



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 662 • Publication 662

Kimmo Pärssinen

Multilingual Text-to-Speech System for Mobile Devices: Development and Applications



Tampereen teknillinen yliopisto. Julkaisu 662
Tampere University of Technology. Publication 662

Kimmo Pärssinen

Multilingual Text-to-Speech System for Mobile Devices: Development and Applications

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 27th of April 2007, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2007

ISBN 978-952-15-1763-1 (printed)
ISBN 978-952-15-1751-8 (PDF)
ISSN 1459-2045

Abstract

A multilingual text-to-speech system is different from a collection of language-specific synthesizers in the sense that it applies the same procedures and techniques to all languages it supports. Ideally, all language specific information and data should be stored in data tables and structures, and all algorithms should be shared by all languages. However, in practice this is relatively hard to achieve since many languages have quite different requirements for synthesis techniques and e.g. for text analysis and it is not straightforward to extend a method that is suitable for one language to new languages. Therefore, multilinguality in text-to-speech presents a number of technical challenges when designing a new system.

One of the main problems in the current text-to-speech systems is the time consuming internationalization process of the synthesis technology. Development requires knowledge about the human speech production and about the languages being developed. The development, implementation and integration work of a fully functional system requires multidisciplinary skills, such as signal processing, language processing and phonetics as well as software programming. Therefore, it is important to separate the language creation from the actual speech synthesis engine development. This thesis presents methods and techniques to improve the language development process in a speech synthesis system. The main idea is to separate the language independent synthesis engine and the language specific data and also provide a framework and tools, including an integrated development environment that can be used to ease the language creation process.

In multilingual text-to-speech, common algorithms and techniques should be applicable for multiple languages. A multilingual rule-based number expansion framework is proposed in the thesis. The framework is also extended to cover additional text normalization tasks. The thesis also presents a text-to-speech framework that has been successfully localized for over 40 languages. The system consists of a language independent synthesizer, a rule interpreter and a data configurable prosody model and language specific data that is used to control the speech synthesis. The introduced text-to-speech system is especially suitable for devices having limited memory resources, such as mobile phones.

The size of the synthesizer increases every time a new language is added. Furthermore, the most memory intensive parts of the whole text-to-speech system are

the ones which contain the language specific information. Such components are for example, lexicons and, in the case of concatenative speech synthesis, speech databases. For devices having limited memory resources, support for multiple languages is a major design and implementation challenge. The thesis presents a novel technique to reduce memory consumption by using an existing synthesis language to approximate a new language on a phonetic level. The presented technique can also be useful if the language portfolio has to be rapidly increased.

The last part of this thesis discusses the application of a text-to-speech system as part of the voice user interface. Moreover, the role of the automatic speech recognition system in some applications is also briefly covered. A preliminary usability study and evaluation of using a concatenative text-to-speech system to read text messages is presented. The synthesis quality of the system is found to be suitable for reading text messages. Furthermore, text-to-speech can be especially useful in situations where eyes-free operation of the device is needed.

Preface

First and foremost I would like to thank my thesis advisor and Head of Institute of Signal Processing in Tampere University of Technology, Prof. Moncef Gabbouj, for his help and guidance. I would also like to thank Dr. Panu Somervuo and Dr. Martti Vainio for their valuable comments and advice. Furthermore, I would like to thank Marko Moberg for his contribution to our research and team work. I also want to thank Dr. Olli Viikki for reminding me to continue and finish what I have started.

This thesis has been mainly carried out during the years 2003-2006. During this time I've been working in a research group at Nokia Technology Platforms focusing on speech synthesis and automatic speech recognition.

I would also like to express my gratitude to my friends. Finally, I want to say special thanks to my parents and my sister for their support throughout my life and studies.

Tampere, April 2007

Kimmo Pärssinen

Contents

Abstract	i
Preface	iii
List of Publications	vii
List of Figures	ix
List of Abbreviations	xi
1 Introduction	1
1.1 Organization of the Thesis	3
1.2 Summary of the Publications	3
1.2.1 The Author’s Contribution to the Publications	4
2 Human Speech Production and Phonetics	7
2.1 Properties of Speech	7
2.1.1 Acoustic Level	8
2.1.2 Phonetic and Phonological Level	11
2.2 Higher-level Linguistic Descriptions	15
3 Text-to-Speech and System Overview	17
3.1 Text-to-Speech System	19
3.1.1 Text Analysis and Prosody Modeling	21
3.1.2 Synthesis Stage	22
3.2 Automatic Speech Recognition System	23
4 Speech Synthesis Techniques	27
4.1 Concatenative Synthesis	27
4.1.1 Unit Selection Synthesis	28
4.1.2 Diphone Synthesis	30
4.2 Formant Synthesis	31
4.3 Other Synthesis Techniques	33

4.3.1	Articulatory Synthesis	34
4.3.2	Linear Predictive Based Methods	34
4.3.3	Hidden Markov Model Based Synthesis	35
5	Multilingual Text-to-Speech	39
5.1	Multilingual Text-to-Speech System	39
5.1.1	Multilingual Text-to-Speech System Framework	41
5.2	Text Analysis	42
5.2.1	Sentence Segmentation and Tokenization	42
5.2.2	Handling Non-standard Words	43
5.2.3	Morphological Analysis	44
5.2.4	Word Class Assignment and Prosodic Phrasing	44
5.3	Grapheme-to-Phoneme Conversion	46
5.4	Prosody Modeling	47
5.4.1	Fundamental Frequency Contour	47
6	Text-to-Speech in Voice User Interface	51
6.1	Voice User Interface	52
6.1.1	Usability Testing	52
6.2	TTS in Voice User Interface	53
6.2.1	Applications of Text-to-Speech and Automatic Speech Recognition	55
7	Conclusions	57
7.1	Some Methods for Multilingual Text-to-Speech in Mobile Devices	58
7.2	Text-to-Speech in Voice User Interface	59
	Bibliography	61
	Publications	73

List of Publications

The thesis is based on the following publications. In the text these publications are referred to as Publication 1, Publication 2, etc.

1. Moberg, M., **Pärssinen, K.** "Comparing CART and Fujisaki Intonation Models for Synthesis of US-English Names", *Proceedings of Speech Prosody 2004*, pp. 439-442, 23-26 March, Nara, Japan.
2. Moberg, M., **Pärssinen, K.** "Cross-Lingual Phoneme Mapping for Multilingual Synthesis Systems", *Proceedings of International Conference on Spoken Language Processing 2004*, pp. 1029-1032, 17-21 October, Jeju Island, Korea.
3. Moberg, M., **Pärssinen, K.** "Integrated Development Environment for a Multilingual Data Configurable Synthesis System", *Proceedings of International Conference of Speech and Computer 2005*, pp. 155-158, 17-19 October, Patras, Greece.
4. **Pärssinen, K.**, Moberg, M. "Multilingual Data Configurable Text-to-Speech System for Embedded Devices", *Proceedings of Multiling 2006*, 9-11 April, Stellenbosch, South Africa.
5. **Pärssinen, K.**, Moberg, M. "Evaluation of Perceptual Quality of Control Point Reduction in Rule-Based Synthesis", *Proceedings of International Conference on Spoken Language Processing 2006*, pp. 2070-2073, 17-21 September, Pittsburgh, Pennsylvania.
6. Moberg, M., **Pärssinen, K.** "Multilingual Rule-Based Approach to Number Expansion: Framework, Extensions and Application", *To appear in International Journal of Speech Technology (accepted 2006)*, Springer.
7. **Pärssinen, K.**, Moberg, M., Gabbouj, M. "Reading Text Messages Using a Text-to-Speech System in Nokia Series 60 Mobile Phones: Usability Study and Application", *Technical Report, Tampere University of Technology Report 2006:3*, Tampere, Finland.

8. **Pärssinen, K.**, Salmela, P., Harju, M., Kiss, I. "Comparing Jacobian Adaptation with Cepstral Mean Normalization and Parallel Model Combination", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2002*, Vol. 1, pp. 193-196, 12-17 May, Orlando, Florida.

The author has also published the following two publications related to this thesis. In the text, these publications are referred to as Publication A and Publication B.

- A **Pärssinen, K.**, Moberg, M., Harju, M., Viikki, O. "Development Challenges of Multilingual Text-to-Speech Systems", *W3C Internationalizing W3C's Speech Synthesis Markup Language, Workshop II*, 30-31 May, 2006, Heraklion, Greece.
- B Moberg, M., **Pärssinen, K.** "Using Text-to-Speech in Mobile Phones", *Proceedings of The Phonetics Symposium 2006*, pp. 125-133, 30-31 August, Helsinki, Finland.

List of Figures

1.1	Phases of a text-to-speech system	2
2.1	The human vocal organs	9
2.2	Two cycles of glottal pulse waveform	10
2.3	Spectrogram of the word "appointment"	11
2.4	A part of the waveform of the sound [ə]	12
3.1	Two-stage model of a text-to-speech system	21
3.2	Subword unit based speech recognition system	24
4.1	Basic structure of cascade formant synthesizer	31
4.2	Basic structure of parallel formant synthesizer	32
4.3	Block diagram of Klatt88 formant synthesizer	33
4.4	HMM-based speech synthesis system	36
5.1	Modular text-to-speech system	40

List of Abbreviations

ASR	Automatic Speech Recognition
CART	Classification and Regression Tree
CELP	Code Excited Linear Prediction
CMN	Cepstral Mean Normalization
CTS	Concept-to-Speech
DRT	Diagnostic Rhyme Test
FS	Feature Structure
FST	Finite-State Transducer
G2P	Grapheme-to-Phoneme
HCI	Human Computer Interaction
HMM	Hidden Markov Model
IPA	International Phonetic Alphabet
LPC	Linear Predictive Coding
MAP	Maximum a Posteriori
MLDS	Multi-Level Data Structure
MLLR	Maximum Likelihood Linear Regression
MLPC	Multi-pulse Linear Prediction Coding
MLSA	Mel Log Spectrum Approximation
ML-TTS	Multilingual Text-to-Speech
MOS	Mean Opinion Score

MRT Modified Rhyme Test

MSD-HMM Multispace Probability Distribution Hidden Markov Model

PI Physical Impairment

POS Part-of-Speech

PSOLA Pitch Synchronous Overlap and Add

RELP Residual Excited Linear Prediction

SSML Speech Synthesis Markup Language

SUS Semantically Unpredictable Sentence

TOBI Tones and Break Indices

TTS Text-to-Speech

VUI Voice User Interface

WLP Warped Linear Prediction

WOZ Wizard of Oz

Chapter 1

Introduction

Nowadays devices such as personal computers, mobile phones and even other domestic appliances have become more complex and automated. Therefore, interaction between humans and computers also becomes more demanding setting new requirements for Human Computer Interaction (HCI). Human computer interaction is a multidisciplinary area with various fields involving computer science, psychology, anthropology, education, design and different fields of engineering, which reflects the complicated nature of an individual's interaction with a computer. This includes factors such as an understanding of the user and the task the user wants to perform with the system, understanding of the tools and techniques that are needed to achieve this and also understanding the software engineering tools.

In most of the devices requiring interaction with the user the interaction still takes place in a traditional way taking advantage of displays and keyboards for transferring the information between the user and the system. However, the development in speech technology, both in automatic speech recognition and text-to-speech side, has made it possible to consider alternative ways of interaction between human beings and computers and other devices having computer-like capabilities. Speech is the most natural way for humans to communicate and it could also be used as a mode of communication between human beings and computers. This kind of modality in the user interface sets many demands on the device. The system should be able to understand at least the key parts of the user's speech and also be able to generate speech output for the user in order to enable interaction.

Designing a voice user interface requires the use of Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) techniques in order to make the communication between users and devices possible. ASR is needed to give the machine the ability to recognize the user's commands. A TTS system on the other hand is required to be able to provide speech output to the user of the system instead of displaying the information on the device. This thesis discusses some techniques and methods for multilingual text-to-speech and also presents some applications

taking advantage of TTS in the user interface.

The goal of a text-to-speech system is to automatically produce speech output from new, arbitrary sentences. The text-to-speech synthesis procedure consists of two main phases. The first is text analysis, in which the input text is transcribed into a phonetic or some other appropriate representation, and the second is the actual generation of speech waveforms, in which the acoustic output is produced from the information obtained from the first phase. A simplified version of the synthesis procedure is presented in Figure 1.1.

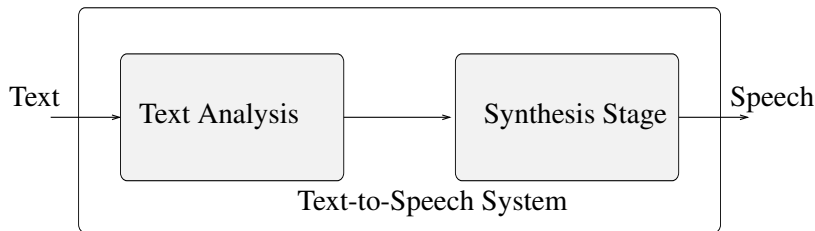


Figure 1.1: Phases of a text-to-speech system

Mobile devices usually have a rather small display and they are also often used in situations in which the user is not able to pay much attention to the screen. Therefore, using voice user interfaces in such devices can provide several advantages compared to the traditional input and output methods relying on keyboards and displays. However, applying speech technologies, such as TTS, in mass produced mobile devices introduces some restrictions and requirements to the system and technology used. For example, a memory consumption of several megabytes is rarely acceptable, and for that reason support for multiple languages on a single device with a limited memory resources is a major challenge. On the other hand, the wide language support is considered to be important. If the language support is not wide enough, the technology easily remains a niche feature that is not widely used. It is also important to be able to provide all users with access to the voice user interface regardless of their native tongue.

Because of these requirements, a text-to-speech system should be able to provide relatively easy and rapid language development and configurability to be able to adjust in platforms having a small amount of memory and low computational resources. In this thesis methods to achieve these requirements are presented. The thesis also discusses the principles of applying a TTS system as part of the user interface, and applications taking advantage of an available speech synthesis system are introduced.

1.1 Organization of the Thesis

The thesis consists of eight publications and an introductory review of relevant speech processing and text-to-speech areas. The introductory part is organized into seven chapters. In Chapter 2, the fundamentals of human speech production and different properties of speech are presented. This introduction provides the background information to understand the transformations required when converting from text to speech. Chapter 3 gives an overview of a text-to-speech system. This chapter also includes a short introduction to the history of speech synthesis. Two main stages, namely the text analysis and synthesis stage of a TTS system, are presented and some main techniques are introduced. Also at the end of the chapter, a brief overview of automatic speech recognition is given and some adaptation techniques improving robustness are mentioned. An ASR system can be applied as a part of the voice user interface together with TTS.

Chapter 4 describes different speech synthesis techniques that can be used in the synthesis stage. The main emphasis is on formant synthesis and concatenative synthesis techniques. Other methods are also discussed. In Chapter 5, the language dependent part of a TTS system is discussed and the concept of a multilingual TTS system framework is addressed. In Chapter 5, different procedures and techniques included in the text analysis and prosody modeling are presented. The chapter also presents some techniques for quickly adding a new language to the TTS system.

Chapter 6 discusses the concept of a voice user interface and focuses on applying a TTS system as a part of a user interface. Some usability testing methods that are normally applied during the development and evaluation of different applications are also presented. Finally, Chapter 6 discusses some practical applications taking advantage of text-to-speech technology to provide voice interaction to the user. Finally, the conclusions are drawn in Chapter 7.

