Current Trends, Frameworks and Techniques Used in Speech Synthesis – A Survey

Benoy Kumar Thakur, Bhusan Chettri, Krishna Bikram Shah

***Abstract*— Vocalized form of human communication is Speech. Here, we have reviewed some of the most popular and effective techniques used to generate synthetics speech. In this quest we are able to find the scenario where one method is advantageous over another. We have discusses Text To Speech Architecture putting more emphasize on the two components, namely, Natural Language Processing (NLP) and Digital Signal Processing (DSP). We have also reviewed some of the most popular generic frameworks like MBROLA, FESTIVAL, and FLITE that available in public domain for the development of a TTS synthesizer.**

***Index Terms*— Speech Synthesis, Synthesized Speech, Text-to-Speech, TTS, Artificial Speech, speech synthesizer.**

I. INTRODUCTION

Speech is the vocalized form of human communication. It is based upon the syntactic combination of lexical and names that are drawn from very large vocabularies generally consisting of more than 10,000 words. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech, which are the sound units in speech synthesis. Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a Speech Synthesizer, and can be implemented in software or hardware. A Text-To-Speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech [1].

Synthesized speech is created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units. A system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely “synthetic” voice output [2].

Synthetic or artificial speech has been developed steadily during the last decades. The objective of this survey is to map the current situation of speech synthesis technology. Speech synthesis may be categorized as restricted (messaging) and unrestricted (text-to-speech) synthesis. The first one is suitable for announcing and information systems while the latter is needed for example in applications for the visually Impaired. The text-to-speech procedure consists of two main phases, usually called Natural Language Processing (NLP) or high level and Digital Signal Processing (DSP) or low-level synthesis. In high-level synthesis the input text is converted into such form that the low-level synthesizer can produce the output speech. The three basic methods for low-level synthesis are the formant, concatenative, and articulatory synthesis. The formant synthesis is based on the modeling of the resonances in the vocal tract and is perhaps the most commonly used during last decades. However, the concatenative synthesis which is based on playing prerecorded samples from natural speech is becoming more popular. The most accurate method is articulatory synthesis which models the human speech production system directly, but it is also the most difficult approach. Since the quality of synthetic speech is improving steadily, the application field is also expanding rapidly. Synthetic speech may be used to read e-mail and mobile messages, in multimedia applications, or in any kind of human-machine interaction. The evaluation of synthetic speech is also an important issue, but difficult because the speech quality is a very multidimensional term. The most commonly used criteria for high-quality speech are intelligibility, naturalness and pleasantness. Since these are multidimensional factors that depend on each other, the comprehensive high quality is formed by the interaction of numerous details. Thus, the elimination of background noise, musical noise, mumbling, and the various pops and cracks, does not result in the ultimate quality, but the speech should also be made rich in nuances and it should carry information about the personality of the speaker. Moreover, it would be advisable to include in the speech some features that describe the emotional state of the speaker because this improves the naturalness and makes the speech more lively. Given the continuously increasing processing capacity, more and more complicated speech synthesizers can be used even on personal computers. The overall goal of the speech synthesis research community is to create natural sounding synthetic speech. To increase naturalness, researchers have been interested in synthesizing emotional speech for a long time. One way synthesized speech benefits from emotions is by delivering certain content in the right emotion. Take for example, good news are delivered in a happy voice, therefore making the speech and the content more believable. Emotions can make the interaction with the computer more natural because the system reacts in ways that the user expects. Emotional speech synthesis is a step in this direction. The implementation of emotions seems straightforward at first but a closer look reveals many difficulties in studying and implementing emotions. The difficulties start with the definition of emotions. Researchers agree that emotions are not as often thought of, just a subjective experience or feeling. Today, speech synthesizers of various quality

Manuscript received on April 26, 2012.

Benoy Kumar Thakur, Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Sikkim, India, (e-mail: b9thakur2004@gmail.com).

Bhusan Chettri, Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Sikkim, India,
(e-mail: bhusan.chettri@gmail.com).

Krishna Bikram Shah, Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Sikkim, India, (e-mail: krishnabikramshah@gmail.com).

are available as several different products for all common languages, including English, Hindi, French, Bangali and Nepali.

