A Pragmatic Solution to an Indian Accented English Speech Synthesizer Using Residual Excited Linear Predictive Coded Voice

Sachin Agarwal Pallavi Agarwal

Indian Institute of Information Technology- Allahabad
Jhalwa, Allahabad – India
{sachinagarwal,pallaviagarwal}@ug.iiita.ac.in

Abstract. This paper elucidates a practical solution to an Indian accented English text to speech synthesizing system. The paper covers the complete procedure to generate the speech signal of the text, in Indian accented voice. The technique described considers the various prosodic features that need to be incorporated into the synthesized speech to make it appear natural and in the way an Indian speaks. The paper describes the complete method for synthesizing the speech including the pre-processing of the text and prosody analysis in the Indian style of speaking. The diphones extracted from an Indian's voice are coded using Residual Excited Linear Predictive (RELP) coding technique and the resulting formants and residual values of the diphones are used to resynthesize the final output speech for the text to be synthesized. 'PALSA' – an Indian accented English speech synthesizer has been successfully implemented using the mentioned technique to produce the Indian accented speech and is described in this paper.

1. Introduction

1.1 Motivation

The English language, being the lingua franca, is spoken in diversified accents in different parts of the world. Therefore different regions in the world use different styles of speaking the same English language. Same is the case with the Indian sub–continent. Various Text to Speech (TTS) synthesis systems available today lay stress mainly on the American or British accent, like freeTTS [1], Festival [2] or even the Windows® TTS system, but it is difficult for the people in non-English speaking nations to perceive such accent as they are not familiar with it and find it uncomfortable to understand the pure American or British accent. The Indian national language is Hindi and it influences the style of speaking English. Thus Indians are used to that speaking style, accent and pronunciation. For example, in Indian style of speaking phonemes for "Welcome" are "w ae l k ax m" while in American style, the phonemes would be "w eh l k ax m" according to the conventions used in Carnegie Mellon Pronouncing Dictionary (CMPD) [3]. Apart from the differences in phonemes

Bhanu Prasad (Editor): IICAI-05, pp. 458-472, 2005.
Copyright © IICAI 2005

there are subtle but important differences in the rules for prosody, which further differentiate the Indian accent from British or American accent.

1.2 The Human speech system

Human beings produce different sounds with the different configurations for the tongue lips and vocal tract. Speech production in human beings is due to the combined efforts of lungs, glottis with vocal cords and articulation tract, which includes mouth and nose cavity. The sounds produced can be categorized in to two broad categories: voiced and unvoiced.

The production of voiced *speech* takes place when the vocal cords vibrate due to the air pressure produced by the lungs and produces a quasiperiodic pressure wave. The pressure impulses produced are also called pitch impulses and the frequency of the pressure signal is known as the pitch frequency or fundamental frequency. The pressure waves produced stimulate the vocal tract and for some values of the fundamental frequency, the resonance occurs with the natural frequency of the vocal tract. When the cavities resonate, they radiate a sound wave which is the speech signal and the frequencies are known as formant frequencies. The constrictions are produced by the tongue and lips movements which change the natural frequency of the vocal tract and thus different sounds can be produced. The production of the unvoiced speech is not due to the regular vibration but because of the turbulent airflow due to a constriction in the vocal tract.

1.3 Organization of the paper

The rest of the paper has been organized as follows. The related work in this field has been mentioned in section 2. In section 3 we present the analysis of Indian and American accented speech. Although the analysis shows the differences between American and Indian accented speech, there are similar differences between the Indian and other English accents throughout the world. Section 4 gives an overview of the architecture of the system and the detailed description of all the modules is given in section 5. The result section(section 6) presents PalSa - an implementation of the Indian accented speech synthesis technique described in the paper. The conclusion of the research, followed by acknowledgements are in sections 7 and 8 respectively.

2. Related Work

Text to speech systems have been a boon to our society, especially in the field of web education and for the visually handicapped people. A lot of text to speech systems, both commercial as well as free, are available today.

FestVox[5], the project aims to build of new synthetic voices and it has been used in many Text to Speech systems like FreeTTS[1] , Festival[2] and Flite[4] to build the synthetic voices.

Festival Speech System[2] provides general framework for building speech synthesis systems. It uses Festvox for building new voices. Festival is multi-lingual (currently English (British and American), and Spanish) though English is the most advanced.

Flite [4] is a small run-time speech synthesis engine developed at Carnegie Mellon University. It is designed for embedded systems like PDAs as well large server installation which are used to serve synthesis to many clients. Flite is derived from the Festival Speech Synthesis System [2] from the University of Edinburgh and the FestVox project [5] from Carnegie Mellon University.

