Intelligent Speech Synthesizer

[2]Gregor Rozinaj, [1]Renata Rybárová
Department of Telecommunications, FEI STU Bratislava, renata.rybarova@stuba.sk
Department of Telecommunications, FEI STU Bratislava, gregor.*rozinaj@stuba.sk*

Abstract

This paper is focused on a design of an intelligent speech synthesizer and prosody modification of Slovak language. It contains analysis and module architecture design for the speech synthesizer with learning system implemented for each module (phonetic transcription, phonetic transcription of abbreviations, word class identification, and prosody modification module). The scope of this paper also includes analysis of prosody modification using sinusoidal models.

Keywords*: speech synthesis, intelligent system, speech synthesizer, prosody, sinusoidal model, sinusoid parameters*

1. Introduction

Speech is fundamental human expression and basic way how people share or exchange information, ideas or feelings. The newest technology tries to bring synthesized speech in everyday's life. However this "machine" speech is not always the same as human voice.

The idea that a machine could generate speech has been in minds of people for a long time already, but the realization has only really been practical only within the last 50 years. The rise of concatenative synthesis began in the 70s, and has largely become practical as large-scale electronic storage has become cheap and robust.

With the rising hardware power of electronic devices (including phones) the human computer interaction has started to develop quite a lot of new PC and mobile applications in the last decade. The goal of this effort is to integrate multiple input-output modalities into one interface. In this interface the speech modality is processed by automatic speech recognition (ASR) and text-to-speech (TTS) synthesis systems. Depending on the implemented complexity couple of modes of speech interaction has emerged: embedded systems that run locally on the mobile devices [1][2][3], client-server systems that put most of the load on a remote server [4][5], or those that are capable of both these modes [6].

The main purpose of this article is focused on a design of a speech machine for Slovak language – an intelligent speech synthesizer with prosody modification. Improvements of already known methods, analysis, module architecture and new algorithms are devised for achieving better synthetic speech quality. The learning mechanism can be implemented for each module using various levels of intelligence. These levels will be described in following chapters.

One of the possibilities of prosody modification is usage of the sinusoidal models. Main advantage of these models is their parametric form which allows a very flexible signal change in both time and frequency domain which means that we would be able to change the tone pitch and other characteristics of voice via simple change of parameter. In our analysis of dependencies between human voice and sinusoidal parameters a HNM model was used.

2. Speech synthesizer's architecture

Speech synthesis means creating human-like speech using a machine, which is known as speech synthesizer.

There are several types of these synthesizers, but each is designed to achieve the same goal: to reproduce given text in the clearest and most understandable manner. There are four basic approaches: synthesis using units, formant synthesis, articulation synthesis, HMM synthesis.

The principle of each synthesizer's core is the same: speech units are chosen from the speech database and put together to create desired outcome. To achieve natural synthesized speech the synthesizers should fulfill more complex tasks like preprocessing and post-processing. It's better to implement each process independently and create module for each process. Depending on desired

quality and/or availability of resources it can be chosen which module should contribute on process of synthesis and vice versa which one can be turned off to save time or energy. The basic overview of the modular architecture is shown on Figure 1[7].

In this figure user accesses the synthesizer via a web interface. User types a text and sends request for synthesis. One main process is used to control communication between the synthesizer's modules. Each module fulfills different task: one module deals with text analyze, another one with synthesis, the other one with post processing. There are rules and defined format of incoming and outgoing messages between modules. In our proposal we suggest to use XML format, which basically is a set of rules for encoding documents in machine-readable form. It is a textual data format with strong support via Unicode for the languages of the world and is widely used for the representation of arbitrary data structures, for example in web services. The main advantages of XML format are:

- transparent and verifiable information exchange
- development IDE support
- markup and passing of structured information
- easy to add new information

Each element in XML file (sentence, word, val,…) has its own meaning. And since most of the XML elements are optional, the XML file is used to pass information and arguments between the modules. The first module receives only the entered text and adds the output into the optional XML tags. At the end of the process information from all participating modules are entered into the XML file [7].