II. ARCHITECTURE OF A TTS SYNTHESIZER

This section discusses in more detail the two components, namely, Natural Language Processing (NLP) component and Digital Signal Processing (DSP) component (Fig. I) of a typical TTS synthesizer [3]

NLP Component

A general NLP module can be seen as consisting of three major components (Fig. II) [3].

Text Analyzer: The text analyzer consists of the following four modules:

- A pre-processing module, which organizes the input sentences into manageable lists of words. It identifies numbers, abbreviations, acronyms and idiomatics and transforms them into full text where so ever needed.

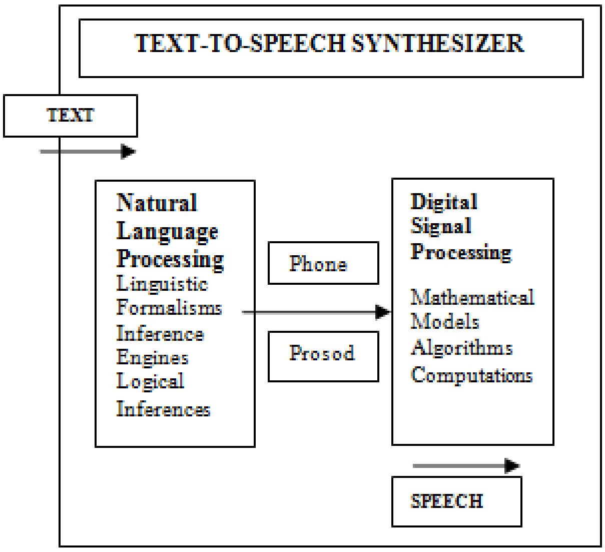

Fig. I: Basic components of a TTS synthesizer

- A morphological analysis module, the task of which is to propose all possible part of speech categories for each word taken individually, on the basis of their spelling.
- The contextual analysis module considers words in their context, which allows it to reduce the list of their possible part of speech categories to a very restricted number of highly probable hypotheses, given the corresponding possible parts of speech of neighboring words.
- Finally, a syntactic-prosodic parser, which examines the remaining search space and finds the text structure (i.e., its organization into clause and phrase-like constituents) which more closely relates to its expected prosodic realization.

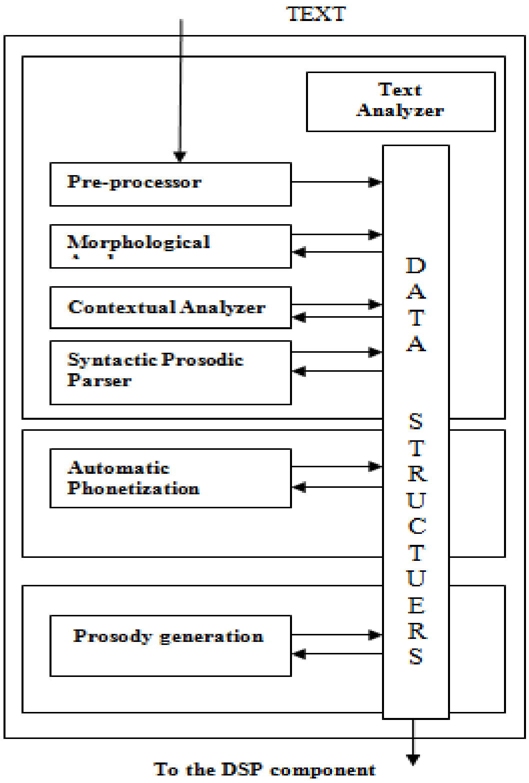

Fig. II: NLP components

Automatic Phonetization: The Automatic Phonetization module (also known as Letter-To-Sound (LTS) module) is responsible for the automatic determination of the phonetic transcription of the incoming text. Pronunciation dictionaries may not always help in this module due to the following facts:

- The pronunciation dictionaries refer to word roots only. These do not explicitly account for morphological variations (i.e. plural, feminine, conjugations, especially for highly inflected languages).
- Words embedded into sentences are not pronounced as if they were isolated. It can also be noted that not all words can be found in a phonetic dictionary. The pronunciation of new words and of many proper names has to be deduced from the one of already known words.