1.2 Summary of the Publications

In Publication 1 [75], two intonation models are compared against a reference created with natural intonation for synthesis of US English names. The models developed are direct classification and regression tree (CART) based pitch estimation and a simple superposition model where fundamental frequency contours are generated by overlaying multiple components of different types.

Publication 2 [76] describes a method for rapidly increasing the language portfolio of an existing TTS system with minimum effort and memory consumption. The method is referred to as cross-lingual phoneme mapping and it modifies the phonetic transcription of a new language by presenting it with the phoneme set supported by an existing TTS system.

In Publication 3 [77], an integrated development environment for the language

development of a formant synthesis based text-to-speech system is presented. The development environment is capable of supporting platform independent development of multiple synthesis languages using only data such as rules and parameters and gives the user possibility to tune the TTS system without a deep knowledge of the underlying implementation.

Publication 4 [93] presents a multilingual data configurable TTS framework especially suitable for devices and applications where a low memory footprint is required. In Publication 5 [92], the perceptual effect of reducing the number of control points for formant contours in rule-based synthesis is discussed and studied. Publication 6 [78] presents a novel method and a framework for multilingual rule-based number expansion. Possible extensions for normalizing also other non-standard words such as different abbreviations by applying the same rule-based framework are presented in Publication 6 [78].

Publication 7 [94] introduces a text message reader application for Nokia Series 60 mobile phones taking advantage of a state-of-the-art TTS system. The user interface design and immediate usability test of the application are also discussed in the publication. Finally, in Publication 8 [96], three different adaptation methods for noise robust automatic speech recognition are compared. These techniques could be applied for example in a speech recognition system that is part of a voice user interface. Furthermore, many of the algorithms that are applied in speech recognition can also be utilized in Hidden Markov Model (HMM) based speech synthesis. For example, the adaptation methods originally used in ASR have also been applied in HMM synthesis.

1.2.1 The Author's Contribution to the Publications

In Publication 1 [75], the author implemented the prosody models and tuned the model parameters. The author also did extensive background research and helped writing the publication. In Publication 2 [76], the author designed and implemented the cross-lingual phoneme mapping in the TTS framework. The author has also implemented some phoneme mappings and helped writing the publication. Publication 3 [77] was the result of collective efforts. The author helped and supervised implementing the user interface of the language development tool. The author also designed and implemented the multilingual TTS system that is used in the language development tool. Publication 4 [93] presents a multilingual text-to-speech system. The author designed and implemented the framework for language specific rules and data, including the conversion tools, syntax and grammar of the rules, and the interpreter of the rules. In Publication 5 [92], the author investigated different interpolation techniques and implemented them in the TTS system. The author also helped writing the publication. Publication 6 [78] presents a multilingual number expansion framework. The author implemented the framework including the extension and helped writing the publication. In Publication 7 [94],

the author was responsible for the user interface of the application, helped in the usability test and implemented the text-to-speech support in the application. The author also drafted the manuscript. Publication 8 [96] compares different adaptation methods for noise robust speech recognition. The author developed and implemented the modifications to the adaptation methods, performed the tests and solely wrote the publication.

Chapter 2

Human Speech Production and Phonetics

This chapter discusses the physical mechanisms behind human speech production and describes the general structure of the human articulatory system. Different aspects and properties of human speech are also introduced in this chapter. This includes the acoustic speech signal itself and the phonetic and higher level linguistic information describing human speech. The purpose of this chapter is to give the reader an understanding of how human speech is produced and how speech can be analyzed. The aim of this chapter is also to provide the background information on important concepts and terms related to speech synthesis discussed in the following chapters.

2.1 Properties of Speech

Speech is the most natural way for human beings to communicate. Speech is generated by human respiratory and articulatory systems that consist of different organs and muscles. Coordinated action of these speech production organs result in sound waves that propagate in the air. The information conveyed by speech can be analyzed in many ways. In general, the following levels of description are distinguished: acoustic, phonetic, phonological, morphological, syntactic, semantic and pragmatic levels. These levels are mainly related to the transformations required when converting text to speech. Furthermore, the information communicated in spoken language can be categorized as linguistic and paralinguistic. Whereas the verbal content, the actual meaning of the words, is considered as linguistic information, the paralinguistic channels contain information about the speaker. Paralinguistics is concerned with factors of how words are spoken, i.e. volume, intonation, speed, breathing, hesitation etc. [22, 111].

2.1.1 Acoustic Level

Speech is physically produced by a variation of the air pressure, caused and emitted by the articulatory system. Speech can be analyzed by first digitizing the speech waveform and then applying various digital signal processing operations in order to highlight its acoustic traits such as fundamental frequency, intensity and spectral energy distribution. Each acoustic trait is related to a perceptual quantity, in other words, pitch, loudness and timbre.

Speech signals have both deterministic and stochastic behavior. The stochastic nature of speech can be seen when the same word is pronounced several times. Even if the waveforms of different instances of the same word clearly share same characteristics, there are still differences in the waveforms. This randomness is caused by the fact that there are always small differences in the pronunciation of a certain word. This happens because it is impossible to control the articulatory system accurately enough to produce exactly the same waveform twice.

The deterministic behavior of speech is mainly due to short-time periodicity in voiced sounds. In addition of being periodic in a short-time scale, speech signals of voiced sounds also contain harmonic components. These harmonic components can be seen as peaks in the speech spectrum. Although voiced sounds are deterministic in their behavior, there is always a stochastic component present in speech waveforms. The prosody independent randomness can be observed as differences in the waveforms of the same sound pronounced several times and this stochastic component can be usually modeled as spectrally shaped white noise [41, 111].

Human Speech Production System

Speech can be described as the result of the coordinated action of a number of muscles. The respiratory organs provide the energy needed to produce speech sounds forcing an air flow in the trachea and through vocal cords. Vocal cords are composed of two contiguous membranes and the tension is controlled by the neighboring muscles. Vocal cords provide an aperture in the larynx, called glottis. The size of the glottis can be varied using the muscles in the larynx. In voiceless sounds the glottis is open, resulting from the abduction of the vocal folds, see Figure 2.1.

The glottis is open at between 60% and 95% of its maximal opening and the air flows relatively freely through the larynx. If the flow of air has a low volume-velocity (200-300 cc/sec) then the air flow is smooth [13]. This kind of airflow is termed nil-phonation. Nil-phonation is used for many voiceless speech sounds such as [f,s] in the English words "feet" and "seat".

On the other hand, if the volume-velocity is above the 300 cc/sec then turbulence will occur as the air flows through the glottis. This is termed breath and is found in the sound [h] and, in those languages that use them, in voiceless vowel sounds.

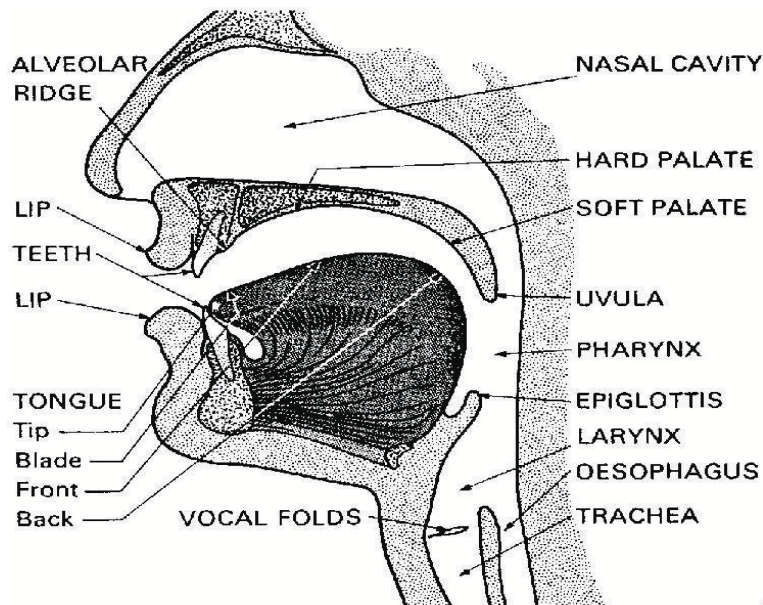


Figure 2.1: The human vocal organs

Voiced phonation is produced through the vibration of the vocal folds. The vibratory cycles of the vocal folds are repeated on average about 120 times per second for an adult male speaker, and about 220 times per second for an adult female. However, these frequencies may be increased or decreased by speakers when they raise and lower the pitch of their voice. Voiced phonation involves a pulsing action that expels short puffs of air very rapidly, and this action creates a humming noise at the larynx that adds to the perceptual salience of voiced sounds. For example, when comparing sustained [z] to [s] one can feel the vibration of the vocal folds within the larynx used with [z] by holding the fingers against one's Adam's apple. The vibration of the vocal folds is produced with the cooperation of both muscular and aerodynamic forces, with the balance of these forces altering subtly during the vibration cycle. The length of the cycle defines the fundamental frequency, F_0 , of the voiced sound. This ensues that the glottal waveform is composed of a sequence of pulses when transmitted through the vocal tract [22, 111].

Vocal Tract and Formants

The human vocal tract, shown in Figure 2.1, is basically composed of certain cavities and organs that are used to modify the excitation airflow from the lungs and larynx. Modifying the airflow is performed by changing the shape of these cavities and using articulatory organs to control the manner of articulation. The main cavities of which the vocal tract is composed are the pharyngeal cavity, the

oral cavity and the nasal cavity. Changing the shape of the cavities changes the properties of the vocal tract and results in different sounds. The most important articulatory organs are the tongue, lips and teeth.

The vocal tract is effectively an acoustic tube having certain physical properties. These physical properties can be varied by changing the shape of the vocal tract. This can be understood as a speech processing filter having a certain frequency response. The filter receives its excitation from the lungs and larynx and the excitation signal is then filtered by the filter representing the properties of the vocal tract. An example of the shape of the excitation is shown in Figure 2.2. In Figure 2.2 T_p marks the time during which the vocal folds are opening, T_n is the time between most excitation and vocal fold closure and T_0 is the time during which folds are open. Similarly, T is the total duration of the cycle and $T - T_0$ marks the time when the vocal folds are closed. The average rate of air flow is denoted as u_0 and u_{ac} is the maximum rate of airflow. Similarly, T_0/T is the percentage of time in each cycle that the vocal folds are open, often referred to as Open Quotient. Filtering the excitation signal with the filter representing the vocal tract modifies the spectral properties of the excitation signal resulting in the produced output speech signal. Because the human vocal tract can be seen as an

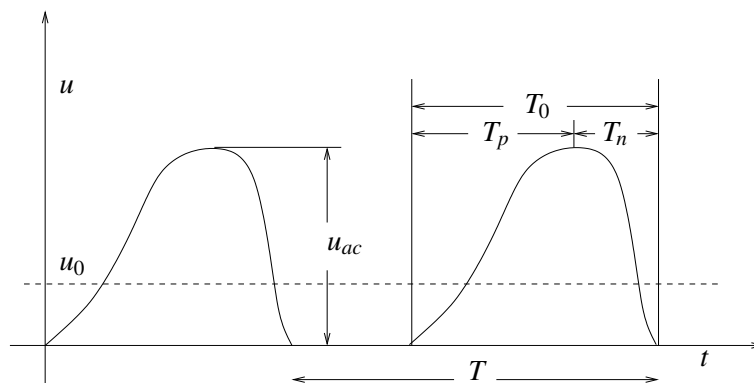


Figure 2.2: Two cycles of glottal pulse waveform

acoustic tube whose shape can be altered, then depending on the current shape it has a tendency to amplify signal components of certain frequencies. The vocal tract therefore acts as a resonator whose behavior can be described by a transfer function which is dependent on the shape of the vocal tract.

The resonator has certain frequencies which it tends to amplify over other frequencies. These resonant frequencies are called formants. The formants can be seen in the speech spectrum as visible peaks having certain amplitudes and bandwidths. Formants can be seen very clearly in a wide band spectrogram in Figure 2.3, where they are displayed as dark bands. The darker a formant is reproduced in the spectrogram, the stronger it is (the more energy there is, or the

more audible it is). The location of the formants in the spectrum is the main factor

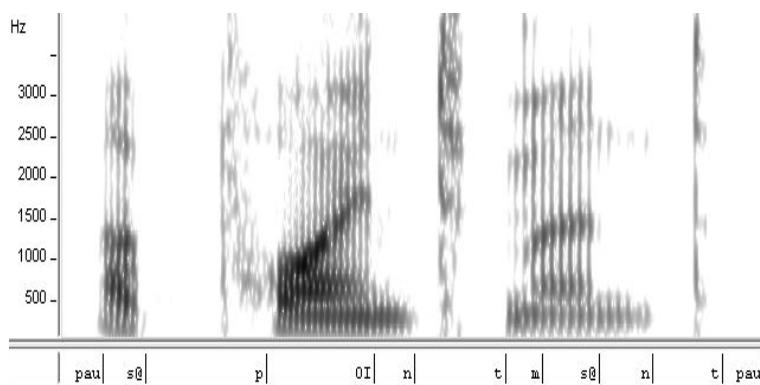


Figure 2.3: Spectrogram of the word "appointment"

distinguishing different vowels and some other voiced sounds from each other. The number of formants in human speech is not fixed but in general there are no more than six formants present in the speech spectrum. In order to properly distinguish between different voiced sounds only two or three formants are needed. Formants are often denoted by F1, F2, ..., Fn where the number indicates which formant has the lowest frequency and which one has the highest. F1 stands for the lowest frequency and for males it is usually located in the frequency range 300 Hz - 800 Hz. F2 in turn is located between frequencies 600 Hz - 2800 Hz, F3 between 1300 Hz - 3400 Hz. The higher formants have their own typical bands in a similar manner. Similarly, the fundamental frequency that is used to measure pitch is often denoted as F0 [111].

2.1.2 Phonetic and Phonological Level

Phonetics is a field of science that studies speech and the way speech signals are produced by the articulatory system. In phonetics, different sounds are usually marked by using International Phonetic Alphabet (IPA) symbols [44]. IPA is a system of phonetic notation devised by linguists and it is intended to provide a standardized, accurate and unique way of representing the sounds of any spoken language. Phonetics can be divided into three different categories depending on the emphasis of their research. These areas are articulatory phonetics, acoustic phonetics and auditory phonetics. Auditory phonetics studies the human speech perception and it will not be covered in this context.

Articulatory phonetics studies human speech articulation and its relation to the production of different sounds. In articulatory phonetics speech sounds are grouped into broad phonetic classes related to their articulation. Two parameters can be employed to look at how sounds are articulated: sound type (manner of articulation), and the sound position (place of articulation).

The weakest manner of articulation is called as a vowel. Vowels are characterized by two following properties. The first property is that vowels are voiced sounds. This means that vowels are sounds containing voicing, which is generated by the periodical vibration of the vocal folds in the larynx. From the signal processing point of view the speech signal segment containing a vowel sound has a periodical waveform and harmonic spectrum. An example of a part of the periodic waveform of vowel [ə] can be seen in Figure 2.4. The second main property of vowels is that during the pronunciation of a vowel sound the air can flow completely unobstructed through the vocal tract. Vowels can be further divided

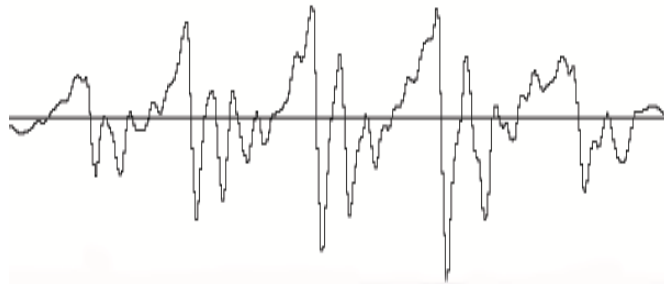


Figure 2.4: A part of the waveform of the sound [ə]

into monophthongs and diphthongs. A monophthong or "pure vowel" maintains the same articulatory positions throughout the sound, and so the perceived sound quality is steady. On the other hand, diphthongs are vowels where the tongue position moves during the production of the vowel, so that two different qualities of sound are perceived. This can also be seen in the spectrogram of sound [OI] shown in Figure 2.3, where the formants have two targets instead of one. The English language has 20 vowels out of which 12 are monophthongs and 8 diphthongs [4].