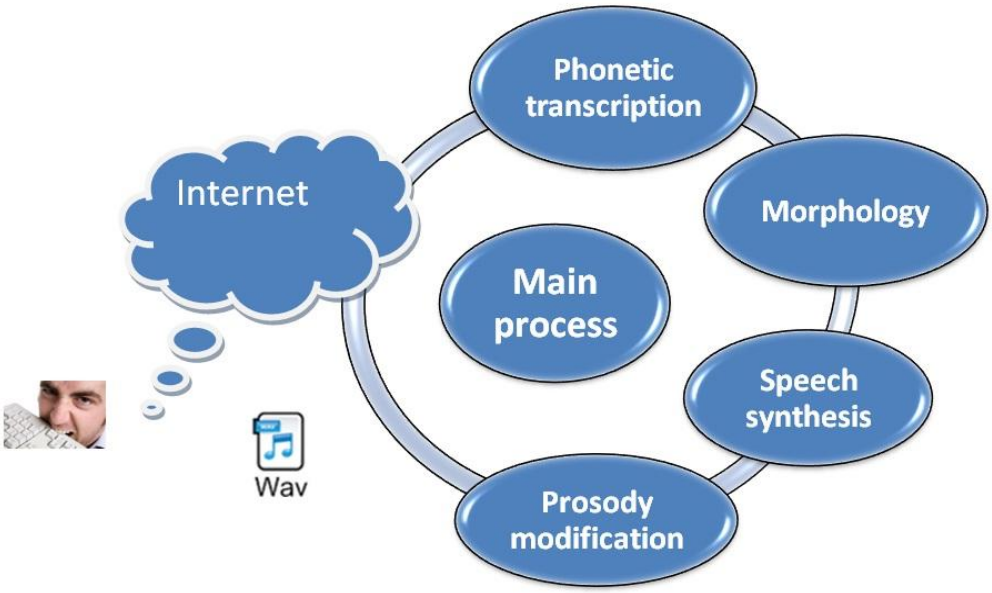

Figure 1. The modular architecture of speech synthesizer

3. Learning mechanism

Naturally synthesized speech is a speech that is not recognizable from human speech. Creating this kind of speech is very difficult and complex task. The first idea how to reduce artificiality in synthesized speech was to create of database with units cut from recorded human speech. The size units used in database depends on database volume - the larger the database, the larger unit in use: phonemes, diphones, words or group of words. The larger the database unit is the more natural speech we get. Nevertheless human ear still detects an artificially merged signal that by design locally shows some inconsistencies.

The goal of perfect synthesized speech requires more than good database. Text has to be analyzed and prepared for synthesis, e.g. it has to be retyped into machine language, rewrite abbreviations and numbers into full form and for Slovak language also determine parameters like word class, gender, case and number. Afterwards prosody of synthesized speech should be modified to make the speech more fluent and natural.

3.1. Phonetic transcription module

People in generally know how to pronounce words in their tongue language. We try to implement the very same knowledge into synthesizers.

The first method of phonetic transcription is transcription according to letter – to - sound (LTS) rules. Rules can be created by a phonetic expert or by a corpus method. The corpus method automatically generates LTS rules according to contextual events in a manually described corpus. Because there are always new words coming into language, for example from other languages, its necessary update LTS rules regularly.

The basic assumption for creating LTS rules is that each phoneme has its own transcription into phonetic form. There are three different categories of letters in Slovak language. The first group consists of phonemes which are written and pronounced the same way, there is no dependency on phonemes before or after them (for example a). The second group consists of different phonemes that are pronounced the same way (for example i,y). The last group contains phonemes which can be pronounced in more ways in dependency from phonemes before or after them (for example d can be pronounced as d or t) [7].

The other method of phonetic transcription is transcription according to a list of exceptions. So each word has his transcription and this is saved in the list.

The phonetic transcription based on known methods described above is following:

- In the first step the list of exceptions is checked. In case word is found there, the phonetic transcription form from list is taken. In other case the LTS rules are used.
- For words, not present in list of exceptions, LTS rules are used. Generally it's more comfortable to use LTS rules, because make phonetic transcription of large text only with the list of exceptions would be time consuming. LTS rules are organized in a tree structure and it makes searching much faster. Also adding a new rules or modification of existing one is very simple.

None of these methods correctly covers all words and exceptions, therefore the best way is to use both of them. The list contains exceptions in the language, for example words from other languages for which the LTS rules are not applicable. At first the synthesizer checks the list of exceptions. If a word is not found there, LTS rules are applied. In case when a word is not correctly retyped because LTS rules are incorrect or missing, we propose remediation actions, which are described in chapter 3.5 [7].