The two popular ways of implementing an Automatic Phonetization module are:

- Dictionary-based solutions that consist of storing a maximum of phonological knowledge into a lexicon.
- Rule-based transcription systems that transfer most of the phonological competence of dictionaries into a set of letter-to-sound (or grapheme-tophoneme) rules. This time, only those words that are pronounced in such a particular way that they constitute a rule on their own are stored in an exceptions dictionary.

Prosody Generation: Prosodic features consist of pitch, duration, and stress over the time. With a good control over these features, gender, age, emotions, and other features can be incorporated in the speech and very natural sounding speech can be modeled. However, almost everything seems to have an effect on prosodic features of natural speech and it makes accurate modeling very difficult (Fig. III)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

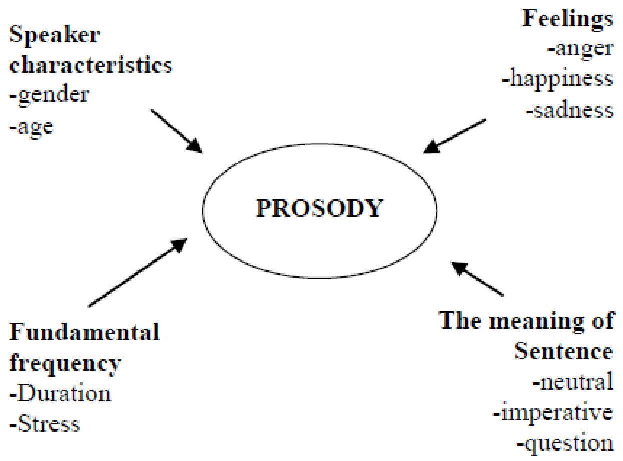

Fig. III: Prosodic Dependencies

Prosodic features can be divided into several levels such as syllable, word, and phrase level. For example, at word level vowels are more intense than consonants. At phrase level correct prosody is more difficult to produce than at the word level. The three features of the prosody are described below in brief. The pitch pattern or fundamental frequency over a sentence (intonation) in natural speech is a combination of many factors. The pitch contour depends on the meaning of the sentence. For example, in normal speech the pitch slightly decreases towards the end of the sentence and when the sentence is in a question form, the pitch pattern will raise to the end of sentence. In the end of sentence, there may also be a continuous rise which indicates that there is more speech to come. A raise or fall in fundamental frequency can also indicate a stressed syllable [6]. Finally, the pitch contour is also affected by gender, physical and emotional state, and attitude of the speaker. The duration or time characteristics can also be investigated at several levels from phoneme (segmental) durations to sentence level timing, speaking rate, and rhythm. The segmental duration is determined by a set of rules to determine correct timing. Usually, some inherent duration for phoneme is modified by rules between maximum and minimum durations. In general, the phoneme duration differs due to neighboring phonemes. At sentence level, the speech rate, rhythm, and correct placing of pauses for correct phrase boundaries are important. The intensity pattern is perceived as a loudness of speech over the time. At syllable level, vowels are usually more intense than consonants and at a phrase level, syllables at the end of an utterance can become weaker in intensity. The intensity pattern in speech is highly related with fundamental frequency. The intensity of a voiced sound goes up in proportion to fundamental frequency [6].

DSP Component

The DSP component is also called as the synthesizer component. Different TTS synthesizers can be classified according to the type of synthesis technique that is used to synthesize the speech. The methods are usually classified into three groups Articulatory,Formants and Concatenative synthesis .

Articulatory Synthesis: Articulatory synthesis is based on modeling the human speech production system hence this approach is capable of producing high quality speech. This method is computationally more difficult when compared to other methods [7]. Articulatory models are useful to study the physics of speech production [8]. Articulatory synthesis involves models of the human articulators and vocal cords. The articulators are usually modeled with a set of area functions between glottis and mouth. For rulebased synthesis the articulatory control parameters may be for example, lip aperture, lip protrusion, tongue tip height, tongue tip position, tongue height, tongue position and velic aperture. Phonatory or excitation parameters may be glottal aperture, cord tension, and lung pressure [7].The rule base consisting of the area functions for various sounds can be obtained by X-ray analysis of natural speech.