FreeTTS [1] is an open source text to speech synthesizer implemented in Java™ programming language. It is based on Flite.

All the above systems are developed for non-Indian accented speech. The incompetence of the Indians with American accented speech systems has motivated many research fellows to contribute their efforts for Indian accented speech systems. A rule based system to generate non-native pronunciation of English has been developed for Indian accented pronunciation of English as described in [15].

Although most of the systems support the multi-lingual text to speech synthesis but none of them considers the Indian accented way of speaking English.

3. Analysis of Indian and American accented speech

There is a subtle difference between the Indian and American speech. The study of recorded speech signals show that there are two main differences between the speech signals of an Indian and an American, these are:

1. Pronunciation of words.
2. Prosody with which the words are spoken.

Here we present an abstract of our extensive analysis of speech signals produced by Americans and Indians.

The waveforms below shows the speech signal of word “dye” spoken in American and Indian accent. The basic sound units (phoneme) for the word “dye” in American accent are ‘d ay’ and in Indian accent are ‘d aa iy’. (the conventions used are same as the Carnegie Mellon Pronunciation Dictionary [3]).

The vertical red marks show the transition boundaries between the phonemes. The difference between the American and Indian accent is clear from figure 1. The first phoneme being same in both the accents, thus the waveforms are similar. The difference starts creeping in the waveforms with the dissimilarity in phonemes.

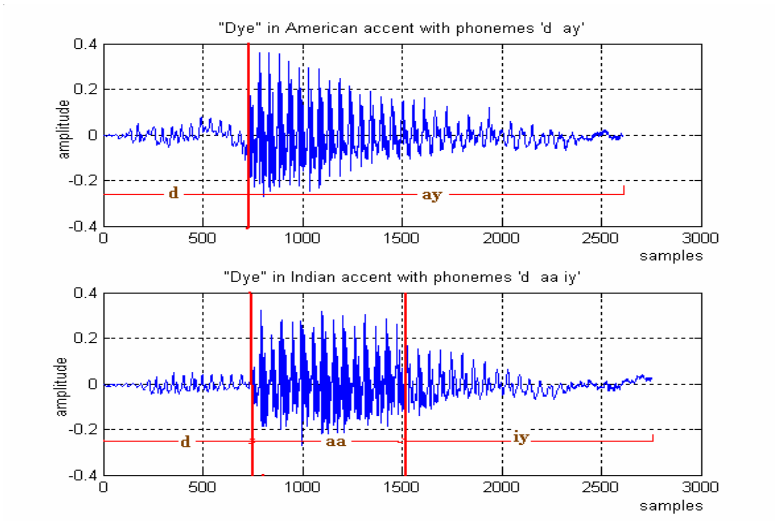

Fig. 1. Speech signal waveforms for the word "Dye" in (i) American and (ii) Indian accent.

The above example shows the difference in a word spoken in Indian and American accent. The difference is more perceptive when a whole sentence is analyzed.

In the sentence "I am a boy", the phonemes for the corresponding words in American accent are : I – ay, am -- ae m , a-ax , boy -- b oy while in Indian accent are : I--aa iy, am – ae m, a-ax , boy- b oy.

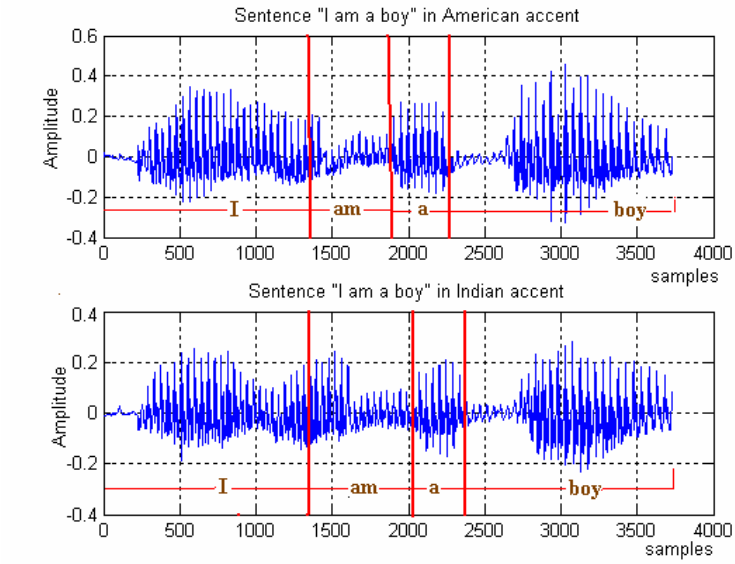

Fig. 2. Speech signal waveforms for the sentence "I am a boy" in (i) American and (ii) Indian accent.