Whereas vowels are characterized by the fact that the vocal tract is open and not obstructing the airflow in any way, consonants cover the other cases. Consonants are sounds during which air is not allowed to flow freely through the vocal tract. Instead, the vocal tract provides a partial or complete obstruction of the airflow during the pronunciation of consonant sounds. Consonant sounds can be divided into different classes and sub-classes based on the manner and place of articulation. The two main classes of consonants are obstruents and sonorants.

Obstruents are consonant sounds during which the airflow in the vocal tract is obstructed partially or completely by the articulatory organs. Based on whether the obstruction is partial or if the vocal tract is completely blocked, obstruents can be further divided into fricatives and plosives or stops. Fricatives are pronounced with the articulators close together, but not so close as to block the airflow completely. This creates turbulence to the airflow, which has noise-like characteristics.

In English there are the following pairs of fricatives, both voiceless and voiced: [f, v, θ, ð, s, z, ʃ, ʒ] [4]. When the articulators are brought close together so that they make firm contact and the airflow in the oral cavity is completely blocked, the resultant manner of articulation is termed a stop. Plosives are formed by creating a complete closure somewhere in the upper vocal tract, for example by making a firm contact between the tip of the tongue and the alveolar ridge shown in Figure 2.1. The stoppage of the airflow is quite short, approximately 40-150 ms, but it results in a build-up of air pressure behind the closure so that, when the articulators part, the air bursts out with a typical popping sound. Plosives are found in all languages and in English there are six plosive sounds: [p, b, t, d, k, g] [4].

Sonorants form the second main class of consonant sounds. Sonorants are voiced consonant sounds during which the vocal tract is not constricted in a way that creates noise-like turbulence to the airflow. When considering the manner of articulation, sonorants can be further divided into nasals and approximants. Nasal sounds have a complete closure in the oral cavity but air is allowed to escape freely through the nasal cavity. This means that no build-up of air pressure occurs, and so these sounds do not have any plosion. English has three nasals: [m, n, ŋ]. Approximants, on the other hand, are sounds during which some obstruction to the airflow is created in the oral cavity by the lips and tongue. Approximants combine features from both vowel and consonant sounds and can be further classified into glides and liquids [4].

Consonant sounds can be further grouped by considering the place of articulation describing the place within the vocal tract where the articulators form a stricture. Some categories of the place of articulation are briefly described in this section. Articulations made with the two lips are called as bilabial. In these articulations the upper and lower lips are brought together. Labio-dental articulations are produced with the lower lip approximating to the underside of the upper front teeth. Dental fricatives occur in English as pronunciations of the "th" spellings. For example, the voiceless dental fricative, [θ] is the sound "th" in "thin" whereas its voiced counterpart [ð] is the sound "th" in "this". Alveolar sounds are all formed by raising the tip and/or blade of the tongue up to the alveolar ridge to form a contact or near contact. These sounds are common in English, for example alveolar plosive stops, [t, d] and a nasal [n]. Another category of place of articulation is palatal. In palatal sounds the front of the tongue dorsum is raised up to the hard palate. English has sound [j] for example the "y" in word "yes". Similarly, in velar sounds the back of the tongue dorsum is raised up to the soft palate (or velum). The velar plosives and nasal are found in English: [k, g, ŋ]. Uvular sounds are made by raising and retracting the back of the tongue to the uvula [4].

Acoustic phonetics on the other hand studies the properties of acoustic signals that are related to different sounds. Different articulations result in different kinds of acoustic signals that have their own characteristics to differentiate different sounds. These characteristics can be examined for example by studying the

waveform of the speech signal in the time domain or by analyzing its spectrum [4].

Phonological Level

Phonology forms a bridge between phonetics and higher level linguistics. In previous sections speech sounds have been described from an acoustic or physiological point of view as if it did not convey any information. Phonology introduces abstract linguistic units as opposed to speech units referred to as phonemes. A phoneme is the smallest unit in speech where substitution of one unit for another might make a distinction of meaning. For example, in English the words "pick" and "tick" differ in the initial phoneme. The same IPA symbols that can be used for representing the sounds of any spoken language are also applied to mark the phonemes. Therefore, in a way phonetics studies realizations of the abstract linguistic units i.e. phonemes [4, 41].

Other ways to group speech into different segments which can be interpreted as acoustically meaningful units have also been introduced. Such segments include for example diphones, triphones and demi-syllables. The reason to divide speech into more complex segments than sounds representing the realizations of the individual phonemes is to be able to include information about the transition effects between different sounds into the segments. These more complex sound segments are especially useful in the field of speech synthesis and recognition.

Diphones are speech segments containing the transients between two successive sound units. The diphone starts in the middle of the preceding sound and ends in the middle of the following sound unit. Diphones are able to retain the transition information which is found to be useful especially in concatenative speech synthesis. For example, in Figure 2.3 a diphone could start in the middle of [p] and end in the middle of [OI].

Demi-syllables resemble diphones closely. The difference is in the segment boundaries. While the start and end points of diphones are in general in the middle of the sound units in demi-syllables these points are located in the places where the transition between a consonant and a vowel ends and the steady state part of the vowel begins.

Triphones are speech segments that contain information about three successive sounds. The start point of a triphone is located in the middle of the first sound unit and the end point in the middle of the third sound unit [22]. In Figure 2.3 a triphone could start in the middle of [OI] and end in the middle of [t] containing totally two transitions compared to the diphone that contains one transition between different sound units.

Speech Prosody

Phonological description of speech is not complete if it does not include the duration, pitch and intensity of the phonemes as they convey additional information about speech. These three components are determined as prosody. In tonal languages, for example Chinese, the pattern of pitch within a word is needed to supplement knowledge of the phonemes to determine the identity of the word. In most European languages pitch, intensity and timing do not normally affect the identities of the words, but they provide useful information about what is being said.

Prosodic features can be used to indicate the mood of the speaker and to emphasize certain words. Prosody is also the main factor determining which syllables are stressed. The most important prosodic feature indicating stress and word prominence is pitch and especially the change of pitch on stressed syllables. Also, the sound duration increases on stressed syllables but there also are many other factors that affect the durations of sounds, e.g. their positions in a sentence and the neighboring sounds.

The correct prosody is also helpful in the interpretation of spoken language. Speech in which the prosody is significantly different from the one normally used by a native speaker can be very difficult to understand [4].

2.2 Higher-level Linguistic Descriptions

Speech can be further analyzed at morphological, syntactic, semantic and pragmatic levels. They form the higher level of speech analysis and are described here in brief.

In most European languages the lexical richness is several hundred thousand words. However, when studying a language it is easily noticed that although there are numerous words they often share some parts of their spelling as if they were formed from other smaller words or parts of words. Morphology is the part of linguistics that describes word forms as a function of a reduced set of meaningful units, called morphemes.

Morphemes are the minimum meaningful units of language. For example, the word "played" contains two morphemes, "play" and a morpheme to account for the past tense, "ed". Morphemes are abstract units that can appear in several forms in the words they affect. When there is a direct mapping between abstract morphemes and segments in the textual form of a word, these segments are referred to as morphs. Morphs can further be categorized into roots and affixes and the addition of common affixes can greatly increase the number of morphs in a word. A high proportion of words in languages such as English can be combined with prefixes and/or suffixes forming other words but the pronunciations of the derived words are related to the pronunciations of the root words [54].

Not all sequences of words listed in the lexicon of a language form a correct sentence. The list of permissible sentences, although infinite in natural languages, is restricted and defined by their syntax. Syntax should not be confused with the rules that are used to describe it and are organized into grammars. Grammars can also be used to describe the hierarchical structure of sentences, and the operation of finding the syntactic structure of a sentence with respect to a given grammar is called parsing.

Basically the syntax of the language can be described by using many different grammars depending on the part of speech categories, the rule definitions and the formalism chosen for the rules. Traditional grammars are only one of these rules and are not particularly well suited for automated and computerized text analysis as they assume prior knowledge and use of the language. In contrast to traditional grammars, formal grammars, used to describe computer languages, have also been applied in describing the syntax of natural languages. Formal grammars can be directly applied with computerized text analysis. They have also been extended with semantic features making them powerful tools for natural language processing.

Although the syntax of the language restricts the set of well-formed sentences it does not rule out sentences that have no real meaning at all. The study of word meanings and how they relate to each other is called lexical semantics. When semantic information is included in the lexicons the number of word classes increases rapidly and the rules describing the relationship between different words becomes complex. Usually, the distinction between semantics and syntax is not very clear and the syntactic descriptions are often semantically bearing. Semantic meaning is often viewed as context independent. In contrast to this, pragmatic meaning is defined to be context dependent. Everything that refers to the context and intentions of the speaker is related to pragmatics. Pragmatic analysis is much less developed compared to the semantic analysis of speech [41].

When developing a text-to-speech system all these different properties and levels of speech and human speech production should be taken into account. The task becomes especially demanding if the same text-to-speech system has to be able to support multiple languages, each having its own, very different characteristics. These challenges and possible solutions and techniques are considered in the following chapters.

Chapter 3

Text-to-Speech and System Overview

Speech synthesis or artificial speech has been studied for centuries. In this section a brief overview on the development of speech synthesis techniques is given as it may give useful information and help understand the principles behind the present TTS systems and how they have developed to their current form.

The earliest efforts made in speech synthesis were done over two hundred years ago by a Russian professor Christian Kratzenstein in St. Petersburg 1779 [25, 26]. He explained the physiological differences between five long vowels and also made an apparatus to produce them artificially. A few years later, in Vienna 1791, Wolfgang von Kempelen introduced his acoustic machine that was able to produce single sounds and some combinations [57, 101]. Both of these early speech synthesizers were mechanical machines trying to mimic the human speech production system. Research and experiments with mechanical and semi-electrical analogs of vocal system continued until 1960s. Mechanical and semi-electrical experiments conducted by known scientists e.g. Herman von Helmholtz and Charles Wheatstone are described in [25, 26, 101].

The first full text-to-speech system for English was developed by N. Umada and his colleagues in 1968 [57]. It was based on an articulatory model and included a syntactic analysis module with sophisticated heuristics. The speech quality of the system was relatively intelligible but monotonous and far from the quality of modern TTS systems. In mid 1960s the first speech synthesis experiments with Linear Predictive Coding (LPC) based methods were made. Linear prediction was first used in low-cost systems and its quality was quite poor compared to present systems. However, with some modifications the method has been successfully used in many present systems. In 1979 Allen, Hunnicutt and Klatt presented the MITalk TTS system [2]. Two years later Klatt introduced his famous Klattalk system which used a new and more sophisticated voicing source described in detail in [57]. The technology used in these systems forms the basis

of many synthesis systems such as DECtalk [37] and is useful in applications and devices where the memory footprint of the TTS system should be small.

Modern TTS systems involve quite sophisticated and complicated methods and algorithms. One of the methods applied recently in speech synthesis is based on concatenating small units of real human speech in order to form the synthesized speech output [41, 43]. These systems take advantage of a large speech database trying to find the optimal speech segments that would match the input text. Another method is based on the use of Hidden Markov Models (HMMs) to model the speech units and concatenating them accordingly, see for example [131]. Both of these techniques are presented in more detail in Chapter 4.

As the early speech synthesizers were mainly focused on the actual synthesizer part the biggest challenges nowadays are in the text analysis phase. The system should be able to extract meaningful information from any given text to be able to produce natural sounding synthesized speech output. One of the main problems modern TTS systems face is the quality of prosody of synthetic speech. When a human being reads aloud a certain text, one gathers contextual information over the course of the text. In real life the prosody of a particular sentence is often determined by information presented several sentences ago. However, current TTS systems are not able to utilize this kind of information well, and the synthetic speech still lacks the rhythm and other natural variations humans can produce naturally.

The term TTS is used to describe the process of converting given raw text into synthetic speech. Concept-to-Speech (CTS) is a term often used for speech synthesis where the input is not text, but rather a machine generated message. A TTS system usually comprises two main components, namely text analysis and speech generation. The text analysis part has to resolve the ambiguities inherent in written text and produce a clean linguistic representation of the sentence to be spoken. In CTS, the situation is different. There is no prior input text as such, but a system generates automatically some text that can be marked linguistically. These linguistic markings are usually such that they are very difficult to estimate from the input text. This generated linguistic information can then be used to improve the prosodic properties of the synthetic speech. This kind of CTS system can be used for example in dialog systems. The user could ask e.g. "Tell me what time it is" and the CTS system would generate the answer. Therefore, in the CTS case, there is not text ambiguity: the generator can annotate the text it produces with the information needed to guide the synthesis. The synthesis techniques that are applied in CTS systems have to be able to take advantage of the linguistic information marked in the text. This means that the synthesizer has to be able to modify the speech prosody accordingly. HMM synthesizers are found to be especially useful, since speech parameters can be easily altered by modifying HMM parameters by applying for example different adaptation techniques applied commonly in automatic speech recognition [115, 122].

In this chapter an overview of speech synthesis techniques is given. The chapter will also briefly discuss automatic speech recognition techniques which can be used when implementing a voice user interface taking advantage of a text-to-speech system and automatic speech recognition.

3.1 Text-to-Speech System

A text-to-speech system synthesizes speech from a given text. Although TTS is not yet able to replicate the quality of recorded human speech, it has improved greatly in recent years. There exist different synthesis technologies suitable for different applications. The systems differ for example by memory footprint, generality, language coverage, portability and speech quality. The memory footprint can be determined to consist of both static and dynamic memory consumption. The term generality indicates if the system is optimized for a certain specific domain. For example, a non-general system could have a limited vocabulary support and limitations in the length of spoken utterances. Furthermore, a TTS system can be multilingual or just monolingual. Portability describes how easy it is to transfer the system to another implementation platform or operating system [22, 108]. Finally, the quality of a TTS system is often determined by using the following four measures [18]:

- **Intelligibility:** How well the user can understand what is said. This is the most important quality factor of synthesized speech.
- **Naturalness:** How much the synthesized speech sounds like real human speech.
- **Accuracy:** Describes the correctness of what is synthesized. For example, making correct choice between "British" and "Best regards" when the input is "Br".
- **Listenability:** Describes how well the user is able to tolerate listening of the synthetic speech without fatigue.

Having defined these four categories it is clear that they are not independent from one another. For example serious errors in accuracy will lead to less intelligible speech, and this will be perceived as less natural and the listenability of the synthesized speech becomes worse.

Basically, there does not exist any simple metric that could be applied to any TTS system and which would reveal the overall quality of the system. One reason for this is that it is usually not very meaningful to assess TTS systems in isolation, but it is often more useful to evaluate them in different applications in which the system would be used in practice. Different applications have differing needs for

a TTS system. A synthesized voice that is appropriate for a system intended to entertain may not be the best for e.g. a mainstream application providing information to the user [91].