Format Synthesis: This models the pole frequencies of speech signal or transfer function of vocal tract based on source-filter-model. Thus speech is generated by an acoustic-phonetic production model, based on formant of the vocal tract. The acoustic-phonetic parameters such as energy, pitch and resonance (formant) frequencies associated with speech and heuristic rules are used to derive the model. The first formant synthesizer, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953 [6]. PAT consisted of three electronic formant resonators connected in parallel. The input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency, and noise amplitude.

Concatenative Synthesis

This uses different length prerecorded samples derived from natural speech. Here, speech is generated by combining splices of pre-recorded natural speech. The articulatory and formant synthesis are also classified as rule-based synthesis methods whereas the concatenative technique falls under database driven synthesis method. The formant and concatenative methods are the most commonly used in present synthesizers. The formant synthesis was dominant for long time, but these days, the concatenative method is becoming more and more popular. The articulatory method is still too complicated for high quality implementations, but may arise as a potential method in the future. There are a number of different methodologies for Concatenative Synthesis such as TDPSOLA, PSOLA [9], MBROLA [10] and Epoch Synchronous Non Over Lapping Add (ESNOLA) [10]. There are three main subtypes of concatenative synthesis:

Unit selection synthesis: This type of synthesis uses large speech databases (more than one hour of recorded speech). During database creation, each recorded utterance is segmented into some or all of the following: individual phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some hand correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially-weighted decision tree. Unit selection gives the greatest naturalness due to the fact that it does not apply a large amount of digital signal processing to the recorded speech, which often makes recorded speech sound less natural, although some synthesizers may use a small amount of signal processing at the point of concatenation to smooth the waveform.

Diphone synthesis: It uses a minimal speech database containing all the Diphones (sound-to-sound transitions) occurring in a given language. The number of diphones depends on the phonotactics of the language: Spanish has about 800 diphones and German has about 2500 diphones. In diphone synthesis, only one example of each diphone is contained in the speech database. At runtime, the target prosody of a sentence is superimposed on these minimal units by means of digital signal processing techniques such as Linear predictive coding, PSOLA[9] or MBROLA[10]. The quality of the resulting speech is generally not as good as that from unit selection but more natural-sounding than the output of formant synthesizers. Diphone synthesis suffers from the sonic glitches of concatenative synthesis and the robotic-sounding nature of formant synthesis, and has few of the advantages of either approach other than small size. As such, its use in commercial applications is declining, although it continues to be used in research because there are a number of freely available implementations

Domain-specific synthesis: It concatenates pre-recorded words and phrases to create complete utterances. It is used in applications where the variety of texts the synthesizer will output is limited to a particular domain, like trains schedule announcements or weather reports. This technology is very simple to implement, and has been in commercial use for a long time: this is the technology used by gadgets like talking clocks and calculators.

Some other synthesis methods are:

Hybrid Synthesis: marries aspects of formant and concatenative synthesis to minimize the acoustic glitches when speech segments are concatenated.

HMM-based Synthesis: It is a synthesis method based on Hidden Markov Models (HMMs) [11]. In this type of synthesizer, speech frequency spectrum (vocal tract), Fundamental frequency (vocal source), and duration (prosody) are modeled simultaneously by HMMs. Speech waveforms are generated from HMMs themselves based on Maximum likelihood criterion.

HNM –based Synthesis: A HNM model was first presented in [12]. HNM assumes that the speech signal is composed of a harmonic and a noise part. The harmonic part responds to the quasiperiodic components of the speech and the noise part responds to non-periodic components. These two components are separated in the frequency domain by a time-varying parameter called maximum voiced frequency Fm. The bandwidth up to Fm is represented by harmonic sinusoids and the bandwidth from Fm is represented by a modulated noise component. Unvoiced parts of speech are represented only by noise part. The speech signal is obtained as a sum of the harmonic and the noise part. The harmonic part contains only harmonic multiples of fundamental frequency. The noise part can be modeled by coding spectral envelope using AR filter, where the synthesis is done by filtering white noise by the AR filter. Since the noise part has no fundamental frequency, the F0 is set to 100 Hz as stated in [12]. The phases of sinusoids are set randomly because the noise is a stochastic signal.

Sinusoidal Model based Synthesis: Sinusoidal models are based on a well known assumption that the speech signal can be represented as a sum of sine waves with time varying amplitude and frequencies [13].