We noticed that although the phoneme difference is only present in word “I”, even then the two waveform differs substantially. From the waveforms, the difference in pronunciation of “I” can be easily figured out. The prosodic difference in Indian and American accent is perceptible in “am” and “boy”.

4. Architecture of the system

Modularity is one of the key aspects of system that makes it corrigible, improvable and extendable. The modularity of the system enables its functionality to be enhanced to multilingual text to speech systems. The system consists of five building blocks which are:

1. Text Normalization – this phase converts the raw input text to the speakable form.
2. Prosody analysis of utterances – the utterances formed in the text preprocessing phase is further divided into phrases.
3. Word to phoneme conversion – this phase converts the words in the text into the basic sound units, called phonemes.
4. Prosody Analysis of syllables - this phase assigns the various prosodic features such as accent, tone, break indices, stress, etc to the syllables in words.
5. Speech synthesis – this is the final module that converts the processed input text data into the speech signal.

The workflow of the system can be shown as below.

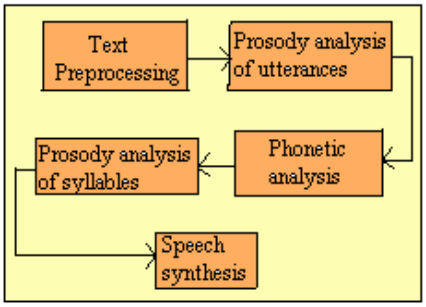

Fig. 3. Architecture of the system.

All the modules have been described further in following sections.

5. Detailed description of the system

5.1 Text Preprocessing

The text-preprocessing phase (converts text into speakable form) can be divided into following sub-phases:

1. Tokenization of the text.

2. Utterance formation.
3. Text Normalization.

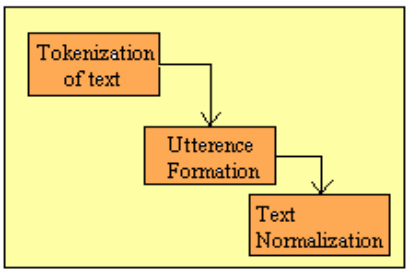

Fig. 4. Architecture of Text preprocessing module.

Tokenization of the text. The speaking style changes with the presence of punctuation marks and paragraph changes in the written text. In order to extract these features from the raw input text, this phase divides the text into small chunks known as tokens. The tokens consists four units:

1. *prepunctuation* : includes the following characters preceding a word " ' ` ({ [
2. *whitespace*: includes blank space(s) , new line character or carriage return
3. *word*: includes the numbers, English words, roman numerals, abbreviations and the single character symbols.
4. *Postpunctuation***:** includes the following characters following a word " ' ` . , : ; ! ? ( ) { } [ ]

These tokens group together to form utterances according to the rules referenced in following section.

Utterance Formation. An utterance is complete unit of talk bounded by silence. Utterances are formed by a set of tokens and the set is determined by using the rules given below. In all the rules below the utterance boundary is determined between two consecutive tokens they are addressed in the text below as previous-token and current-token.

Note that the condition for ‘ Rule n ‘ will be checked only if the condition for all the rules above it i.e. 0 to n-1 are checked.

1. *Rule 1*: If the current token is the first token of the text then there will be no utterance boundary between them.
2. *Rule 2*: If the current token whitespace has more than 1 ‘\n’ then there will be utterance boundary between the token.
3. *Rule 3*:If the previous token postpunctuation has ‘?’ , ‘!’ or ‘:’ then there will be utterance boundary.
4. *Rule 4*: If the previous token postpunctuation has ‘,’ or ‘.’. and current token have whitespace(s) and first character of the current token word is in uppercase then there will be utterance boundary.
5. *Rule 5*: If the previous token prepunctuation have ‘.’ and first character of current token is in uppercase then check:

    a) If the last character of the previous token is in uppercase.

b) If the previous token have word length less than 4 and the first character in uppercase.

If any of (a) or (b) above are true then there is no utterance boundary otherwise there will be the boundary.
Basically, the last rule is for detecting abbreviations. These utterances are independent of each other and are processed separately.

The rules stated above are the standard rules followed in the English language.

Text Normalization. The text to be synthesized to speech may contain numbers, abbreviations, timestamps, e-mail ids, telephone numbers, etc, the "word" unit in the token is processed in this stage to convert it into speakable format.