3.2. Module of phonetic transcription of abbreviations

The other big task for speech synthesizer is abbreviations. Correct reading of abbreviations makes synthesized speech more natural. And as reading of abbreviations is very individual and differs from person to person, this makes this task even more difficult. There are more possibilities to how to read abbreviations: they can be replaced by words, spelled or pronounced if it's possible. It also important to consider how many time is abbreviation placed in text. If it's in article more times, it can be replaced by whole words (full meaning of abbreviation) the first time and spelled later on, or it can be read in case it's readable [7].

The process how to improve already existing methods for reading abbreviation is as follows [7]:

- Preprocessing phase looks for abbreviation in the synthesized text. In case of longer article is this text analyzed and based on abbreviation count it is decided how to read it. If possible, the abbreviation is replaced by its glossary explanation. In case of multiple occurrences the first time it can be read in full meaning and later only spelled, or it can be mixed – spelled and read in full meaning.

- Another possibility how to increase the naturality of abbreviation expansion is to simply read them as words. Of course not every expression can be pronounced fluently, e.g. expressions like „FIFA“, „UNESCO“ are comfortable to pronounce. So one part of processing should be analysis if the abbreviation is vocable. In Slovak language a word is vocable if it contains vowels (a, e, i, o, u) or syllabical consonant (r, ŕ, l, ĺ). In that case can be spelling be replaced by reading and the synthesizer chooses method for reading as previously described.
- In case abbreviation is not in glossary, it is checked if it is vocable. In that case is read or spelled. In case it's not vocable, synthesizer spells it every time.

For intelligent reading and intelligent synthesizer it is necessary not only complete abbreviation glossary, because in ideal case synthesis process should determine all parameters for abbreviation like word class, gender, case and number. So the process of reading abbreviation is not isolated, it also depends on correct text preprocessing and determination of all word parameters.

3.3. Word class identification module

The words in Slovak language are inflected, e.g. they have different forms in different cases. Each word has a group of phonemes which are never changed, called a word core. The difference is in a suffix – the last three phonemes. In Slovak language words with the same suffix are grouped together to the word class.

In speech synthesis process it is very important to determine the correct word class. As was mentioned before, it's part of the correct abbreviation reading and also the correct numbers reading.

Identification of the word class was done using last three phonemes of words. This method has a success rate 90%. Because our goal is a perfect speech synthesizer, the following steps were implemented to increase the success rate [7]:

- Determination of word class according suffix. All 3-phonemes suffixes are stored in vocabulary. In case word class is ambiguously defined using this vocabulary, 4-phonemes suffix should be used. Even if 4-phonemes are not sufficient, 5-phonemes suffix is used. Separate vocabularies for 4 and 5 phonemes suffixes are used.
- The other possibility is to determine word class from context. Basic assumption is that the word class of other words in sentence is known. Rules from first point are used to define word classes. For words where the word class is not defined or the word can be in more word classes, word class is defined from the context.

In ideal case the synthesis process should determine all parameters for words like word class, gender, case and number. So the process of reading abbreviations or numbers is correct in any type of sentence.

3.4. Prosody modification module

The synthesized speech usually sounds artificial and unnatural. The main goal of post-processing is to modify prosody (e.g. to set frequency, energy and speech rate) in such a way that it is indistinguishable from human speech.

The Slovak language has several types of intonation in sentences – for example three types of questions: question type Y –expected answer yes or no, type R – this question contains more sentences connected with word whether and type Z – all other types. For all other sentences there is only one type. In case a comma is detected, we differentiate these types of compound sentences: type V – the first sentence has upward melody and the second one decreasing melody, type O – all sentences have decreasing melody.

Parameters such as fundamental frequency, speech rate, rhythm and accent also have rather large influence on intonation. The change of intonation requires [7]:

- Change of fundamental frequency
- Change of speech rate
- Change of energy
- Change of prosody of whole sentence according sentence type

There are more ways how to change listed parameters: PSOLA, TD-PSOLA, MBROLA [8], GNM [9][10] or sinusoidal models [10]. In our work we use sinusoidal models, specifically harmonic plus noise model (HNM). The results are presented in chapter 4.

