauses in a natural speech has been observed and a look up table (Table 3) is generated. Finally, the lookup table is made use of to insert pauses between sentences that improve naturalness.

Table 3

Sentence Type	Duration in seconds
Affirmative (.)	1
Exclamatory (!)	0.9
Question (?)	0.8
Comma (,)	0.5

5. Voice transformation

Voice transformation is the process of transforming one or more features of an input signal to target values. By features, we mean fundamental frequency of voicing, duration, energy and formant positions. The reconstructed signal should be of high quality and without artefacts due to signal processing. There are many potential applications of this technique in concatenative speech synthesis. The method can be applied to transform the speech corpus to different voice characteristics like female, child and old man.

In order to design a speech system with multiple voices, it is highly impractical to collect data from multiple speakers. To deal with the problem of generating multiple voices, a voice transformation has been implemented to generate speech that sounds distinctly different from the voice of the input speaker. The relevant input units are pre-processed through a transformation phase that can change a male voice to female-like or child-like voice, while preserving the temporal characteristics. In this way, the database from the single speaker is thus leveraged to yield an apparent multiple speaker synthesis system.

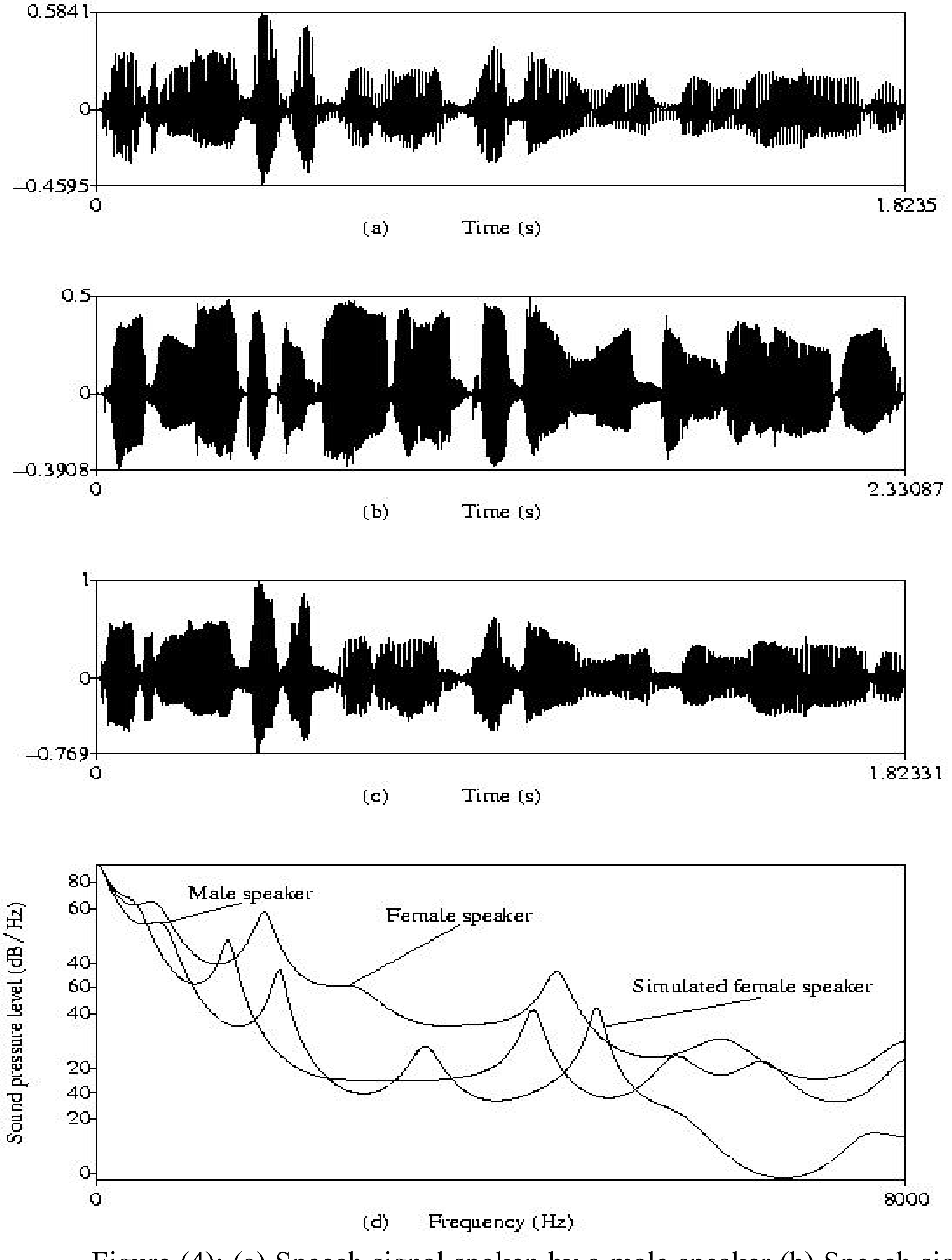

Figure (4): (a) Speech signal spoken by a male speaker (b) Speech signal spoken by a female speaker (c) Simulated female speech signal after modifying speech signal spoken by the male (d) LPC spectrum of speech signal spoken by male, female and simulated female speaker.

The transformation system can be utilised to alter the fundamental frequency i.e., pitch using a pitch modification algorithm and shifting of formant frequency. The pitch is modified using DCT [8] based pitch modification algorithm. The pitch modification factors required to convert a male voice into a female voice is around 1.4 to 1.6, and into a child voice is from