For example: the number 786 has to be converted into "seven hundred eighty six" before further processing. To automate and simplify the process of converting tokens into words, the method of parsing the regular expressions for various types of text is quite suitable and accurate.

For example: one of the regular expressions for time is "[0-9][0-9]:[0-9][0-9]:[0-9][0-9]".
And for abbreviations "([A-Za-z]\\.)*[A-Za-z]" .
Similarly for decimal numbers : "[0-9]+\\.[0-9]+"
And for apostrophe at the end of the word ,one of the formats can be : "[A-Za-z]+\\'(s|ll|d|ve)".

More such checks will lead to better processing of the text.

5.2 Prosody Analysis of Utterances

Phrase Division. The utterances formed above are divided into the phrases. The breaks are inserted at the beginning of phrases which are being made according to the following rules:

1. New utterance starts with a new phrase.
2. The phrase break-up occurs wherever colon (:) is present in 'post-punctuation' of the token and hyphen ('-') appears in the 'word' part of the token.

5.3 Word to Phoneme conversion

In Hindi Language, pronunciation does not depend on the sequence of the letters in the words. As the Indian accent of speaking English is greatly influenced by Hindi Language, there are a lot of differences in the pronunciation. For example, for the word "God", the phonemes in the Indian style are " g aw d", and in the American style are " g aa d" (phoneme naming convention being used in whole text is of Carnegie Mellon Pronouncing Dictionary (CMPD)).
Before moving further in this stage, it is better to add the part of speech tag to the words to be synthesized into the speech, to have the correct pronunciation of the word in the given text. For example, the suggested ways to have the phonemes as spoken in Indian style are:

1. *Dictionary search*: The available English lexicon dictionaries generally, have the phonemes spoken in either American or British accent. The pronunciation of the words is an important factor that determines the accent of the spoken words in speech. To have the speech in Indian accent suitable changes in the phonemes of the words is to be made so as to reflect the way English is spoken by the Indians. The changes in the dictionary can be made manually or the process can be automated by the algorithms of phoneme detection in speech signals containing the words. The same dictionary prepared above can be used in both the methods stated below.The dictionary may contain a limited number of words. Dictionary search method for phoneme extraction fails whenever the word is not present in the dictionary. Even if the dictionary containing all the possible words in the English language is taken for the purpose of extracting phonemes, the method fails in case of proper nouns. Moreover the user may give any text to be converted in the speech. The system should be able to handle such situations with robustness.
2. *Rule based method*: search method fails Alternative method is to automatically build up the rules [6] for all the possible combinations of the letters. In the reference [6] ,given above the CART trees are trained by the lexicon dictionaries dataset. These CART tress can be used to get the phoneme(s) of unknown word from aligned data. Separate CART trees are made for every letter to parallely build up the rules.
3. *Artificial neural networks*: For the words not present in the dictionary, the artificial neural networks can be trained [7] to guess the phonemes for the words as the human brains do .The method in [7] uses the back-propagation model to train the network using the existing phonemes for the words in lexicon dictionary.

5.4 Prosody Analysis of Syllables

One of the most important aspects of the synthesized speech is that it should appear as natural as possible. Prosody analysis at the syllable level involves: adding the stress, tone and accent features, calculation of the phonemes duration in the speech and the F0 contour generation. The groups of phonemes for each phrase in each utterance formed above are analyzed to mark the tone and accent features in the utterance. Each group of phonemes sharing such properties is known as a syllable.

Adding the stress. This is a vital phase in speech synthesis to produce the realistic effects in speech. Stress is one of the key issues to be emphasized before any further processing. The Lexicon dictionaries are provided with the stress information for the phonemes. The stress information is also modified while making changes in the dictionary for the conversion of dataset in the Indian accent. For example The pronunciation of ‘food’ is given as ‘ f uw d’ in CMPD with no stress on any part of the word. But the Indians speak the same word as ‘f uw1 d’. The digit ‘1’ represents stress on the syllable.

Adding tone and accent features. Tone and accent features can be added by recording the large corpus of speech in an Indian’s voice. The ANN based syntactic prosodic model and GMM based acoustic prosodic model described in [8] can be used

to predict the ToBI labels by giving the recorded corpus as input to the model. The alternative approach is to model the attention and working memory (AWM) [9]. Various prosodic features can be extracted by using the above recorded speech corpus and by mapping AWM.