3.5. Improvements of speech synthesis and learning methods

Another way how to improve speech synthesis is a process of learning. The learning mechanism is proposed in more levels for all modules and will be explained in more detail using examples for the specific modules. Possible levels are [7]:

- Customized speech synthesizer interface ensuring that a user can easily manually correct wrong synthesis (e.g. phonetic transcription or transcription of abbreviation - in case the abbreviation in glossary has different meaning or it's not in glossary, etc). The user can retype it like a text or in SAMPA – depending on synthesizer's interface. The changed word is stored in a temporary glossary. After certain time or sufficient number of corrections new rules are created (for phonetic transcription, prosody, etc.). Manual correction is the simplest type of learning mechanism.
- The next level is that system offers all correction possibilities for the wrong word. The explanation of this case for phonetic transcription is as follows: User marks wrong word. The synthesizer detects critical phoneme or phonemes in this word (such as phoneme which can be pronounced in more ways. In Slovak language for example ‚t' can be pronounced as ‚t', ‚t' or ‚d') and offers all possibilities based on LTS rules. The than user chooses the correct form and can hear synthesized text again. In case any of offered types of transcription are correct, the user still can retype it manually (the procedure above).
- The other option is very similar previous one, but it's more intelligent. The system of speech synthesizer offers all correction possibilities for wrong word. Possibilities are again based on LTS rules plus critical phoneme or phonemes are replaced by each phoneme from alphabet. These possibilities are sorted by their probability. User chooses the correct form and can hear synthesized text again. In case no offered types of transcription are correct the user can retype the text manually (the procedure above). We provide an example for better understanding:. We've got word ‚teraz'(it means now in English). Critical phonemes are detected: ‚t' and ‚z'. All possibilities are listed, in the begining ‚t' is replaced by ‚t', ‚t' or ‚d', than ‚z' is replaced by ‚s'. Than ‚t' is replaced by all phonemes from alphabet starting with the ones with the higher probabilityAcoustic modification. This method requires a perfect speech recognizer and we assume to have one. User hears synthesized text and in case of a mistakes reads the word or the whole sentence. The recognizer recognizes this new word or sentence and replaces the wrong one in synthesized text. The synthesizer compares the glossary and the temporary glossary/list of exceptions and if the correction is not there, it's automatically added into the temporary glossary. In this particular case it is useful to know more about users – we call it user quality. It can happen that different users have different pronunciation based on the part of country they came from (some of which are not correct). This has to be considering in the process of new rules creation.
- Correction of multiple words in synthesized text. In case one word is corrected, the synthesizer marks the same word/words in text and offers the user to change it automatically. User can choose to correct only some of them. For example in case of wrong word class, user corrects one word, the others are in correct form but all are marked. User chooses not to correct the others.

The process of correction is only one part of the learning process. The other part is learning of the speech synthesizer. All corrections are stored in temporary glossary. Our proposal is to store both, wrong and corrected word together with information about user who made correction. If there is enough data in the temporary glossary, new rules (for example new LTS rules) are automatically created or the word is shifted into the list of exceptions. It depends on program theshold whether to create a new rule or put the word into the list.

There are two possibilities to create a rule: manually or automatically. Both methods have its advantages and disadvantages. During a manual process all corrections are checked, so the wrong ones are dismissed and won't be used. New rules are created only from correct corrections or the word is

shifted to list of exceptions. So, theoretically, there should not be any mistakes in new rules or new words in the list. However in case of automatic process the words are not double-checked. There is a set of thresholds when to create new rule, when to put a word into the list of exceptions and when to leave it in temporary glossary till next rules creation. The automatic process is of course much faster and more mistake-prone than the manual one. The process of new rules creation should be periodically repeated [7].

4. Prosody modification using HNM

A HNM model was first presented by Ioannis Stylianou in [11]. HNM assumes that the speech signal is composed of a harmonic and a noise part. The harmonic part responds to the quasi-periodic 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 lower band of the spectrum (below Fm) is assumed to be represented solely by harmonics while the upper band (above Fm) is represented by a modulated noise component. While these assumptions are clearly not-valid from a speech production point of view they are useful from a perception point of view: they lead to a simple model for speech which provides high-quality synthesis and modifications of the speech signal [12].