Intelligibility tests are concerned with the ability to identify what was spoken/synthesized. They are less concerned about the naturalness of the signal, although naturalness is related to and influences intelligibility. One way to evaluate intelligibility is the so called comprehension task in which the users are played back passages related to the TTS application, e.g. e-mails or text messages, and asked questions about the passage. For example, "What was the main point of the passage?" and so on. Other possible tests of measuring intelligibility are referred to as phonetic tasks that deal with identifying the precise sounds within a specific word that was synthesized. In the Diagnostic Rhyme Test (DRT) [42, 86], the intelligibility of word-initial consonants is tested by playing back pairs of words with a different first consonant. Another test is the Modified Rhyme Test (MRT) [42, 86] which is similar to the DRT, but includes tests for word-final intelligibility (e.g. bath vs. bass). Other phonetic tasks that are used for measuring intelligibility are for example the Standard Segmental Test, see [48], and the Cluster Identification Test [48]. All these tests require that subjects identify specific sounds within a signal. Transcription tasks are also used to measure the intelligibility of the synthesized speech. In the Semantically Unpredictable Sentence Test (SUS) [36] subjects are asked to transcribe sentences that have no inherent meaning or context, and therefore do not afford the possibility of deriving phonetic information from any source but the signal.

Naturalness of the synthetic speech can be assessed using various listening tests. One well known subjective method is the Mean Opinion Score (MOS) test. In the MOS test, listeners are asked to rate the speech quality of different systems, usually synthesizing the same set of utterances. Another possible test is, for example, the Forced-Choice Ranking, in which subjects are asked to rank the synthesized utterances [59].

Testing the accuracy of a TTS system is close to a truly objective test. In the test, one should collect sentences that contain multiple examples in multiple contexts of the types of text anomaly that the TTS system is likely to encounter, and simply mark the wording of the output as correct or incorrect.

Some synthesis techniques may for example be more natural or more intelligible and the goals of a synthesis system will often determine what approach is used. However at the general block diagram level all modern TTS systems regardless of the actual synthesis technology share the common basic high-level structure shown in Figure 3.1.

The generation of synthetic speech is often viewed as a two-stage analysis-synthesis process. The first part of this process involves analysis of the text to determine the underlying linguistic structure. The abstract linguistic description includes the phoneme sequence and other information such as the stress pattern

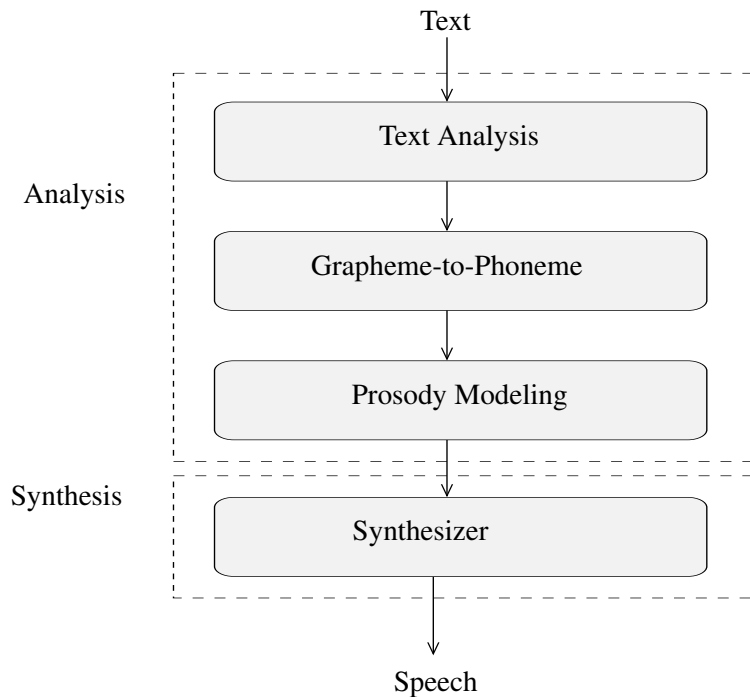


Figure 3.1: Two-stage model of a text-to-speech system

and syntactic structure, which can influence the way the text should be spoken. Furthermore, the analysis part also includes prosody modeling. The second part of the TTS system is responsible for generating synthetic speech from the linguistic description [41, 53].

The modular TTS system structure shown in Figure 3.1 is relatively straightforward to tailor a general TTS system for a specific application or even to a new language [41, 53].

3.1.1 Text Analysis and Prosody Modeling

Text consists of alphanumeric characters, blank spaces and possibly a variety of special characters. The first step in text analysis usually involves pre-processing of the input text which includes expanding numerals, abbreviations etc. and converting it to a sequence of words. The pre-processing module will also detect the instances of punctuation and other relevant formatting information. The following text normalization modules then convert the sequence of words into a linguistic description. An important part of these modules is to determine the pronunciation of individual words. Word pronunciation is normally obtained using some combination of a pronunciation dictionary and letter-to-sound rules. To reduce the size of pronunciation dictionaries, most TTS systems include some kind of

morphological analysis. This analysis module determines the root form of each word and therefore avoids including all derived forms in the dictionary. Usually, syntactic analysis is also required to be able to determine the pronunciation of certain words. Once pronunciations have been determined for individual words some adjustments are usually performed to incorporate phonetic effects occurring across word boundaries [22, 41].

In addition to determining the pronunciation of the word sequence, the text analysis module has to determine other relevant information, e.g., how the text should be spoken including phrasing, lexical stress and the pattern of accentuation of the different words, i.e., sentence-level stress. This information is then used when generating the prosody for the synthesized speech. Text analysis and prosody generation modules and different techniques applied there are described in more detail in Chapter 5.

3.1.2 Synthesis Stage

The final modules in the TTS system perform the speech sound generation based on the information obtained from the analysis part of the text-to-speech system. The synthesis stage is usually achieved by applying concatenative synthesis techniques, although formant synthesis is also used especially in devices in which low memory footprint and consumption are important factors. Another important synthesis technique suitable for devices having small amount of memory is HMM synthesis. One advantage of the HMM-based synthesis is also that compared to concatenative synthesis techniques, changing the prosodic and other properties of the synthesized speech is much easier, which improves the understandability, especially in noisy environments, see e.g. [72].

Concatenative synthesis is based on the concatenation of segments of recorded speech. One well known concatenative synthesis method is usually referred to as unit selection synthesis and it takes advantage of large speech databases of recorded human speech [43]. 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 systems may use a small amount of signal processing at the point of concatenation to smooth the waveform. In fact, the output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness often requires unit selection speech databases to be very large which makes them easily impractical for devices having only a small amount of memory.

Another well-known concatenative speech synthesis technique is referred to as diphone synthesis and it uses a minimal speech database containing all the diphones occurring in a given language [22]. At runtime, the target prosody of a sentence is superimposed on these minimal units by the means of digital signal

processing techniques. The quality of the resulting speech is generally not as good as that from unit selection but often more natural-sounding than the output of formant synthesizers [22].

A third concatenative synthesis technique is called domain-specific synthesis and it concatenates pre-recorded words and phrases to create complete utterances [22]. It is used in applications where the variety of texts the system will output is limited to a particular domain, such as transit schedule announcements or weather reports. The naturalness of these systems can potentially be very high because the variety of sentence types is limited and closely matches the prosody and intonation of the original recordings. However, because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases they have been pre-programmed with [22].

Formant synthesis does not use any human speech samples at runtime. Instead, the synthesized speech is created using an acoustic model [41]. Parameters such as fundamental frequency, formant frequencies and bandwidths, voicing, and noise levels are varied over time to create a waveform of artificial speech. Many systems based on formant synthesis techniques generate artificial, robotic-sounding speech, and the output would not be mistaken for human speech [41]. However, maximum naturalness is not always the main goal of a speech synthesis system, and formant synthesis systems have some advantages over concatenative systems. Formant synthesized speech can be very reliably intelligible, even at very high speed, avoiding the acoustic glitches that can often plague concatenative systems. High speed synthesized speech is often used by the visually impaired people for quickly navigating computers using a screen reader. Also, formant synthesizers are often smaller systems than concatenative ones because they do not have a database of speech samples. They can thus be used in embedded systems where memory space and processor power are often scarce as mentioned earlier [41].

3.2 Automatic Speech Recognition System

Today most automatic speech recognition systems utilize a statistical approach for speech recognition by applying Hidden Markov Models [97]. HMMs have been actively studied since 1970s and they provide a framework for ASR. This framework includes automatic training of the statistical parameters of the HMMs and also decoding algorithms to perform the speech recognition. Several techniques to improve the performance of the recognition systems have been introduced. These include for example context-dependent modeling, dynamic feature parameters, mixtures of Gaussian densities and different model adaptation techniques and feature normalization methods. Many of the algorithms that are applied in ASR can also be utilized in HMM-based speech synthesis. For example, context dependent modeling and various adaptation techniques for speaker and environment adapta-

tion have been utilized also in speech synthesis [72, 113].

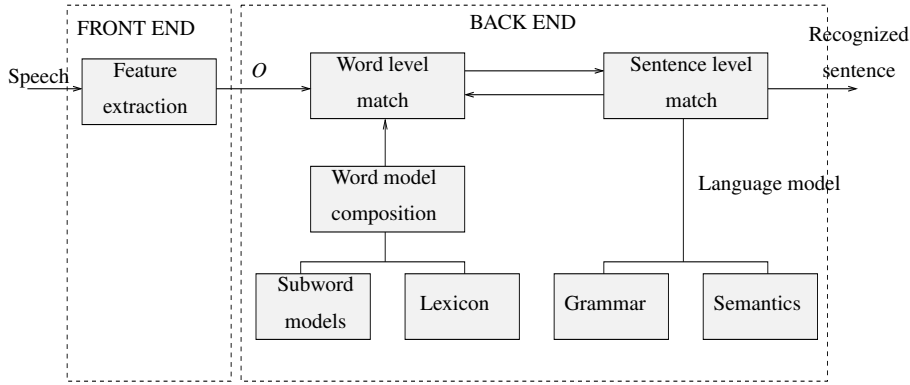


Figure 3.2: Subword unit based speech recognition system

Figure 3.2 shows a simplified block diagram of an automatic speech recognition system. In this system, two main blocks can be separated, namely the feature extraction unit, referred to as the front-end, and the unit performing the recognition, often referred to as the back-end. Although many different signal analysis techniques have been proposed for ASR, the most commonly used front-end is based on mel-frequency cepstral coefficients [97].

The back-end unit is a pattern classifier that tries to decide the correct order of the pre-trained models that compose the observed speech patterns. The output of the speech recognition system is the most likely word sequence [97]:

$$\hat{\lambda} = \arg \max_{1 \leq v \leq V} Prob(\lambda_v | O) = \arg \max_{1 \leq v \leq V} Prob(O | \lambda_v) Prob(\lambda_v) \quad (3.1)$$

where O is a sequence of feature vectors produced by the front-end and λ_v is the v^{th} acoustic model corresponding to a word or a sentence. To perform a word level match for the feature vectors, the word models are usually obtained by concatenating the subword models based on the rules determined in the lexicon. The recognition of sentences is based on the word recognition and the rules provided by the grammar that tells the possible words that can follow each other. The semantics consists of rules and other information that can be used to discard meaningless sentences [64]. Several techniques have been proposed to solve Equation 3.1. For example, $Prob(O | \lambda)$ can be solved by applying the Viterbi algorithm presented in [123] or using the Forward-Backward algorithm described e.g. in [97]. Another basic problem in ASR is that the model λ should be optimized to better match the observation sequence during training of the pattern classifier. One way to do this is to use iterative techniques such as the Baum-Welch algorithm [5]. This algorithm is a version of the expectation maximization algorithm and it is able to find the model parameters that are locally optimal in the maximum-likelihood sense [19].

By collecting a large enough speech database one can train an HMM-based speech recognition system which works reasonably well, if the testing conditions approximately correspond to the training data. However, regardless of how large the training database is, there will always be such speakers or noise conditions whose characteristics do not appear during the training process. In real-world conditions, both the environment and speech characteristics tend to vary continuously in time. It is therefore obvious that a static speech recognizer can only seldom meet the high requirements set for an ASR system.

Different methods have been developed to tackle the mismatch between the training data and the operating environment and the speaker. A dynamic ASR system allows its parameters to be adjusted so that the updated system better matches the present conditions, and if the adaptation criteria are meaningful, dynamic recognition systems are able to outperform ASR systems which make no use of adaptive methods. These methods can be split into two different groups [32].

The first class of methods attempts to modify the acoustic models used in the speech recognition stage in such a way that they better resemble the incoming speech patterns. These techniques include model adaptation methods, such as Maximum a Posteriori (MAP) Adaptation [35] and Maximum Likelihood Linear Regression (MLLR) [65] and for example Parallel Model Combination, see e.g. [33, 34]. These algorithms, such as MAP and MLLR, have also been successfully used in HMM-based speech synthesis [72, 113].

In the second class, the corrupted waveform can be pre-processed such that the resulting parameters are related closely to those of clean speech. The compensation can be based on the statistical information about the interfering noise source, e.g. noise subtraction or such feature representation that is robust against noise. Cepstral Mean Normalization (CMN) is by far the most popular feature normalization method in speech recognition. In its classical form [3], the average of each cepstral component is subtracted from the feature vectors representing speech. Knowing that the channel effects introduce a bias in the cepstral domain [31], CMN is an efficient method for compensating for the channel distortion, and similar methods can also be used in HMM speech synthesis [127]. In Publication 8 [96], three different environment adaptation methods were compared using an HMM-based recognition system. Several other adaptation methods have been presented, see for example [32].

Chapter 4

Speech Synthesis Techniques

The last stage in a TTS system is the actual synthesizer and it is responsible for producing the actual synthesized speech output. Several different synthesis techniques have been introduced and in this chapter concatenative synthesis, rule based synthesis and some other important synthesis techniques, such as HMM-based synthesis, are covered. In concatenative synthesis small units of real recorded speech are concatenated after each other to form the final output. The best concatenation based synthesizers are able to provide relatively natural sounding synthetic speech. However, the drawback of this synthesis technique is that since most of the contextual information of the speech units is embedded in the data, the database size increases, when different phonetic contexts have to be taken into account and stored in the database to be able to provide natural sounding speech.

Rule based or formant synthesizers are mainly favored by phoneticians and phonologists as they constitute a cognitive and generative approach of the phonation mechanism. Formant synthesizers also have a much smaller memory footprint and requirements than concatenative systems making them suitable for devices having small amount of memory. The drawback of rule based synthesis is that the sound is usually quite mechanical since the rules controlling the synthesis are very difficult to develop.

In HMM-based synthesis the speech spectrum and excitation parameters are modeled by context dependent HMMs and during speech synthesis the HMMs are concatenated according to the input text. HMM-based speech synthesizers are able to produce natural sounding speech and their memory requirements are also relatively small [22, 119].

4.1 Concatenative Synthesis

Connecting prerecorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. However, the drawback is that

the systems are usually limited to one voice and often require more memory capacity than other methods. One of the most important aspects of concatenative synthesis is to find the correct sound unit length. The selection is usually a trade-off between longer and shorter units. With longer units high naturalness, less concatenation points and good control of coarticulation are achieved but the amount of required units and memory is increased. With shorter units, less memory is needed but the sample collecting and labeling procedures become more difficult and complex. In present systems units that are commonly used are words, syllables, demisyllables, phonemes, diphones and sometimes triphones [53].

In this section two well known concatenative synthesis strategies are presented, namely the unit selection synthesis and diphone synthesis. Unit selection synthesis uses a large speech database with usually fixed unit size e.g. demisyllables. The basic principle is to collect speech units in different phonetic and prosodic contexts and find the best matching sequence of units given the textual input. Another concatenative synthesis technique presented is based on concatenation of diphones and it uses a minimal speech database containing all the diphones occurring in a given language [22].