Finding Duration. The calculations of the play duration of phonemes is a critical task in TTS systems. To find the duration of the phonemes, a large speech corpus is recorded in the voice of an Indian and the following rules [10] are applied:

1. If the target phoneme (with its surrounding phonemes) exactly matches the selected phoneme (with its surrounding phonemes) from the corpus, then the duration of the target will be same as the duration of the selected phoneme.
2. If the target phoneme does not exactly match the phonemes then the duration of the target is calculated using the linear stretching of the phonemes.

Alternatively, the mean value of the phoneme duration in whole corpus can be calculated and the rules can be built up. It will state the change in the value from the duration of the phoneme in different contexts. The duration of the phoneme can be calculated using the relation given below.

Duration = mean duration of the phoneme + (standard deviation) * z-duration (**1**)

The rules can be built up by building the CARTs or the neural networks approach.

F0 Contour generation. The ToBI labels are used to generate F0 contour, which gives the abstract structure of the speech signal to be produced. The F0 contour is produced by the ToBI labels by using the linear regression approach given in [11]. This model uses different weights for different parameters for the intonation of the waveform to get the target F0 values, by the use of relation below.

$$\text{target} = I + w_1 f_1 + w_2 f_2 + w_3 f_3 + \ldots + w_n f_n \quad (2)$$

where, I and $w_{1\ldots n}$, weight for the features, $f_{1\ldots n}$ of the syllable are estimated using the data.

Features in the above equation includes : accent, endtone , break index type, lexical stress, number of syllables from start and to end current phrase, number of stressed syllables from start and end to the current phrase, number of accented syllables from start and end to the current phrase, number of syllables from last accented syllables. It is one of the important phases for incorporating Indian accent and style features in the output speech signal The estimation of the values of I and $w_{1\ldots n}$ pertinent to the Indian prosody is done by using the corpus based approach [12]. First the database of the speech signal, recorded in an Indian's voice is maintained. These speech signals are the group of syllables appearing in different contexts. The target F0 values are estimated by using these waveform contours. The rules for the values of F0 of syllables in different context are thus prepared using a large database set of such syllable groups.

The rules can be built by using automated methods like:

1. *Neural networks approach*: The network is trained using the F0 target values and the syllable features from the corpus recorded in voice of an Indian. The inputs for the neural networks will be the features (mentioned above) of the syllable and the output will be the F0 target value.
2. *Building CARTs*: The inner nodes of the CART are the decision nodes where comparison of the value of different features of the syllable is done and then the decision regarding the direction of descent in the CART is made accordingly. These CARTs are prepared by using the input data and adjusting their decision nodes and structure according to the input data set.

5.5 Speech Synthesis

Speech synthesis is the final stage in text to speech synthesis and finally generates the samples of the speech signal corresponding to the text to be spoken using all the features that have been derived in the above stages to make the speech more natural. The chronological sequence of the operations for this stage is described below.

Diphone Unit selection. In unit selection approach consecutive phones are combined to form the diphones. The diphone is composed of two parts each representing the consecutive phonemes For example: The phones for the word God are: 'g aw d' and the diphones will be – g-aw , aw-d. The final speech signal will be smooth if the phase match is considered while concatenating the consecutive diphones. The above method gives smooth speech but it is difficult to incorporate prosodic features in this approach. Hence the voice produced will not sound as natural as it should have been.

The alternative approach is taking half of each diphone as a unit and the diphones for the text are taken in such a way so as to have the first phone in the diphone same as the second phone of the predecessor diphone. Every diphone has two units associated with it. The first unit has the endtime same as the endtime of the first phone of the diphone and second unit has the endtime equal to the mean of the endtimes of the phonemes in the unit. Suppose in the example taken above, the endtime for the phones be represented as (g, t_1), (aw, t_2), (d, t_3) where t_i (i=1,2,3) represents the endtimes of i^{th} phoneme, then the units and their endtimes will be: (g-aw, t1), (g-aw, (t1+t2)/2), (aw-d, t2), (aw-d, (t2+t3)/2). To make the speech natural, pause is added at the start and end of every phrase. It is incorporated in the speech signal by adding a special symbol 'pause' in start of the phrase to represent silence. Two special units are added in the start of the phrase as 'pau-X' where 'X' represents the first phone of the phrase. The endtime of 'pause' is kept fixed, so as to have a fixed duration of silence at the beginning of every phrase.