The speech signal s(t) can thus be obtained as a sum of the harmonic and the noise part

$$s(t) = h(t) + n(t) \tag{1}$$

The harmonic part contains only harmonic multiplications of fundamental frequency. The signal is represented as a sum of sinusoids with corresponding frequencies, amplitudes and phases:

$$s_h(t) = \sum_{k=-L(t)}^{L(t)} Ak(t) e^{jk\omega_{0(t)}t} \tag{2}$$

where L(t) is the number of the harmonics, ω0(t) is the fundamental frequency and Ak(t) is the amplitude of the kth harmonic. Ak(t) can take on one of the following forms:

$$A_{k(t)} = a_k(t_i) \tag{3}$$
$$A_{k(t)} = a_k(t_i) + tb_k(t_i)$$
$$A_{k(t)} = a_k(t_i) + tc_k(t_i) + t^2 d_k(t_i),$$

where $a_k(t_i), b_k(t_i), c_k(t_i), d_k(t_i)$ are complex numbers. We use the simplest first form, called HNM1, with constant amplitudes described before.

The noise part n(t) can be modeled in two different ways. The first way is by coding spectral envelope using AR filter. The second way how to describe the spectral envelope is using the same method as for harmonic part. Since the noise part has no fundamental frequency, the F0 is set to 100 Hz. The phases of sinusoids are set randomly because the noise is a stochastic signal.

To change prosody using sinusoidal model, one has to know the relation between sinusoidal parameters for different tones. Analysis of human voice is more complicated than analysis of general audio signal like musical instruments, since the human voice is more variable. We analyzed different tones of all vowels. The first finding was that differences between the average and actual values are minimal, thus actual value can be replaced by average one. This will simplify further analysis and calculations. The next finding was that each vowel has its own dominant sinusoid, for example for A it's the first and the seventh sinusoid, for E it's the first and the third one, etc. We compared sinusoid amplitudes of these vowels at 220Hz and 440Hz. All amplitudes are more or less constant for each sinusoid. Based on this we can assume that the dependency is almost linear. The results are shown on **Chyba! Nenašiel sa žiaden zdroj odkazov.**and Figure 3. The adjustment of intonation (fundamental frequency and speech rate) by a mere 30% translates a normal synthesized sentence to a question. In case of bigger change the speech starts to be unnatural. It means that it's not enough to change only frequency of sinusoids and frame size but also phase, however this need to be further analyzed [7].

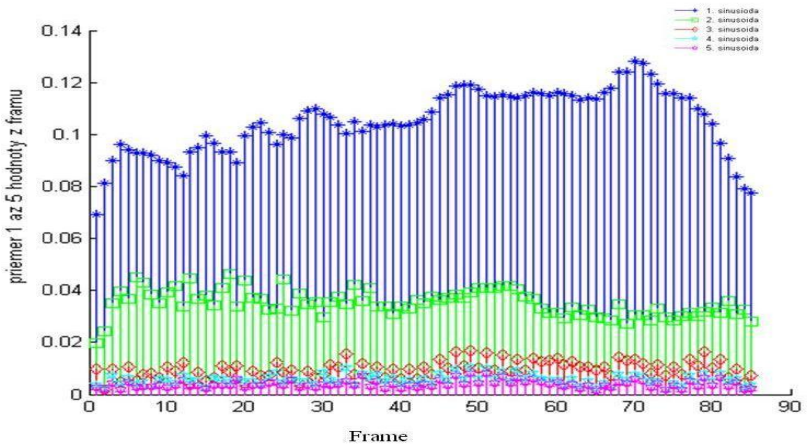

Figure 2. Amplitudes of the 1st-5th sinusoid, phoneme A 220Hz

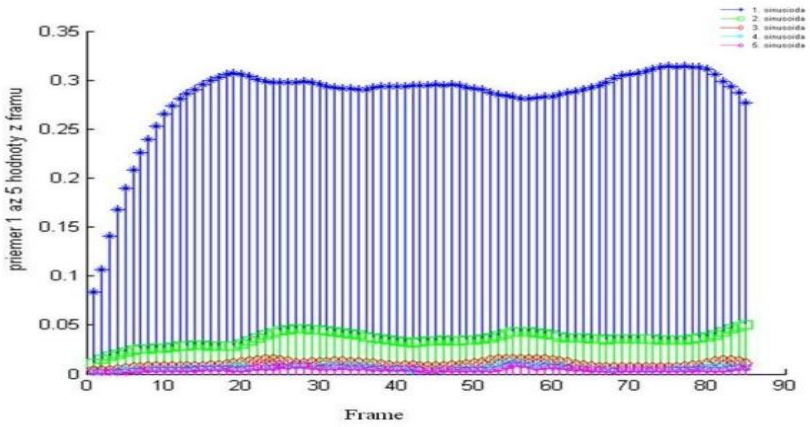

Figure 3. Amplitudes of the 1st-5th sinusoid, phoneme A 440Hz

5. Conclusion

The above work is focused on a design of an intelligent speech synthesizer and on prosody modification of Slovak language.