4.1.1 Unit Selection Synthesis

The basic idea in the unit selection technique is that one is able to synthesize new naturally sounding utterances by selecting appropriate sub-words from a database of natural recorded speech. There are many conditions that have to be met before the unit selection system is able to work. In unit selection the system has to be able to decide which units it should use for synthesis to maximize the quality of the output speech in terms of intelligibility, naturalness and other criteria of quality presented in the beginning of Chapter 3. Therefore, the systems are trying to find the optimal sequence of units which minimizes the concatenation cost. Usually, the cost is split into a target cost that describes how close a database unit is to the desired unit and a continuity cost that describes how well two adjacently selected units join together. The target cost is calculated during runtime and it exploits only features that are computable from the text. Various features have been proposed in the literature typically encoding the phonetic, metrical structure and the prosodic context of the units. The continuity cost exploits all features of candidate units and it is generally computed as the Euclidean or Mahalanobis distance between spectral features representing boundary frames of the corresponding units. Determining continuity costs is usually computationally expensive, and therefore it would be desired to be able to calculate the continuity costs offline and store them into a lookup table. However, it is practically impossible to pre-compute and store all possible unit combination costs due to the high number of units, but since most of unit combinations are very rare, it has been shown that it is enough to store continuity costs for a small subset of the total database without sacrificing

the synthesis quality.

The overall unit selection process is designed to optimally minimize both of these costs. This can be expressed more formally using the following notation for the target cost C^t

$$C^t(t_i, u_i) = \sum_{j=1}^P w_j^t C_j^t(t_i, u_i), \quad (4.1)$$

which is the weighted sum of differences of relevant features. In Equation 4.1 C^t is defined as the target cost, t_i is the i^{th} target unit and u_i denotes the i^{th} unit in the unit database. Similarly w_j^t is the weight factor used for the j^{th} target cost C_j^t and P is the number of features compared. In addition, we can define continuity cost C^c as a weighted sum of features' differences between adjacent units. This can be expressed more formally as

$$C^c(u_{i-1}, u_i) = \sum_{k=1}^q w_k^c C_k^c(u_{i-1}, u_i). \quad (4.2)$$

In Equation 4.2 C^c is defined as the continuity cost, u_{i-1} is the $(i-1)^{\text{th}}$ unit and u_i denotes the i^{th} unit in the unit database. Similarly w_k^c is the weight factor used for the k^{th} continuity cost C_k^c and q is the number of different features compared. The weighted sum of these two costs has to be minimized in order to find the string of best matching units from the unit database [43].

Each unit u_i in the database is represented by a state in a state transition network and state occupancy costs are given by a measure of unit distortion i.e. target cost, and state transitions are given by a measure of the continuity distortion. The unit selection process resembles the Hidden Markov Model based automatic speech recognition, but instead of using probabilities as in ASR, unit selection applies cost functions. The unit selection algorithm selects from the database an optimal sequence of units by finding the path through the state transition network that minimizes the combined target and continuity costs [43]. For example, if the word to be synthesized is "appointment" and it is found in the unit database, the algorithm can choose the whole word (if it minimizes the total cost) instead of selecting individual units. The spectrogram of the word "appointment" is shown in Figure 2.3. By selecting longer sequences the system is able to reduce the number of segment concatenation points and it does not have to take care of how to smoothly combine say e.g. different diphones or triphones.

Speech Unit Database

Unit selection systems usually select from a finite set of units in the speech database and try to find the best path through the given set of units. When there are no examples of units that would be relatively close to the target units, the situation can be viewed either as lacking in the database coverage or that the desired sentence

to be synthesized is not in the domain of the TTS system. Therefore, to achieve good quality synthesis, the speech unit database should have a good unit coverage. In the simplest sense, this means recording more data from the speaker since with more data it is more likely that a database will contain a unit that is closer to the target unit and also likely to have a better continuity.

On the other hand, the problem of increasing the database size is that there will always be holes i.e. situations that there are no units that would be close to the target unit in the database. This is due to the phenomenon of relatively frequently occurring rare events in language [66]. In practice, this means that common events in a language are very common but there are so many rare events that they are also common. Also, covering for example all triphone contexts in even a few phrasal conditions is impractical since the database size would increase too much for the currently available mass storage systems. Therefore, rather than trying to collect a very large database it is possible to try to select the "right" data. By "right" data, we mean that the units in the database would cover the identified acoustic and phonetic space of the language reasonably well. There are many suggestions for designing the database inventory and utterances to be recorded. For example, one solution is to first model the acoustic space of the speaker and find the units that are acoustically distinct and frequent enough to deserve coverage. In this method one first builds a cluster tree from a general speech database that has a good phonetic coverage. After creating the tree, the number of uses of each cluster is counted using typical utterances for the domain and finally utterances which have the highest score and coverage are greedily selected [7]. This results in a manageable set of utterances and the database provides better synthesis quality (in terms of the database size) than databases that are not well constructed. Most unit selection systems use a fixed unit size, but longer contiguous segments can be selected due to the selection algorithm. Typically, the units are based on either e.g. demisyllables, diphones or triphones but different size units can also be used [8, 11, 43].

The size of the speech database is often also reduced by utilizing various coding methods on stored speech units. Another option is to reduce the number of units stored, and different unit selection methods can be used to find the balance between the database size and the quality of synthesized speech [11, 15, 55].

4.1.2 Diphone Synthesis

Compared to the unit selection synthesis technique, diphone synthesis uses a minimal speech database containing all the diphones occurring in a given language. 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 [69], Pitch Synchronous Overlap Add (PSOLA), see e.g. [14, 81],

or MBROLA, see e.g. [23]. Diphone synthesis usually suffers from the sonic glitches at concatenation points and 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 [41].

4.2 Formant Synthesis

Formant synthesis is based on the source-filter model of speech that is described in [25]. In a formant synthesizer there exist two basic filter structures, namely parallel and cascade, but for a better performance a combination of them is usually applied. In theory, formant synthesis also provides an infinite number of sounds (or sound units), which makes it flexible.

Usually, at least three formants are required to produce intelligible speech. Each formant is modeled with a two-pole resonator which enables both the formant frequency that is the pole-pair frequency and its bandwidth to be specified [20].

Rule-based formant synthesis is based on a set of language/sound unit specific rules which are used to determine all the parameters required to synthesize the desired utterance. Some typical parameters used in the current formant synthesis systems include: fundamental frequency, voiced excitation open quotient, degree of voicing in excitation, formant frequencies and their amplitudes, frequency of an additional low frequency resonator and the intensity of the low and high frequency region [2].

A cascade formant synthesizer is shown in Figure 4.1 and it consists of band-pass resonators that are connected in series. In cascade structure there is only one amplitude control (A), and the relative intensities of the formants are determined by their frequencies (F1, F2, F3) and bandwidths (BW1, BW2, BW3). The output of each formant resonator is then applied as an input of the following resonator. The cascade structure has been found to be better for non-nasal voiced sounds. However, the generation of fricatives and plosive bursts is difficult in a cascade implementation [56].

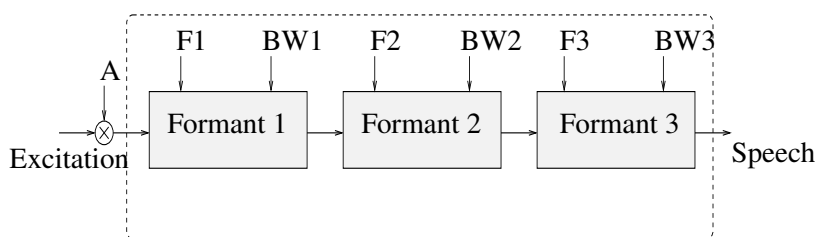


Figure 4.1: Basic structure of cascade formant synthesizer

A parallel formant synthesizer is shown in Figure 4.2 and it consists of res-

onators connected in parallel. The excitation signal is applied to all formants simultaneously and the outputs are summed together. Adjacent outputs of formant resonators have to be summed in opposite phase to avoid unwanted zeros or antiresonance in the frequency response. The parallel structure enables controlling of bandwidths (BW1, BW2, BW3) and gains (A1, A2, A3) for each formant (F1, F2, F3) individually and it has been found to provide better quality for nasals, fricatives and stop consonants [56].

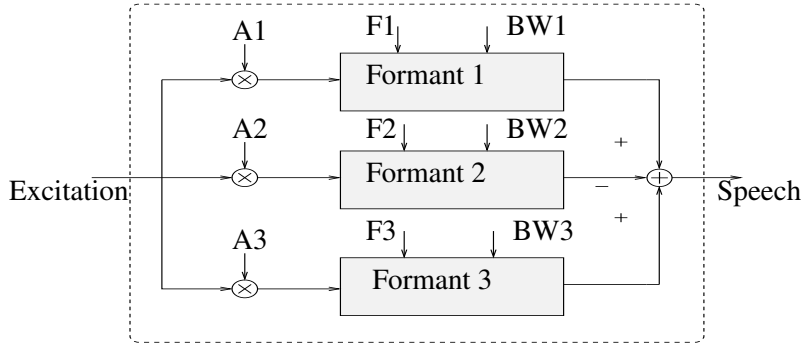


Figure 4.2: Basic structure of parallel formant synthesizer

When formant synthesis is applied in a TTS system, cascade and parallel models are usually combined and tuned to provide better quality. A well known model is the Klatt88 formant synthesizer that has been applied in several TTS systems, such as MITalk, DECTalk and Klattalk [2, 20]. This model includes a more complex formant synthesizer, applying both cascade and parallel models with additional resonances and anti-resonances for nasalized sounds and a sixth formant for high frequency noise. The model also includes a bypass path to give a flat transfer function and radiation characteristics. Klatt88 uses a rather complex excitation model that is controlled using 39 different parameters that are updated every 5 ms. These parameters can be controlled and modified by a set of rules that are applied during synthesis. In Publication 4 [93], a low-footprint TTS system based on formant synthesis is presented. During synthesis the parameters such as formant values and transitions are modified according to the language specific rules. It has been shown earlier that a coarse representation of formant contours for vowels, for example, using 20% and 80% points of the phoneme duration, is adequate for their correct identification and the increase in modeling complexity does not necessarily improve the identification accuracy [40]. The best vowel identification rate is obtained by determining the formant contours based on the onset, target and the offset values of the formants [84]. Based on these results Publication 5 [92] studies the possibility to simplify the rules controlling formant contours by reducing the number of control points that are used to define the formant contours during speech synthesis from four to two control points. In the same publication the per-

ceptual impact of various interpolation techniques between control points, e.g., linear interpolation, cubic spline interpolation and smoothing of the piecewise linear interpolation, was also evaluated. Although formant synthesizers already have a relatively small memory footprint, some other methods for optimizing the footprint even further have also been presented in [80, 103].

A high level diagram of the Klatt88 formant synthesizer is shown in Figure 4.3. In Klatt88 three different voicing source models are available for the

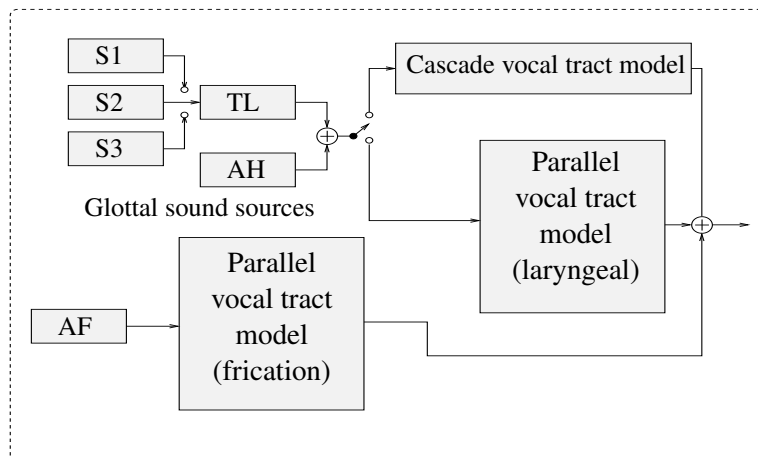


Figure 4.3: Block diagram of Klatt88 formant synthesizer

user: the standard impulse source (S1) described in [56], the new Klatt88 glottal model (S2) [58] and the modified Liljencrants-Fant model (S3) [24, 58]. In Figure 4.3 TL denotes the spectral tilt, AH is the amplitude of aspiration and AF is used to control the amplitude of frication. An example of the the excitation pulse produced by Liljencrants-Fant model is shown in Figure 2.2 in Chapter 2.

4.3 Other Synthesis Techniques

In addition to concatenative synthesis and formant synthesis, other speech synthesis methods have been introduced and some of them are presented in this section.

One alternative synthesis technique is articulatory synthesis, which tries to directly model the human speech production system [57]. Another speech synthesis technique presented is based on linear predictive methods of representing speech [69]. Linear predictive methods were originally designed for speech coding, but they have also been used for speech synthesis [101]. Although many synthesis techniques can produce high quality speech, they are not able to synthesize speech with various voice characteristics such as speaker individualities, speaking styles, emotions etc. To obtain various voice characteristics in speech synthesis based on the selection and concatenation of acoustical units, a large

amount of speech data representing such variations is needed. In order to generate speech synthesis systems able to generate various voice characteristics with relatively small amount of data, Hidden Markov Model (HMM) based synthesis technique has been proposed [131].

4.3.1 Articulatory Synthesis

The basic idea behind articulatory synthesis is to produce synthetic speech by modeling the human articulatory system directly. This means that a mathematical model is defined for every organ of the human articulatory system. Thus, different models exist for example for the lungs, vocal cords, vocal tract, tongue, lips etc. and with the help of these models articulatory synthesis tries to model human speech production as closely as possible. Because of the exact modeling of the human articulatory system, articulatory synthesis would in theory be a good method to produce very natural sounding speech. However, the problem with articulatory synthesis is the complexity of the implementation, for example it is difficult to obtain data for the development of articulatory synthesis, and requirements of computational efficiency. Therefore, because of these requirements articulatory synthesis has not been widely used in real systems to the present day [57].

4.3.2 Linear Predictive Based Methods

Similar to formant synthesis, basic linear predictive coding (LPC) is based on the source-filter model of speech, and the filter coefficients are estimated automatically from a frame of natural speech. The basis of linear prediction is that the current speech sample $y(n)$ can be estimated from a finite number p of previous samples $y(n-1)$ to $y(n-p)$ by a linear combination with a small error $e(n)$. This results in that the speech sample $y(n)$ can be presented as

$$y(n) = \sum_{k=1}^p a(k)y(n-k) + e(n) \quad (4.3)$$

where p is the linear prediction order and $a(k)$ are the linear prediction coefficients that are found by minimizing the sum of squared errors over a frame. Two methods, namely the covariance method and autocorrelation method, are commonly used to calculate these coefficients, but only with the autocorrelation the filter is guaranteed to be stable [60, 125].

In the synthesis phase, the excitation used is approximated by a train of impulses during voiced sounds and by a random noise during unvoiced sounds. The excitation signal is amplified and filtered with a digital filter for which the coefficients are $a(k)$ and they are updated normally every 5-10 ms. The filter order is typically between 10 and 12 at 8 kHz sampling rate, but for higher quality, at 22 kHz sampling rate, the order is typically between 20 and 24 [52, 60].