Pitchmark Generation. The resonance of some of the signal frequencies produced by vocal tract results in the sharp frequency peaks in the voice signal. These sharp frequency peaks are pitchmarks. The target values and their positions calculated in F0 contour generation are used in this stage. F0 contour provides the overall structure of the waveform for each syllable and pitchmark generation stage adds the finer details of the position of crests and trough in the voice signal to be produced. The pitchmarks are formed when there is resonance with the frequency of the vocal tract with the

signal produced by the source. '*lf0*' is used in the formula below to reflect the effect of the filter on the speech signal produced finally. Positions of pitchmarks are determined by using the following method: The positions of the pitchmarks are calculated, for all the target values, using [13]:

$$t(n) = t(n-1) + \frac{1}{lf0 + (m * t(n-1))} \quad (3)$$

Where

$$m = \frac{(f0 - lf0)}{position} \quad (4)$$

In the above equation, 'position' is the time-position of target value '*f0*', calculated in the F0 contour generation, and '*lf0*' is the pitch value for low pass filter.

Unit Concatenation. Sound database to be used for the TD-PSOLA approach is the RELP encoded voice. Such type of coded voice samples are best suited for this approach as the decoder recovers an approximation of the full-band residual signal, by employing high frequency regeneration which is subsequently used to synthesize output speech. Database is prepared by recording the sound for a text that contains almost all the possible combinations of the phones (that will constitute the diphones). The diphone units are extracted from the sound file and then RELP encoded which will generate the formant and residue values for each sample frame. Frame samples for each diphone can be viewed as consisting of two parts -- the first part for the start of the diphone and second for its ending.

The unit concatenation phase gathers the diphone data from the RELP coded voice and joins them together. Suitable formants are added at the position of pitchmarks from the set of the frame samples for the diphone. The residue values corresponding to the same formant are added for the samples in between the pitchmarks. The set of data of the frame to be used for a particular pitchmark is decided by using the relation:

$$U_{index} = \text{sampling rate} * (S2 - S1)/S \quad (5)$$

where *S1* and *S2* are the number of samples till previous and present pitchmarks repectively and *S* is the number of samples required for the diphone unit. This *Uindex* will give the index of the frame in the diphone.

Re-synthesizing the waveform. The LPC coefficients are calculated by the values of the formant at the pitchmarks, at every channel by using the relation below:

$$LPC_{coefficient} = (\text{formant value} * LPC_{Range}) + LPC_{minimum} \quad (6)$$

LPC_{range} and $LPC_{minimum}$ are determined while encoding the file. The playable sample value can be calculated by using the relation below (using linear prediction technique explained below).There will be one sample for each residue value [14].

$$s'(n) = \sum_{k=1}^{P} a(k) * s(n-k) \quad (7)$$

$$s(n) = e(n) + s'(n) \quad (8)$$

where a(k) is LPC coefficient , s'(n) is the sample for which the value is to be calculated, s(n-k) is the $(n-k)^{th}$ sample whose value has already been calculated and e(n) is the residue part, and P is the number of past samples on which the present sample depends.

6. Results

We have developed 'PalSa –Indian accented English speech synthesizer' using the procedure described in this paper. 'Palsa' converts the raw text given by the user as input in to speech signal. PalSa performs the text preprocessing of the raw text followed by prosody analysis of the utterances formed during text preprocessing as described in the paper.

It makes use of a lexicon dictionary to get the phonemes of the words present in the dictionary. To cover the limitations of the dictionary search method, Palsa uses a neural network with back-propagation model as described in [7], trained using the lexicon dictionary, to get the phonemes of unknown words.

The prosody analysis of the text is done at the utterance as well as syllable level. The rules for labeling the syllable is done as described in section 5.4. The duration of different phonemes is determined followed by F0 contour generation and the adjacent phonemes are concatenated to form diphone units. The sound wave is then generated using the RELP coded diphones.

In order to have an estimate of the efficiency of the system in producing Indian accented speech we have tested it on a sufficiently large group of people with varying age groups as well as varying proficiency in understanding English language. The proficiency in understanding the English language has been taken as a fuzzy set with poor, average and good English proficiency as its linguistic variables mapped, the domain being the percentage efficiency in understanding English varying from 0-100%. The people having the best English in their age group are taken as standard to measure the relative understanding of the people of same age group.

In a similar manner the age group too is a fuzzy set with child, young and old as linguistic variables and the domain ranging from 0 – 70 years of age. The fuzzy rules were created with the results we obtained by playing 100 sentences in American accent followed by Indian accent for the same set of sentences, using PalSa, in front of the people and then recording the percentage of words and sentences they were able to recognize correctly in individual accents. The results for American and Indian accents have been shown below.