The synthesizer architecture is very flexible and allows each module to contribute or not to contribute to synthesis process depending on???dopisat. It is independent on the type of synthesizer (diphone, HMM) and language. The learning mechanism can by implemented for each module (phonetic transcription, phonetic transcription of abbreviations, word class identification, and prosody modification module) in various levels of intelligence:

- the lowest level - manual correction of wrong words,
- the upper level - the synthesizer offers list of possible correction,
- in advanced version this list is sorted by likelihood of appearance,
- the most advanced proposed level of learning is when user just talks to the synthesizer and all wrong words are corrected using speech recognition.

The last, but not least part of learning mechanism is processing of new data coming out from user's corrections. There are two ways how to process data: manually or automatically. The first case is time consuming but mistake free, the second one is much faster and more mistake-prone. The result of both cases are new rules and extended version of list of exceptions.

Prosody modification was realized using sinusoidal models. The main advantage of these models is their parametric form which allows a very flexible signal change in both time and frequency domain. In our case HNM model was used. Based on our analysis we were able to adjust intonation (fundamental frequency and speech rate) of normal synthesized sentence to a question.To achieve better results more analysis of relation between human voice and sinusoidal parameters is needed.

6. Acknowledgment

This work has been supported by the projects VEGA 1/0718/09 and FP7-ICT-2011-7 HBB-Next.

7. References

[1] L. Comerford, D. Frank, P. Gopalakrishnan, R. Gopinath, J. Sedivy, "The IBM Personal Speech Assistant, International Conference on Acoustics", Speech, and Signal Processing, Vol. 1, pp. 1-4, Utah, 2001.

[2] V. Gaudissart, S. Ferreira, C. Thillou, "Mobile Reading Assistant for Blind People", Proc. of the 9th SPECOM 2004, St. Petersburg, Russia, 2004.

[3] S. Dusan, G.J. Gadbois, J. Flanagan, "Multimodal Interaction on PDA's Integrating Speech and Pen Inputs", EUROSPEECH, pp. 2225-2228, Geneva, Switzerland, 2003.

[4] I. Kondratova, "Speech-enabled Mobile Field Applications, Internet and Multimedia Systems and Applications", IMSA 2004, Hawaii, USA, 2004.

[5] H. Roessler, et al., "Multimodal interaction for mobile environments". In Proceeding of International Workshop on Information Presentation and Natural Multimodal Dialogue, Verona, Italy, 2001.

[6] M. Johnston, S. Bangalore, G. Vasireddy, "Multimodal Access To City Help", in Workshop on Automatic Speech Recognition and Understanding, Madonna di Campiglio, Italy, 2001

[7] R. Rybárová, "Learning Methods for Speech Synthesis", Phd thesis, Department of telecommunication, FEI STU, Bratislava, 2010

[8] M. Turi-Nagy, "Utilization of sinusoidal models for audio signal processing", Phd thesis, Department of telecommunication, FEI STU, Bratislava, 2006

[9] M. Abe, S. Nakamura, K. Shikano, H. Kuwabara, "Voice Conversion Through Vector Quantization", J. Acoust. Soc. Jpn., vol. E-11, pp. 71-77, March 1990

[10] Y. Stylianou, O. Capé, E. Moulines, "Continuous Probabilistic Transform for Voice Conversion", IEEE Trans. On Speech and Audio Processing, vol. 6, no. 2, March 1998

[11] Y. Stylianou, "Harmonic plus Noise Models for Speech, combined with Statistical Methods, for Speech and Speaker Modification", PhD thesis, Ecole Nationale Sup`erieure des T´el´ecommunications, January 1999

[12] Y. Stylianou, "Applying the Harmonic plus Noise Model in Concatenative Speech Synthesis" In: Speech and Audio Processing, IEEE Transactions on, Jan 2001, pp.21–29, ISSN: 1063-6676