The main deficiency of the ordinary LP method is that it represents an all-pole model, which means that segments containing antiformants (e.g. nasals and nasalized vowels) are poorly modeled. The quality is also relatively poor for short plosives having a time scale event shorter than the frame size used for analysis. However, modifications and extensions to the basic LP model have been introduced improving the synthesis quality. One example is the Warped Linear Prediction (WLP) model taking advantage of human hearing properties reducing the needed filter order significantly from 20-24 for 22 kHz synthesis to 10-14 [61]. The basic idea is that the unit delays in the filter are replaced by all-pass sections. Depending on the used warping function WLP provides a better frequency resolution at low frequencies and worse at high frequencies, which is however similar to human hearing properties. Several other variations of linear prediction have been developed to increase the quality compared to the basic model [17, 20]. With these methods the excitation signal used is different from the ordinary LP method. Some examples are e.g. Multi-pulse Linear Prediction (MLPC), where the excitation is constructed from several pulses, Residual Excited Linear Prediction (RELP), where the error signal or residual is used as an excitation signal and the speech signal can be reconstructed exactly, and Code Excited Linear Prediction (CELP), in which a finite number of excitations are stored in a finite codebook [12].

4.3.3 Hidden Markov Model Based Synthesis

Although many TTS systems can synthesize speech with acceptable quality, they are not able to synthesize speech with various voice characteristics such as speaker individualities and emotions. To obtain various voice characteristics in TTS systems based on the selection and concatenation of acoustical units, a large amount of speech data is needed. However, it is relatively difficult to collect and segment large amount of speech data for different languages. Moreover, storing big database in devices having only a small amount of memory is not possible. From these points of view, in order to construct a speech synthesis system that can generate various voice characteristics without big speech databases, Hidden Markov Model based speech synthesis has been proposed, see e.g. [119].

In HMM-based speech synthesis system, shown in Figure 4.4, the frequency spectrum (vocal tract), fundamental frequency (vocal source, i.e. excitation), and duration (prosody) of speech are modeled simultaneously by HMMs. During the actual synthesis, speech waveforms are generated from HMMs themselves based on maximum likelihood criteria.

The spectrum part of the HMM output vector is typically based on mel-cepstral coefficients including zeroth coefficients and their first and second order derivatives. Similarly, the temporal structure of speech, in other words HMM state durations, is modeled by using multivariate Gaussian distributions [130]. During speech synthesis the synthesis filter is controlled by the output vector of an HMM

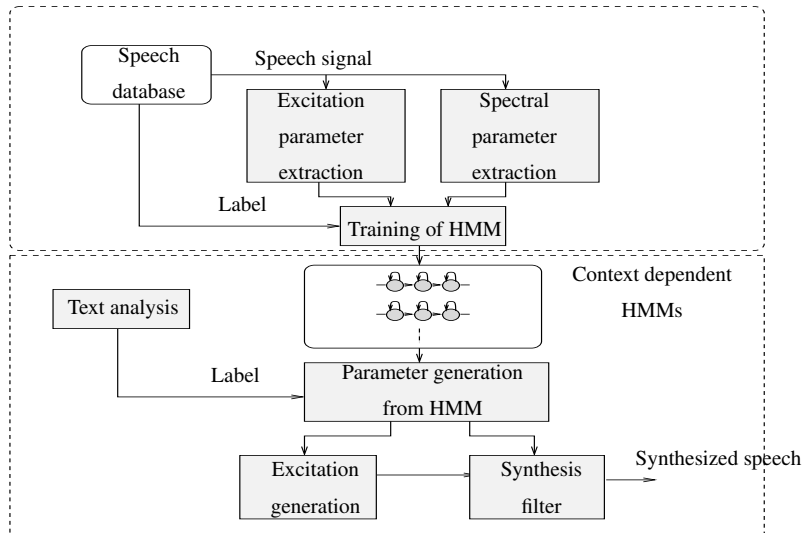


Figure 4.4: HMM-based speech synthesis system

i.e. mel-cepstral coefficients. One solution is to apply the mel-cepstral analysis technique [119], which enables speech to be re-synthesized directly from the mel-cepstral coefficients by using a Mel Log Spectrum Approximation (MLSA) filter [30]. The HMMs are also used to model the fundamental frequency F_0 and the observation sequence for it is composed of one-dimensional continuous values and a discrete symbol which represents whether the phoneme is voiced or unvoiced. Therefore, conventional discrete or continuous HMMs can not be used for F_0 modeling and to model such observation sequences, HMMs based on a multi-space probability distribution (MSD-HMM) have been proposed [118].

Many contextual factors have an effect on the speech spectrum, fundamental frequency pattern and sound duration, and to capture all these effects context dependent HMMs are used [118]. However, as the number of contextual factors increases the number of possible combinations also increases exponentially, and therefore it is not possible to estimate all model parameters accurately with a limited amount of training data. To overcome this problem similar decision tree based context clustering techniques that have been applied in automatic speech recognition have also been applied in HMM-based speech synthesis [87, 105]. Moreover, these techniques were extended for MSD-HMMs [128].

During speech synthesis an HMM corresponding to the input text is constructed by concatenating the context dependent HMMs. The state durations of the constructed HMM are determined by maximizing the output probability of the state durations [130]. Similarly, the sequence of mel-cepstral coefficients and $\log F_0$ values including the discrete voiced/unvoiced parameter is determined by maximizing the speech parameter generation algorithm described in [120]. Fi-

nally, the speech waveform is generated directly from the mel-cepstral coefficients and F_0 values by applying the MLSA filter.

In HMM-based synthesis technique the speech characteristics can be altered by modifying HMM parameters. In fact, it has been shown that voice characteristics of synthesized speech can be changed by applying a speaker adaptation technique [114], a speaker interpolation technique [129], or an eigenvoice technique [104]. Moreover, in HMM synthesis the adaptation techniques can also be used for language adaptation. The HMMs can be trained by applying several monolingual corpora from different languages resulting in a multilingual synthesizer. During synthesis the models can then be adapted to a certain speaker by applying for example MLLR model adaptation [63].

Chapter 5

Multilingual Text-to-Speech: Text Analysis and Prosody Generation

Nowadays, a text-to-speech system is often applied in a mixed language environment in different applications. Therefore, the system should be able to support a number of languages at the same time and also provide a framework for rapid language and voice development. A multilingual TTS system can be defined as a system that takes advantage of common algorithms for multiple languages. Therefore, a collection of language specific synthesizers does not qualify as a Multilingual Text-to-Speech (ML-TTS) system. Ideally, the language specific information and knowledge should be stored as data and all languages should share the common algorithms. In practice, this type of model is difficult to achieve. One reason is that many algorithms and methods applied in TTS systems are often developed for a certain language, usually for Germanic or other Indo-European language, and cannot always be, at least directly, applied to other languages, e.g. Japanese or Chinese.

5.1 Multilingual Text-to-Speech System

A multilingual text-to-speech system can be defined as a TTS system that is able to support multiple languages. The system should also provide a framework for rapid language development. An ML-TTS has to also be able to carry out all the synthesis tasks described in Chapter 3 for all languages the framework supports. To achieve these requirements different approaches can be applied when designing a ML-TTS system. First of all, to make the language development easier and also faster the TTS system should be designed to have a modular structure instead of implementing all sub-modules into a single monolithic system. In a modular TTS

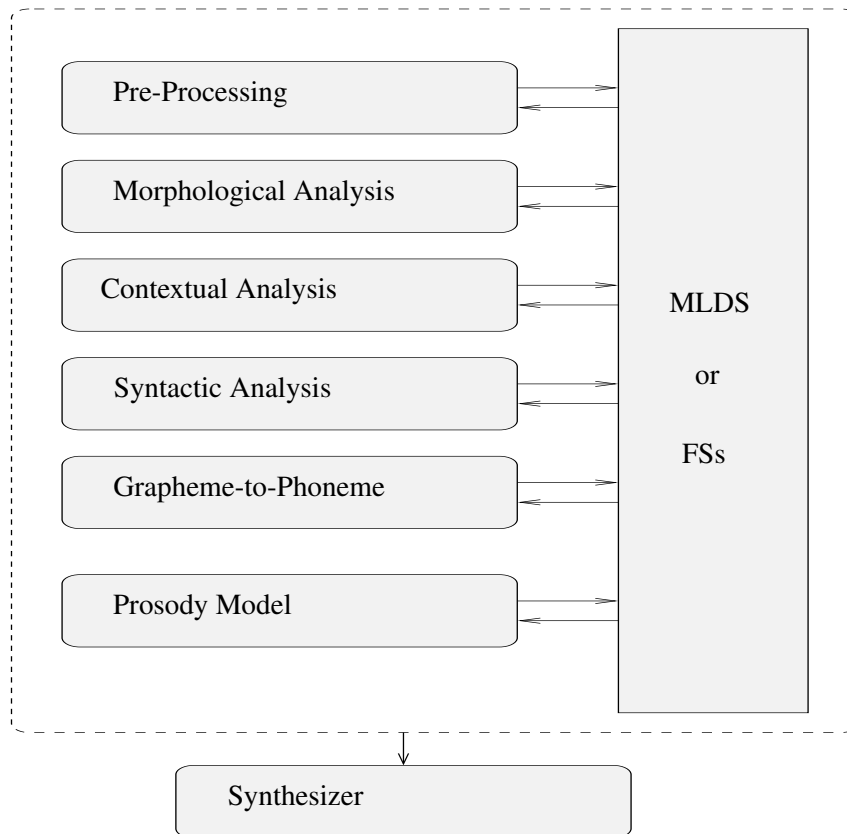


Figure 5.1: Modular text-to-speech system in which MLDS stands for Multi-Level Data Structure and FS stands for Feature Structure.

system, it is also possible to apply language specific processing modules e.g. for text analysis and prosody modeling. However, the drawback of applying language specific techniques in the sub-modules is that if the language specific modules do not share the same computational framework and different languages apply completely different algorithms the system easily becomes complex and difficult to maintain. Also, development of new languages is more difficult if the framework lacks common techniques and basically only defines interfaces between different modules. Therefore, it is desirable to apply methods that can be used for various languages and ideally configured by data. An example of a low memory footprint data configurable text-to-speech system is presented in Publication 4 [93]. The TTS system introduced in Publication 4 consists of a completely language independent engine and language specific data, and the system is currently localized for over 40 languages. There exist also other TTS systems sharing the similar kind of structure e.g. SVOX TTS system [112] and ETI Eloquence [39] and its Delta programming language [38]. A high level block diagram of a modular

text-to-speech system is shown in Figure 5.1.

In the text-preprocessing stage text is usually first divided into sentences and then split into tokens that are separated by white space characters. The text-preprocessing module is also responsible for handling non-standard words, such as abbreviations and numbers. In the morphological analysis stage the input text is analyzed to find the morphemes of each word. Similarly, in the contextual analysis each word is assigned a word class tag and in the syntactic analysis stage the sentence type, e.g. declarative or question, is determined. Finally, in the Grapheme-to-Phoneme (G2P) module the correct pronunciation for the input words is defined and the prosody model is responsible for generating proper prosody for the synthesized speech. Each sub-module takes advantage of a Multi-Level Data Structure (MLDS) or Feature Structure (FS) that are used to store different kinds of dependencies and features describing the textual input. The different stages in Figure 5.1 are described in more detail in the following sections.

5.1.1 Multilingual Text-to-Speech System Framework

Modularity was defined to be one of the design principles of a ML-TTS system. A modular structure can offer many advantages. The first advantage is a standard observation about systems with a modular structure. When the modules of a TTS system share a common interface and the input and output behavior of each module is agreed, it can be easier for different people to work on different modules independently. The second advantage is that pipelining the design makes it possible to stop the processing at any point of the TTS system. For example, when designing a text pre-processing or a G2P module, one might test the sub-module separately. Also, it is of course possible to initiate the processing sequence at any point. For example, one might give a manually tuned phoneme sequence or sound unit durations to the TTS system bypassing the specific modules [108, 110].

Nowadays, many TTS systems parse the text into a multi-level data structure or feature structures as shown in Figure 5.1. The MLDSs and FSs can also be extended easily since it is possible to add extra layers to MLDSs or feature categories in FSs without affecting the previous analysis modules. Furthermore, because the data is made independent of the rule formalism, inter-language portability is better ensured, and MLDS and FS are especially used in TTS systems supporting multiple languages. Another important fact is that if needed it is possible to replace a given implementation of a certain processing block by another implementation having the same functionality but using e.g. a different rule formalism, another programming language or even algorithm [22].

The modular architecture is also an important aid when developing new languages. Some of the modules, for example the actual language independent synthesizer, can be taken and re-used when developing a new language for the TTS system. In the early phase of a development of a TTS system for a new language

one usually lacks much of the detailed information about the language. For example, initially one might only have the G2P conversion module and a speech database or initial set of control parameters for a formant synthesizer available for the new language. Text analysis and prosody modules would be implemented later on. However, one would still be able to get reasonable speech output using this basic or skeletal TTS system. Also the language development of a text-to-speech system usually requires several steps and involvement of experts in various fields. One possible solution to ease the process is an integrated environment for language development e.g. as shown in Publication 3 [77] and in [6, 102] providing the user possibility to tune the different synthesis parameters without a deep knowledge of the underlying software implementation. There are also several other tools for speech processing that can be utilized for different development and analysis tasks, e.g. [62, 107].

5.2 Text Analysis

The text analysis module is the first part of a TTS system. This module can be further divided into submodules, shown in Figure 5.1, handling sentence segmentation, tokenizing and normalization of non-standard words. Also morphological analysis and the assignment of word class features and syntactic analysis are included in the text analysis module. Text analysis is a difficult task and it becomes more challenging if the same framework should be able to support multiple, different languages, such as English, Finnish and Chinese.

5.2.1 Sentence Segmentation and Tokenization

In the sentence segmentation and tokenization stage the input text is first split into sentences and after that the textual input is divided into tokens, usually separated by white space characters. A token can be seen as a categorized block of text, usually consisting of indivisible parts known as lexemes. The main problem in sentence segmentation is the ambiguity of the period that is marking sentence boundaries or abbreviations, sometimes even both at the same time, e.g. "It is 6 a.m." Therefore, the correct function of a punctuation period must be identified. Furthermore, ambiguity regarding capitalized words, proper names vs. sentence initial words must be resolved by the segmentation module. Difficulties also arise from abbreviations that do not differ from normal sentence final words, e.g. "no.", which is also an abbreviation for "number" in English. Several approaches to tackle segmentation problems have been presented. Rule based systems using heuristic period disambiguation operate on local grammars containing abstract contexts for within-sentence periods and sentence boundaries [1, 16, 71]. There exist also automated methods for period disambiguation. Machine learning approaches such as decision tree classifiers use context features such as word

lengths, capitalization and word occurrence probabilities on both sides of the period in question as described in [99].

The simplest approach for tokenization is to split the text at white spaces and punctuation marks, which do not belong to abbreviations identified in the preceding step. However, for some languages, such as Chinese, splitting text into tokens is not as straightforward a task but requires a different algorithm to be used. The difficulty in Chinese is that there is no white space between words. One possible solution is to apply a Finite-State Transducer (FST) to perform the segmentation. FST is a finite state machine with two tapes. (An ordinary finite state automaton has a single tape.) An automaton can be said to recognize a string if we view the content of its tape as input. In other words, the automaton computes a function that maps strings into the set $\{0, 1\}$. Alternatively, one can say that an automaton generates strings, which means viewing its tape as an output tape. On this view, the automaton generates a formal language, which is a set of strings. The two views of automata are equivalent: the function that the automaton computes is precisely the indicator function of the set of strings it recognized. The class of languages generated by finite automata is known as the class of regular languages [50].

It has been shown that FST can also be used to model other text analysis tasks such as handling of non-standard words, morphological analysis, word class assignment and also grapheme-to-phoneme conversion providing a uniform computational treatment of a wide variety of text analysis problems for multiple languages [108].