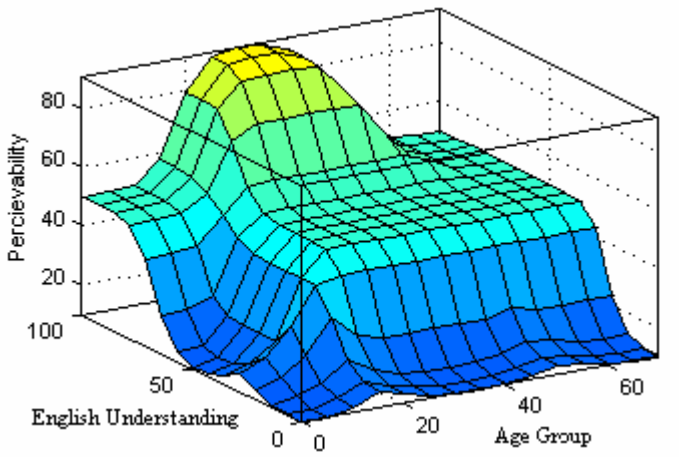

Fig. 5. Results showing the percievabilty of American accented speech for people with varying age groups and corresponding proficiency in understanding English.

The above graph clearly shows that the youth with very good proficiency in understanding English were able to perceive the American accented voice to a very large extent. Even though some of the children and old people who were found to be having a good proficiency in English, were not able to understand the American accented voice completely. The reason being that the majority of Indian children and old people were not acquainted with the American style of speaking English.

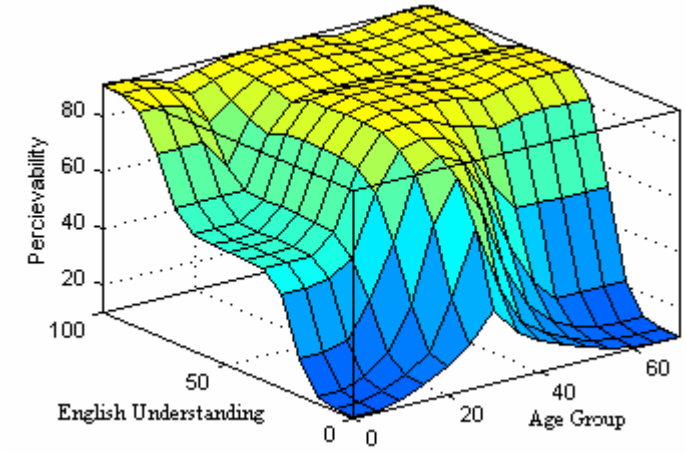

Fig. 6. Results showing the improved percievabilty of American accented speech for Indian people with varying age groups and corresponding proficiency in understanding English.

Remarkable results were obtained when test sentences spoken in Indian accented voice were played in front of people using PalSa. Nearly 100% percievability was obtained in case of children and old people with average and good proficiency in understanding English who gave an average and poor results in case of American accented speech.

The snapshot of PalSa Graphical User Interface is shown below. The text that needs to be spoken in the Indian accented voice is written/pasted in the space provided. The user can choose to play the text either in Indian or American accent by making an appropriate choice of the radio buttons in the top left side of the Graphical User Interface. On pressing the 'speak' button, the phonemes for the text in case of Indian and American accent are displayed in the appropriate text areas.

In order to depict the clear difference between the Indian and American accented speech waveforms, the corresponding signal waveforms are also displayed in the bottom part of the Graphical User Interface. The results for the query 'God' have been included in the snapshot as shown below.