1.7 to 2 and into an old man's voice is from 0.5 to 0.7. Resampling the magnitude spectrum alters the apparent positions of formant. For example, if we interpolate the spectral envelope by a factor of 1.2 and discard the extra points at the upper end, the formants will be moved up by roughly 20 percent. Similarly, if we decimate the spectral envelope, the formants will be moved down. Thus by making the fundamental frequency high by 30% and shifting the formants up by 25%, we could convert male voice to female voice. Figure 4 shows that after the above mentioned conversion of pitch and formant positions, the modified male speech is similar to that of a natural female speaker. Figure 4(a) shows an utterance from a male speaker. Figure 4(b) is the same utterance from a female speaker. Figure 4(c) is the signal simulated by making the fundamental frequency of the male utterance high by 30% and shifting the formants up by 25%. Figure 4(d) shows the formant positions of male, female and simulated female speaker. It can be observed that the formant positions of simulated female speaker match roughly with those of the female speaker.

6. Conclusion

Thirukkural II generates intelligible and acceptably natural speech. It also has the facility to produce different voices like female, child and old man. It can synthesize many of the proper nouns derived from certain other languages. It can also read digits present in the text. The synthesized speech is rendered natural by incorporating prosodic rules. Currently, we are attempting to synthesize emotions such as, sadness, anger and joy.

References:

[1] A. G. Ramakrishnan, "Issues in standardization for Text to Speech in Tamil", Tamilnet2001, Kualalumpur, Malaysia.

[2] Douglas O'Shaughnessy, Speech Communication - Human and Machine, Second Edition, IEEE press, 2000.

[3] R Muralishankar and A G Ramakrishnan, "Human Touch to Tamil Speech Synthesizer ", Tamilnet2001, Kualalumpur, Malaysia, pp. 103 - 109, 2001.

[4] R. Muralishankar et al. "DCT based Pitch Modification", Sixth Biennial Conference on Signal Processing and Communication SPCOM'01, IISc, Bangalore, pp. 114 - 117, July 15-18, 2001.

[5] R. Muralishankar and A G Ramakrishnan, "Robust Pitch detection using DCT based Spectral Autocorrelation", Proc. Intern. Conf. on Multimedia Processing, Chennai, pp. 129-132, 2000.

[6] G. L. Jayavardhana Rama, A. G. Ramakrishnan, V. Vijay Venkatesh, and R. Muralishankar, "Thirukkural: a text-to-speech synthesis system", Proc. Tamil Internet 2001, Kuala Lumpur, pp. 92-97, August 26-28, 2001.

[7] Min Tang, Chao Wang, Stephanie Seneff, "Voice Transformations: From Speech Synthesis to Mamalian Vocalizations", Conference on Speech Communication and Technology, Denmark, 2001.

[8] R. Muralishankar, A. G. Ramakrishnan and Prathibha P, "Dynamic Pitch changes for concatenative synthesis", SPPRA, Greece, 2002.