5.2.2 Handling Non-standard Words

Non-standard words are tokens that need to be expanded into an appropriate orthographic form before the text-to-phoneme module. Normalization of non-standard words includes for example number expansion ("56" \Rightarrow "fifty six"), homograph disambiguation, expansion of abbreviations and symbols ("Grant St" \Rightarrow "Grant Street", "1\$" \Rightarrow "one dollar"), appropriate handling of acronyms (whether they are spelled letter by letter or pronounced as words), e.g. ("BBC" \Rightarrow "b" "b" "c", NATO \Rightarrow "nato"), and also email and URL addresses. A token can be split into several words in this module. In general, normalization is a difficult task. The normalization of non-standard words is not solved by a simple table lookup solely. For example, abbreviations can be context dependent and it is also impossible to know beforehand all possible abbreviations that can be used in the text. Furthermore, the phonetic representation is highly context dependent. While most of the normalization systems try to cope with this problem using heuristic disambiguation and expansion rules, there are also some language modeling and machine learning techniques suitable for text normalization tasks, see for example [109]. Word normalization can be modeled by maximizing the conditional probability of a normalized word sequence given the observed token sequence [109]. In Publi-

ation 6 [78] a multilingual rule-based number expansion is presented. The same framework has been successfully applied for over 40 languages, such as English, Finnish and Chinese. The rule-based system is also able to handle additional text normalization tasks described above.

5.2.3 Morphological Analysis

The task of determining word pronunciation is made easier if the structure, or morphology, of the words is known, and most TTS systems include some morphological analysis. This analysis determines the root form of each word, for example, the root for "gives" is "give". The pronunciation of any derived word can be determined from the pronunciation of the root morphs together with the pronunciation of the affixes, and it is not necessary to include all derived forms in the pronunciation dictionary [54]. Also, even if it is necessary to apply letter-to-sound rules, the rules usually benefit from knowing the morph boundaries and the extracted morphological information can also be utilized in the syntactic analysis stage of a TTS system. In morphological analysis, rules can usually be created to correctly decompose the majority of the words into their constituent morphs. Other possible techniques for identification of morphemes is to apply a statistical approach or e.g. FSTs as mentioned earlier [98, 126].

5.2.4 Word Class Assignment and Prosodic Phrasing

The purpose of the word class assignment module is to assign a word class tag to each token based on its context. This is often referred to as Part-of-Speech (POS) tagging and the input is the tokenized text from the previous module [22]. The tagger has to be able to cope with unknown words that are not found in the dictionary and ambiguous word-tag mappings. For example, in English a word having the exact same way of writing can be either a noun or a verb depending on its role in the sentence [22, 41]. One example is the English word "row" which is pronounced differently depending whether it is a noun or a verb. For example, "I had a row with my friend." and "She told me to row the boat." In the first case the "ow" in "row" is pronounced as in "ouch" and in the second case as in "show".

To cope with the word class assignment problem for different languages, several solutions have been proposed. Rule based approaches operate on dictionaries containing word forms with the associated POS labels and morphological and syntactical features [22]. They also take advantage of context sensitive rules to choose the correct word class labels during the synthesis process. In statistical approaches the most probable POS tag sequence is estimated [9, 49]. In transformation based tagging a hybrid approach is applied and the rules are derived by statistical means [10]. The POS information assigned to each word can then further be used in the TTS system when performing the text-to-phoneme conversion and in the prosody model.

Basic Markov POS Tagger

The principle of the basic classical Markov POS tagger was presented in [49]. In a Markov POS tagger the aim is to estimate the probable tag sequence \hat{T} given the word sequence W :

$$\hat{T} = \arg \max_T [Prob(T|W)]. \quad (5.1)$$

To be able to estimate $Prob(T|W)$ one can apply the Bayes rule to Equation 5.1 and rewrite the equation, given that $Prob(W)$ is constant, as

$$\hat{T} = \arg \max_T [Prob(W|T)Prob(T)]. \quad (5.2)$$

When further assuming that the probability of a word w_i only depends on its tag t_i and that the probability of a tag t_i only depends on a limited tag history the resulting formula for \hat{T} can be given as

$$\hat{T} = \arg \max_{t_1, \dots, t_n} \prod_{i=1}^n Prob(w_i|t_i)Prob(t_i|\tilde{t}_i), \quad (5.3)$$

where \tilde{t}_i is the history of the tag t_i . Finally, \hat{T} given in Equation 5.3 is calculated using the Viterbi algorithm presented in [97, 123].

Prosodic Phrasing

At this stage of the text normalization process, also referred to as syntactic analysis, individual words have been assigned adequate POS categories as the combined result of the morphological analysis and word class assignment algorithm. In order to be able to generate suitable prosody for the sentence, it is necessary to determine the sentence type i.e. declarative, imperative or question and to identify phrases and clauses. Some systems include a full syntactic parsing while others perform a more superficial analysis locating noun and verb phrases and possibly group these phrases into clauses. One possible crude algorithm is called Chinks and Chunks algorithm. This approach divides the words of a given utterance into two as content words and function words, named as chunks and chinks, respectively. The phrases are assumed to begin with a chunk and continue by any number of chinks [67]. For example, in sentence "[She read] [the important pages] [in the park.]" the words "the" and "in" are function words, thus, according to Chinks and Chunks algorithm the phrases are marked between the brackets. Although this simple heuristic works fine on languages such English, it is not suitable for languages having free word order structure, for example Finnish.

There exist also methods to perform statistical prosodic phrasing taking advantage of e.g. neural networks and classification and regression trees (CART). These models try to predict the prosodic phrases based on the POS information, stress, position in the sentence and other relevant features. The general purpose of the module is to be able to produce a reasonable analysis for any text even if the text has syntactic errors.

5.3 Grapheme-to-Phoneme Conversion

The Grapheme-to-Phoneme (G2P) module finds the correct pronunciation for the input words. The simplest approach for G2P conversion is to use a dictionary based approach. In this method a large dictionary contains all the words and their pronunciations. Determining the correct pronunciation of each input word is simple since it only involves looking up each word in the dictionary and replacing it with the pronunciation defined in the dictionary (e.g. "appointment" \Rightarrow "əpɔɪntmənt"). Another approach is a rule-based method where rules for the pronunciation of the words are applied to words to find out their pronunciations based on their spelling. Both of these two approaches have their pros and cons. The dictionary based method is very simple and accurate. However, it fails completely if the word is not found in the lookup dictionary. Also, as the dictionary size grows, the memory requirements of the system also become more demanding. The rule based approach is able to work with any input but the rules easily become very complex. The manual creation of the rules is also a very time consuming and language dependent process [22, 41, 53].

Nowadays, most G2P systems are data-driven and the same methods can be applied for multiple languages [132]. For example, the conversion can be based on statistical decision trees or artificial neural networks using different pools of automatically extracted features. Typical features used are for example: the current grapheme, morphological boundary information, morpheme class the current grapheme belongs to, syllable boundary information, grapheme's position in the syllable and the phoneme history. When training the decision tree the features can be extracted from the training material using for example a centered window of length L for each grapheme at hand [132].

Although a modular multilingual text-to-speech framework can make TTS language development process easier and faster it is still a considerable effort. In addition to the development effort, the support for multiple languages consumes significantly more memory. To speed up the development process and also address the memory requirements in mobile devices it is possible to perform cross-lingual phoneme mappings after G2P conversion. The idea is to model the new language (source language) using an existing synthesis language (target language) on a phonetic level. The cross-lingual phoneme mapping method provides rules for presenting the sounds of the source language with the given set of phonemes of the target language.

To achieve the best quality the target and source languages should be as similar as possible. Because the target language is presented by using the phonemes of the source language by applying a set of mapping rules, the languages should be quite similar both in the prosodic and in the phonetic sense. For example, it is more feasible to map Estonian or Hungarian to Finnish than to American English since Estonian, Hungarian and Finnish all belong to the same language family.

The cross-lingual phoneme mapping method is presented in Publication 2 [76].

5.4 Prosody Modeling

In linguistics, prosody includes the intonation, rhythm and lexical stress in speech. The prosodic features of a unit of speech can be grouped into syllable, word, phrase or clause level features. These features are manifested for example as duration, F0 and intensity. Prosodic units do not need to necessarily correspond to grammatical units. However, phrases and clauses are grammatical concepts but they can also have prosodic equivalents often referred to as prosodic units or intonation units. These units are characterized by several phonetic cues such as certain type of pitch contour and lengthening of vowels within the unit, see for example [4].

The perceived quality of synthetic speech is largely determined by the naturalness of the prosody generated during synthesis. The correct prosody also has an important role in the intelligibility and especially listenability of synthetic speech. Prosody can also convey paralinguistic information to the user, such as joy or anger [4, 22].

In a TTS system, intonation and other prosodic aspects must be generated from the plain textual input. The main challenge is to be able to provide meaningful information for the particular prosody model that is applied in the TTS system [22]. One approach is to parse the text into a tree like structure containing different layers or feature streams. In the lowest stream are the individual phonemes and other phonetic information such as phoneme type, voicing and manner of articulation, for example. The next layer in the tree contains the corresponding graphemes i.e the textual input. Upper layers contain information about syllable and word boundaries. Further on, the words can be grouped into tone groups. Finally, the tree can contain phrase and clause boundary information.

The different level items or nodes in the tree usually contain relevant features that are applied during the construction of the upper layers in the parse tree. Such information includes e.g the lexical stress assigned to a syllable and word class information assigned to word items. The lexical stress for each syllable can be assigned using for example a dictionary or a rule based approach. Another possible solution is to use a statistical method such as classification and regression trees that were trained using the features available in the lower layers of the tree. The advantage of statistical methods is that they can be automatically trained for different languages.

5.4.1 Fundamental Frequency Contour

The fundamental frequency of voiced speech is widely used by all languages to convey information that supplements the sequence of phonemes. In tonal lan-

languages, such as Mandarin and Cantonese, pitch changes are used to distinguish different meanings for syllables being phonetically similar. The four Chinese tones are (1) high level; (2) high rising; (3) low rising; (4) high falling to low. It is not unusual for a syllable to be pronounced in each of the four tones, each yielding a word with a completely different meaning. For example, the word "ma" in tone one means "mother," while "ma2" means "hemp," "ma3" means "horse," and "ma4" means "to curse." In most Western languages pitch does not help directly in identifying words but provides additional information, such as which words in the sentence are most prominent, whether the sentence is a question, statement or command. Pitch also conveys paralinguistic information such as the speaker's mood etc.

Various models have been proposed to generate the fundamental frequency contour. There exist differences between some of the models but the general characteristic of many of the models is that they operate in two stages. The first stage generates an abstract description of an intonation contour and the second stage converts the abstract description into a sequence of F0 values [41].

Superposition models are hierarchically organized and generate F0 contours by overlaying multiple components of different types [41]. One well-known superposition model is the so called Fujisaki model [28] that has been successfully applied to various languages [29, 74]. This model distinguishes between phrase commands and accent commands. The commands are discrete events represented as pulses for the phrase commands and step functions for accent commands. The final F0 contour is obtained by filtering each sequence of commands and combining the output of the two filters with the baseline F0 value [28].

Tone sequence models generate the F0 contour from a sequence of discrete tones that are locally determined and do not interact with each other. One well-known model was developed by Pierrehumbert [90]. In this model tone is defined to be either "high" or "low" and of a different type depending whether it is being associated with a pitch accent, phrase boundary or an intermediate position between a pitch accent and a boundary tone. In synthesis this model is applied by defining a time-varying F0 range and using rules to assign a high or low tone within the range of the defined F0. The final F0 contour is obtained by interpolating between successive targets. Pierrehumbert's model has formed a basis for the intonation transcription system called TOBI (TOnes and Break Indices) [106].

Although the above mentioned intonation models are able to generate natural sounding utterances they require a very detailed specification of the utterance structure. In other words, in superposition models the phrase and accent commands have to be in correct places having a correct amplitude and duration, and in tone sequence models, tone markers have to be defined accordingly. Automatic methods have been proposed to provide information for the various intonation models. For example, CARTs have been used with different intonation models, e.g, the tilt model [21] and PaIntE system [82], and also with Fujisaki

model [73, 83, 100]. In the above mentioned systems, CARTs are used to provide an estimate for features such as accent location and type which can be used in the F0 estimation. It is also possible to apply statistical methods directly estimating segment related pitch values as in the Festival speech synthesis system [8, 116]. An advantage of using statistical methods is that it is possible to automatically extract significant features and dependencies for fundamental frequency contour modeling without writing rules by hand for every new language. See for example [121] that presents different prosody models for Finnish applying artificial neural networks. Similarly, in Publication 1 [75] two different intonation models, namely the Fujisaki model and CART based intonation model, were compared against a natural intonation.

Chapter 6

Text-to-Speech in Voice User Interface

Multimodal systems provide two or more user input modes such as speech, pen, touch, manual gestures, gaze and body and head movements in a coordinated manner with multimedia system output. The growing interest in multimodal interface design is inspired by the goal of supporting more transparent, flexible and efficient expressive means of human-computer interaction.

Multimodal systems can have numerous advantages compared to traditional interfaces. Multimodal interfaces permit flexible use of input and also output modes. This includes the choice of which modality to use for conveying different types of information, to use combined input and output modes or to alternate between modes at any time. Since individual input and output modalities are well suited in some situations and less or even inappropriate in others, modality choice is an important design principle in a multimodal system. For example, as systems become more complex, a single modality simply does not permit all users to interact effectively across all tasks and environments.

Because there are large individual variations in the ability and preference to use different methods of communication, a multimodal interface permits diverse user groups to exercise selection and control of how they interact with the device. In this respect multimodal interfaces have the potential to accommodate a broader range of users including users of different ages, skill levels, cognitive styles, sensory impairments and other temporary illness or permanent handicaps. For example, a visually impaired user may prefer speech input and text-to-speech output. In contrast, a user with a hearing impairment or accented speech may prefer touch, gesture or pen input. Also, the natural alternation between different modes that is permitted by the multimodal interface can also be an effective way of preventing overuse and physical damage to any single modality, especially during extended periods of using the device.

Multimodal interfaces also provide the adaptability that is needed to accom-

modate the changing conditions of mobile use. In particular, systems involving speech, pen or touch input are suitable for mobile use and when conditions change users can switch between these modalities [68, 88].

This chapter focuses on using speech and especially text-to-speech output as one mode of the user interface. In this chapter these types of interfaces are referred to as voice user interfaces (VUIs). Voice user interfaces use speech technology to provide users with access to information, allow them to perform operations and also to support communications.

6.1 Voice User Interface

By definition a voice user interface is what the user interacts with when communicating with a given device through speech. The elements of VUI include prompts, grammars and dialog logic. The prompts, or system messages, are all the recordings or synthesized speech played to the user during the dialog. Grammars define for example the possible words and sentences users can say in response to each prompt. The dialog logic on the other hand defines the actions taken by the system, for example responding to what the user has said by performing a certain action or reading out information or text retrieved from a database or some other information source.

VUI design is a multidisciplinary field and therefore designing a voice user interface requires knowledge about various fields of science, for example speech technology, user interface design, linguistics, cognitive psychology and software development and design. All these fields have contributed to the current understanding of voice user interface design. Usability testing is also an important part in any application design, and applications taking advantage of speech technologies are no exception. Currently many users are not very familiar with voice user interfaces, and designing a working solution is not an easy task. Therefore, it is important to include the target users in the design process of the application taking advantage of speech input and output. In next section some basic usability testing methods are presented and they can be applied when building a VUI [18, 51, 70]. Publication 7 [94] presents a text message reader application taking advantage of a TTS system. Also the results of the immediate usability test of the user interface of the application are discussed in the same publication.