Linear predictive methods: Linear predictive methods are originally designed for speech coding systems, but may also be used in speech synthesis. In fact, the first speech synthesizers were developed from speech coders. Like formant synthesis, the basic LPC is based on the source-filter-model of speech described. The digital filter coefficients are estimated automatically from a frame of natural speech.

III. TTS FRAMEWORKS

MBROLA, FESTIVAL, FLITE are some of the generic frameworks available in public domain for the development of a TTS synthesizer. Some of this act as back-end engines and others are full-featured commercial TTS frameworks.

MBROLA: It is a high-quality, diphone-based speech synthesizer that is available in public domain. It is provided by the TCTS Lab of the Faculte Polytechnique de Mons (Belgium) which aims to obtain a set of speech synthesizers for as many languages as possible. The MBROLA speech synthesizer is free of charge for non-commercial, non-military applications. Anyone can send in his or her own speech recordings and an MBROLA database for synthesis is prepared. There are presently diphone databases existing for several languages: American English, Brazilian Portuguese, Breton, British English, Dutch, French, German, Greek, Romanian, Spanish and Swedish TCTS also provides speech database labeling software: MBROLIGN, a fast MBROLA-based TTS aligner. MBROLIGN can also be used to produce input files for the MBROLA v2.05 speech synthesizer. More information and demos of the different voices and languages and also comparisons between MBROLA and other synthesis methods can be found on the MBROLA project home page [14].

FESTIVAL: The Festival TTS synthesizer was developed in CSTR at the University of Edinburgh by Alan Black and Paul Taylor and in co-operation with CHATR, Japan [15]. It is a freely available complete diphone concatenation and unit selection TTS synthesizer. Festival is the most complete freeware synthesis system and it includes a comprehensive manual. Festival offers a general framework for building speech synthesis systems as well as including examples of various modules. As a whole, it offers full TTS synthesizer through a number of APIs. Festival is multi-lingual (currently English, Spanish and Welsh). The English version is most advanced and the developments for this version are very fast. The synthesizer is written in C++ and uses the Edinburgh Speech Tools for low-level architecture and has a Scheme (SIOD)-based command interpreter for control [16].

FLITE: Flite (festival-lite) is a small, fast run-time speech synthesis engine developed at CMU and primarily designed for small embedded machines and/or large servers. Flite is designed as an alternative synthesis engine to Festival for voices built using the FestVox suite of voice building tools.

Tools Available for Development of a TTS Synthesizer: The tools available for developing a TTS synthesizer include speech API's provided by different vendors, and different markup languages. There exist many different APIs for speech output but there is a trend towards using the Microsoft API for synthesizers running on Windows. Another API that is not so frequently used is the Sun-Java Speech API. These two are described below.

Sun – Java Speech APT: The Java Speech API is being developed to allow Java applications and applets to incorporate speech technology. The API defines a cross-platform API to support command and control recognizers, dictation systems and speech synthesizers. Java Speech Grammar Format provides a cross-platform control of speech recognizers. Java Speech Markup Language provides a cross-platform control of speech synthesizers. Text is provided to a speech synthesizer as a Java String object. The Java Platform uses the Unicode character set for all strings. Unicode provides excellent multi-lingual support and also includes the full International Phonetic Alphabet (IPA), which can be used to accurately define the pronunciations of words and phrases.

SAPI - Microsoft Speech API: The leading vendors are beginning to support Microsoft's Speech API, or SAPI, which is based on the COM specification and is being adopted as the industry standard. The motive of SAPI is to eventually allow interoperability between the speech engines. The Microsoft Speech API provides applications with the ability to incorporate speech recognition (command & control dictation) or TTS, using either C/C++ or Visual Basic. SAPI follows the OLE Component Object Model (COM) architecture. It is supported by many major speech technology vendors. The major interfaces are:

- *Voice Commands:* high-level speech recognition API for command and control.
- *Voice Text:* simple high-level TTS API. The Voice Text object is available in two forms: a standard COM interface IVoiceText and companion interfaces, and also an ActiveX COM object, VtxtAuto.dll
- *Multimedia Audio Objects:* audio I/O for microphones, headphones, speakers, telephone lines, files etc. With the Microsoft Speech SDK, and in particular, the TTS VtxtAuto ActiveX COM object, any developer can create a TTS-enabled application using a few simple commands, such as register and speak.