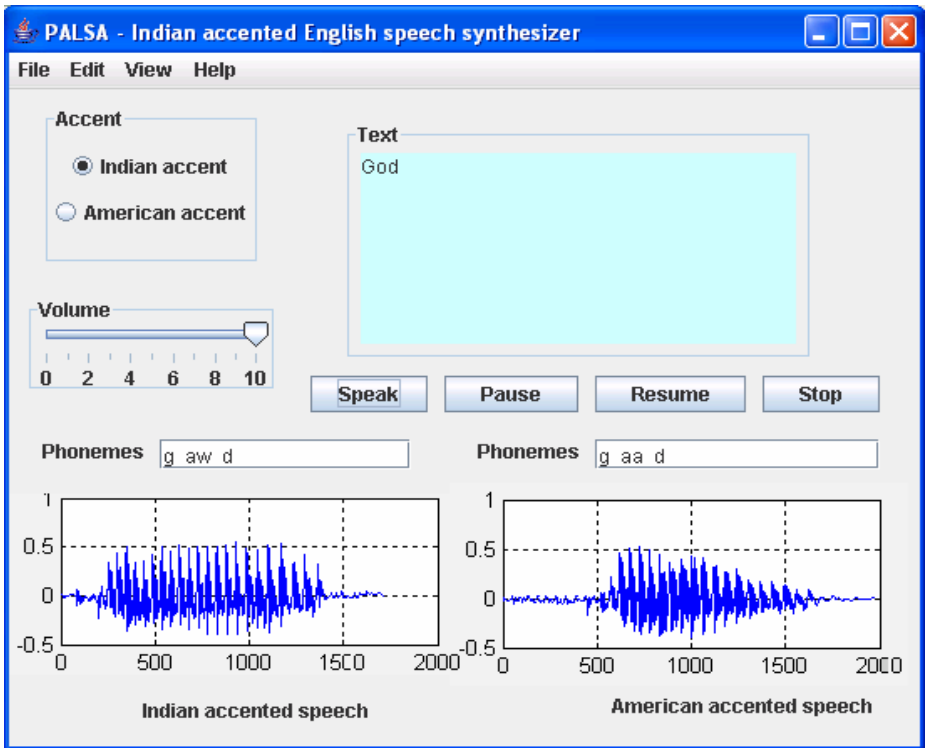

Fig. 7. The Graphical User Interface of 'PalSa – the Indian accented English speech synthesizer

7. Conclusion

The method given in the paper gives the full procedure to build up the speech synthesizing system. As the whole procedure is modular so changes and improvements can be made in any of the steps to reflect the subsequent change in the

output. The procedure can be used to build up the speech synthesizer for different languages if suitable speech corpus is built in that accent.

8. Acknowledgements

We are very thankful to Dr. Sudip Sanyal, Associate professor, Indian Institute of Information Technology(IIIT) – Allahabad. We also express our gratitude to thank Dr. Alan W. Black, Associate Research Professor, CMU and Mr. Willie Walker, Manager and Principal Investigator, Speech Integration Group of Sun Microsystems Laboratories for their kind help through mails and forum responses.

References

1. Kwok, P., Lamere, P., Schröder, M., Vos, D., Walker, W.: FreeTTS – A speech synthesizer written entirely in the Java programming language (2004)
2. The Festival Speech Synthesis System: http://www.cstr.ed.ac.uk/projects/festival/
3. The CMPD Pronouncing Dictionary : Carnegie Mellon University: http://www.speech.cs.cmu.edu/cgi-bin.
4. Black, A.W. and Lenzo, K.A.: Flite System documentation, by, Speech Group at Carnegie Mellon University (2003) : http://www.speech.cs.cmu.edu/flite/doc/
5. Black, A.W. and Lenzo, K.A : Building Synthetic Voices, the festvox system documentation (2003)
6. Black, A.W., Lenzo, K.A.., Pagel, V. : Issues in Building General Letter to Sound Rules In: ESCA Synthesis Workshop, Australia (1998) 77-80
7. Arciniegas, F. and Embrechts, M. J.: Artificial Neural Networks (ANNs) for Phoneme Recognition for Text-to-Speech Applications In: IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy (2000)
8. Chen, K., Cohen, A., Hasegawa-Johnson, M. : An automatic prosody labeling system using an ANN based syntactic prosodic model and GMM based acoustic prosodic model In: Acoustics, Speech, and Signal Processing Proceedings. (ICASSP '04). IEEE International Conference on Volume 1 (2004) I - 509-12 vol.1
9. Cahn , J.E. : A computational memory and processing model for prosody In: arXiv:cs.CL/9904018 v1 Massachusetts Institute of Technology (1999)
10. Meron J.: Prosodic unit selection using an imitation speech database. In: 4th ISCA Tutorial and Research Workshop on Speech Synthesis, Perthshire, Scotland (2001) 53-57
11. Black, A.W., Hunt, A.J. : Generating F0 contours from ToBI labels using linear regression. In: Proceedings of ICSLP 96, Philadelphia, PA, USA (1996) 229-232
12. Black, A. and Raux : A unit selection approach to f0 modeling and its application to emphasis. In: ASRU2003, St Thomas, Virgin Islands (2003)
13. Sun, X. : Predicting Underlying Pitch Targets for Intonation Modeling. In: 4th ISCA tutorial and Research Workshop on Speech Synthesis, Perthshire, Scotland (2001)
14. Fs, Pacheco, Rui, Seara : Prosodic Speech Modification Using RELP. In: IEEE International Telecommunications Symposium, Natal, RN (2002)
15. Kataria, A., Kumar, R., Sofat S. : Building Non - Native Pronunciation Lexicon for English using a Rule based Approach. In: International Conference on Natural Language processing (ICON) 2003, Mysore, India (2002)