Markup Languages: The input to a TTS synthesizer is often a string of words but sometimes it also contains information in form of markers to indicate emphasis, stress placement, speech rate, voice etc. System providers normally have their own markup codes but there is some co-operation between providers to develop standards for markups. A number of markup languages have been established for rendition of text as speech in an XML compliant format.

- SSML: Speech Synthesis Markup Language (SSML) is an XML-based markup language for speech synthesis applications. It is a recommendation of the W3C's voice browser working group. SSML is often embedded in VoiceXML scripts to drive interactive telephony systems. However, it also may be used alone, such as for creating audio books. For desktop applications, other markup languages are popular, including Apple's embedded speech commands, and Microsoft's SAPI Text to speech (TTS) markup, also an XML language. SSML is based on the Java Speech Markup Language (JSML) developed by Sun Microsystems, although the current recommendation was developed mostly by speech synthesis vendors.
- JSML: The Java Synthesis Markup Language (JSML), an SGML-based mark-up language, is being specified for formatting text input to speech synthesizers. JSML allows applications to control important characteristics of the speech produced by a synthesizer. Pronunciations can be specified for words, phrases, acronyms and abbreviations to ensure comprehension. Explicit control of pauses, boundaries and emphasis can be provided to improve naturalness. Explicit control of pitch, speaking rate and loudness at the word and phrase level can be used to improve naturalness and comprehension.

•

IV. CONCLUSION

We find that there is lot of work needed to be done related to prosody section of Natural Language processing of text to speech. In the DLP, we find that Formant synthesis and articulatory synthesis are less used today but these techniques can be suitable for applications that require less memory and low processing cost. The focus nowadays is on the unit selection synthesis combined with harmonic plus noise model (HNM). Festival offers a general framework for building speech synthesis systems.

REFERENCES

1. Jonathan Allen, M. Sharon Hunnicutt, Dennis Klatt, "From Text to Speech: The MITalk system", Cambridge University Press, 1987.
2. Rubin, P.; Baer, T.; Mermelstein, P., "An articulatory synthesizer for perceptual research". Journal of the Acoustical Society of America 70: 321–328, 1981
3. Dutoit T, "High-quality text-to-speech synthesis: an overview. Journal of Electrical & Electronics Engineering," Australia: Special Issue on Speech Recognition and Synthesis, vol. 17, pp 25-37
4. Allen J, "Synthesis of speech from unrestricted text. IEEE Journal", Vol.64, Issue 4, pp 432-42, 1976
5. Allen J, Hunnicutt S, Klatt D , "From text-to-speech: the MITalk system", Cambridge University Press, Inc., 1987
6. Klatt D, "Review of text-to-speech conversion for English", Journal of the Acoustical Society of America, vol. 82, pp 737-93
7. Sami Lemmetty, "Review of Speech Synthesis Technology," Master's Thesis, Dept. of Electrical and Communication Engineering, Helsinki University of Technology, March 30, 1999.
8. O'Saughnessy D, Speech Communications – Human and Machine, University Press. 2001
9. David Öhlin, Rolf Carlson "Data-driven formant synthesis" Proceedings, FONETIK 2004, Dept. of Linguistics, Stockholm University.
10. P A TAYLOR, "Concept-to-Speech Synthesis by Phonological Structure Matching".
11. T.Yoshimura, K.Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum,pitch and duration in HMM-based speech synthesis", Proc. Eurospeech, pp.2347-2350,1999.
12. Y. Stylianou, "Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification," Ecole Nationale Supérieure des Telecommunications, Paris, January 1996.
13. R.J. McAulay and T.F. Quatieri, "Speech analysis-synthesis based on a sinusoidal representation," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, no. 4, pp. 744-754 August 1986.
14. MBROLA, "Project homepage", 1998. Online: http://tcts.fpms.ac.be/synthesis/mbrola.html/
15. Black,"User Manual for the Festival Speech Synthesis System", version1.4.3, 2001. Online: http://fife.speech.cs.cmu.edu/festival/cstr/festival/1.4.3/
16. Black A, Taylor P, Caley R (2001) The Festival speech synthesis system: system documentation. University of Edinburgh. Online: http://www.cstr.ed.ac.uk/projects/festival/.