6.1.1 Usability Testing

Usability testing is a means for measuring how well people can use some human-made object (such as a web page, a computer interface, a document, or a device) for its intended purpose, i.e., usability testing measures the usability of the object. Many different methods to evaluate usability have been proposed. In this section

some high level basics of usability testing are discussed in brief and some well known methods are introduced.

Basically usability testing of any application should begin early in the design process. One common approach enabling usability testing is referred to as a Wizard of Oz (WOZ) test and it is described in more details in [27]. The key idea in the test is to simulate the behavior of the system by having a human acting as the system performing virtual speech recognition and speech synthesis. The WOZ test has several advantages such as early testing, testing is not subject to any software and integration bugs as would be the case using a prototype system, and changes and modifications are quick and easy to implement. Another approach is to run the usability tests using a working prototype. Quite often this is done later in the design cycle than WOZ testing. Naturally, the main advantage of using a prototype is realism. The behavior of the system is likely to more accurately resemble the performance of the final system.

One well known method for quick and easy evaluation of the user interface design is the heuristic evaluation proposed by Jacob Nielsen [85]. Heuristic evaluation is performed as a systematic inspection of a user interface design for usability. The goal of heuristic evaluation is to find the usability problems in the design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators, typically five to ten, to examine the interface and judge its compliance with recognized usability principles (the "heuristics"). There exist also many other methods to evaluate the usability of an application, such as formal usability evaluation, standards inspection, consistency inspection and cognitive walkthrough. These other methods all have their benefits and are preferred under certain circumstances, but heuristic evaluation is the most general of the usability inspection methods and relatively easy to apply [85].

6.2 TTS in Voice User Interface

A TTS system is able to offer a number of conveniences when designing voice user interfaces. Because there is no need for prerecorded prompts, implementation can be relatively easy and straightforward. Prompts can be produced quickly and edited easily. More importantly, the content of TTS messages can be spontaneous and can also be changed dynamically during run time. In contrast, applications relying on prerecorded audio files alone are best suited for conveying information that is static. When the information changes often or needs editing, the speaker has to be asked to record the whole utterance or concatenation units that will in turn form whole utterances. But when the textual content is unconstrained as in the case of e-mail or text message reading it is impossible to use prerecorded audio files.

However, despite the fact that the voice quality of the TTS systems has improved during the last years and TTS as technology offers many conveniences

there are also some drawbacks compared to recorded audio prompts. Users are still aware that TTS is not real human speech. One area of dissatisfaction is that TTS output is usually more difficult to understand compared to human speech. One of the main problems is the quality of prosody in synthetic speech. The prosody of messages is highly dependent on different contextual cues. When a native speaker reads aloud certain text one gathers contextual information over the course of the text, and in real life, the prosody of a particular sentence is often determined by information that was presented several sentences earlier. However, current TTS systems are still not able to produce rhythm that humans can adopt naturally in sentences containing rhyming clauses, or to generate other systematic variations related to meaning. Synthesized speech also usually lacks the pattern of pauses that are found in speech produced by a human reader. These shortcomings in the prosody of synthetic speech increase the cognitive burden required for comprehending TTS. Some of these issues are mitigated to the extent that listeners seem to be able to adopt to the sound of the TTS and there is evidence that repeated exposure to TTS improves comprehension. It is likely that the users learn to lower their expectations towards phonetic and prosodic naturalness [18, 70].

Because human speech is generally preferred over TTS, audio recordings should be used whenever possible. In some cases a single sentence may have sections that are dynamic whereas other parts are static. For example, if the system has to inform that the user has received a text message from a certain person the name of the person is the dynamic part of the sentence. The question in these cases is if the consistency of a TTS system is more important than combining TTS with real recorded speech. Studies have shown that at least in some cases the users prefer prompts that use both TTS and recorded speech rather than TTS for the entire prompt. However, this result is likely to depend on the quality of the TTS system used and should be evaluated case by case [18].

In some cases it is also possible to mark up the text to be given to TTS in order to get as natural sounding a result as possible. For example, inserting short pauses between sentences and major phrases can improve the naturalness and intelligibility. In addition to the use of pauses, there exist prosodic mark-up strategies aiming to make a TTS system simulate natural, human like intonation patterns. It is also possible to add specific phonetic spellings for certain words that are consistently pronounced incorrectly by the TTS system. There exist also a standard markup language to improve the quality of the synthesized speech [117]. The aim is to be able to improve the quality of the synthesized content by giving additional information to the speech synthesis system. Speech Synthesis Markup Language (SSML) [117] was designed to provide a rich, XML-based markup language to give authors of synthesizable content a standard way to control aspects of speech output such as pronunciation, volume, pitch, etc. across different synthesis capable platforms. Nowadays, there is also an ongoing action to extend the language support of the SSML. In Publication A [95], the development chal-

Challenges of a multilingual text-to-speech system are discussed and some ideas how an SSML based markup language could be used and extended are presented. An overview of methods for improving the quality of TTS for a given application is presented in [45]. Another standard markup language for specifying interactive voice dialogues between human and a computer is called VoiceXML. VoiceXML has tags that instruct the voice browser to provide speech synthesis, automatic speech recognition, dialog management, and sound file playback [124].

6.2.1 Applications of Text-to-Speech and Automatic Speech Recognition

Text-to-speech and automatic speech recognition techniques can be applied in several applications, some of which are presented next.

Telecommunications services nowadays take advantage of TTS systems. TTS can be used to convey information about e.g. timetables and different events and to give access to huge databases that can hardly be read and stored as digitized speech. Queries to such information retrieval systems can be made through the user's voice with the help of automatic speech recognition system or through the telephone keyboard.

TTS can also be used to convey information when oral information is more efficient than written messages shown e.g. on a computer display. The appeal is stronger when the user has to be able to focus on other visual sources of information. TTS systems have been incorporated in measurements and control systems where the user is easily overwhelmed by visual information. One example is the speaker independent name dialing system found in many cell phones, see e.g. [46]. The user is able to make a phone call by saying the person's name which is recognized by an automatic speech recognition system and hears the recognition verification using a TTS system. This is beneficial since voice dialing is likely to be used in situations where the user is not able to pay attention to the mobile phone display, for example when one is driving a car [68, 88]. In such situations, text-to-speech systems can also be used to provide eyes free access to e.g. e-mails and text messages. Publication 7 [94] presents an application using a unit selection TTS system for reading text messages. Usually, when automatic speech recognition is used in applications the ASR system takes advantage of some adaptation techniques improving noise robustness and also reducing the speaker variability. Some of these methods were presented in Section 3.2 on page 23.

Text-to-speech and other voice techniques can also be used to aid people with disabilities. For example, people having speech impairments can benefit from TTS technology. Also visually impaired users can take advantage of a TTS system combined with the optical character recognition technology. There also exist many other cognitive, perceptual and physical impairments (PI) that can hinder the traditional use of computing technologies. In situations where the PI is affect-

ing the upper body, preventing the physical interaction with the application TTS especially combined with the ASR system can greatly improve the usability of the application. The solutions designed to improve the usability of a computing system for people having impairments can also provide useful information when designing applications for children and elderly people, two groups whose cognitive, perceptual and physical capabilities may require special attention [47, 89].

It is also possible to apply TTS and ASR in many other applications such as in talking books and toys, different multimedia applications and computer systems. Some applications of TTS and also ASR in mobile devices are presented in Publication B [79].

Chapter 7

Conclusions

Currently, most interactions between a user and different devices still rely on traditional technologies such as keyboards and displays. In many cases these well known input and output devices are well suitable for transferring information and are also relatively hard to replace effectively. However, as devices become more complex, interaction between humans and computers also becomes more demanding and sets new requirements for the user interface. The development in speech synthesis and automatic speech recognition has made it possible to consider using speech in human computer interaction.

Mobile devices usually have a rather small display and also the physical size of the keyboard or keypad can make the use of the device more difficult and demanding. Therefore, such devices can benefit from using speech as part of the user interface. Today, automatic speech recognition and speech synthesis are already applied in different applications in mobile devices, such as name dialing using ASR and screen readers based on TTS, and they can be beneficial especially when the user is not able to pay attention to the device, e.g. when driving a car.

However, applying speech technologies, such as TTS, in mass produced mobile devices introduces some limitations and restrictions on the technology and system. For example, memory consumption of several megabytes is usually not acceptable and therefore supporting multiple languages on a single device with small memory resources becomes a major challenge. At the same time, wide language support is considered to be important and a text-to-speech system should be able to support all the user interface languages the operating system supports. Because of these requirements the TTS system framework should be easily configurable for different hardware resources and also provide a relatively easy and rapid language development framework. This chapter summarizes the methods presented in this thesis for improving the language development process and reducing the memory footprint of a multilingual TTS system. Furthermore, the key issues of applying a TTS system as part of a voice user interface are discussed and

some possible extensions and ideas for future work are pointed out.

7.1 Some Methods for Multilingual Text-to-Speech in Mobile Devices

The development of a TTS system is an interdisciplinary effort and it requires knowledge about human speech production and about the languages being developed. The actual implementation work on the other hand requires software skills and therefore it is often difficult to find a single person to master all the areas of TTS development. Especially, the development of multiple languages usually requires linguistic knowledge that can only be acquired by consulting the experts familiar with the given language. Therefore, the separation of the actual language creation process from the actual TTS engine development is beneficial when developing a multilingual TTS system. Multilingual TTS systems have been usually designed so that the actual TTS engine and language specific modules and data are completely separate and the addition of a new language is usually relatively straightforward, especially, if the framework is designed so that the technologies applied e.g. for text normalization and intonation modeling can be applied for a wide range of languages. A novel multilingual number expansion framework is presented and it has been successfully applied for over 40 languages. The same framework is also able to support other text-normalization tasks such as processing of context dependent abbreviations and interpretation of formatted text e.g. date and time expressions. The further development of the multilingual data configurable text processing framework would deserve more research efforts. It is also possible to introduce specific development environments and tools to ease the language creation process and one example of an integrated development environment is presented in this thesis.

However, the size of the system increases every time a new language is added. The most memory intensive parts of the whole TTS system are just the ones containing language specific information e.g. lexicons and speech databases. The footprint of the system can be reduced by applying different coding methods on stored speech units and other language specific data. Another alternative presented in the thesis is based on cross lingual phoneme mapping. In this method the phonetic transcription of a new language is presented by using the phoneme set supported by the existing TTS system. The synthesis output of the mapped language is phonetically fairly accurate but the intonation is based on the existing language and the method is best applied for the synthesis of short utterances and isolated words when the language portfolio has to be rapidly expanded. In multilingual TTS the attraction of HMM-based synthesis is that speech characteristics can be altered by applying speaker adaptation techniques. The HMMs trained with several monolingual corpora can be adapted to a certain speaker during the

synthesis phase and the development of a truly multilingual text-to-speech system that can be adapted to different languages deserves more research efforts.

7.2 Text-to-Speech in Voice User Interface

In principle, a TTS system is able to offer a number of advantages when designing voice user interfaces. Because there is no need for prerecorded prompts, the implementation can be relatively easy and straightforward. More importantly, the content of TTS messages can be spontaneous and also changed dynamically during run time. Therefore, TTS systems are able to enable applications such as e-mail, news and screen reader applications and can greatly benefit users with certain disabilities. In contrast, applications relying solely on prerecorded audio prompts are best suited for conveying information that is static. Some applications for mobile devices taking advantage of speech technologies are presented in Publication B [79]. Despite the fact that the voice quality of the TTS systems has improved during the last years and that TTS as technology is able to offer many conveniences there are also some drawbacks compared to recorded prompts. Users are still aware that TTS is not real human speech and TTS output is usually more difficult to understand than real human speech. One of the main problems is the quality of the prosody in synthetic speech. When a native speaker reads aloud certain text one gathers contextual information over the course of the text and in real life prosody of a sentence is often determined by information that was available several sentences ago. Current TTS systems are not able to gather such information and produce natural sounding prosody which increases the cognitive load of the user. Another problem when applying a text-to-speech system for reading unconstrained textual input is that if the text violates the official grammar and for example makes extensive use of abbreviations omitting punctuation, the text processing module of a TTS system is usually not able to process the input correctly. This can result in for example wrong expansions of certain abbreviations and totally incorrect intonation. To overcome these problems, more research work in natural language processing and prosody modeling techniques in speech synthesis is required.

Bibliography

- [1] J. Aberdeen, D. Burger, L. Hirschman, P. Robinson, and M. Vilain, “MITRE: Description of the alembic system used for MUC-6,” in *Proceedings of the Sixth Message Understanding Conference*, 1995, pp. 141–155.
- [2] J. Allen, M. S. Hunnicutt, and D. Klatt, “From text to speech, the MITalk system,” Cambridge University Press, Cambridge, 1987.
- [3] B. S. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” *Journal of the Acoustical Society of America, JASA*, vol. 55, pp. 1304–1312, 1974.
- [4] M. J. Ball and J. Rahilly, *Phonetics, the science of speech*. New York, USA: Oxford University Press Inc, 1999.
- [5] E. Baum, T. Petrie, G. G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.
- [6] J. Beskow, “A tool for teaching and development of parametric speech synthesis,” in *Proceedings of Fonetik’98, Swedish Phonetics Conference*, Stockholm, Sweden, 1998.
- [7] A. Black and K. Lenzo, “Optimal data selection for unit selection synthesis,” Scotland, 2001, in 4rd ESCA Workshop on Speech Synthesis.
- [8] A. Black, P. Taylor, and R. Caley, *The Festival Speech Synthesis System, System Documentation*, Centre for Speech Technology Research, University of Edinburgh, 1999.
- [9] E. Brill, “A simple rule-based part-of-speech tagger,” in *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, Trento, IT, 1992, pp. 152–155.
- [10] ———, “Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging,” *Computational Linguistics*, vol. 21, no. 4, pp. 543–565, 1995.

- [11] N. Campbell and A. Black, "Prosody and the selection of source units for concatenative synthesis," in *Progress in Speech Synthesis*. New York, USA: Springer-Verlag, 1997.
- [12] G. Campos and E. Gouvea, "Speech synthesis using the CELP algorithm," in *Proceedings of ICSLP*, vol. 3, 1996.
- [13] I. Catford, *Fundamental Problems in Phonetics*. Edinburgh University Press, 1977.
- [14] F. Charpentier and M. G. Stella, "Diphone synthesis using an overlap-add technique for speech waveform generation," in *Proceedings of ICASSP*, 1986, pp. 2015–2018.
- [15] D. Chazan, R. Hoory, Z. Kons, D. Silberstein, and A. Sorin, "Reducing the footprint of the IBM trainable speech synthesis system," in *Proceedings of ICSLP*, Denver, USA, 2002.
- [16] L. Cherry and W. Vesterman, "Writing tools- the STYLE and DICTION programs," University of California, Berkeley, Tech. Rep., 1991.
- [17] D. Childers and H. Hu, "Speech synthesis by glottal excited linear prediction," *Journal of the Acoustical Society of America, JASA*, vol. 96(4), pp. 2026–2036, 1994.
- [18] M. Cohen, J. Giangola, and J. Balogh, *Voice User Interface Design*. Addison Wesley, 2004.
- [19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. B 39, pp. 1–38, 1977. [Online]. Available: <http://www.stat.washington.edu/hoff/Preprints/jcgs.12-00.pdf>
- [20] R. Donovan, "Trainable speech synthesis," Ph.D. dissertation, Cambridge University Engineering Department, 1996.
- [21] K. Dusterhoff, A. Black, and P. Taylor, "Using decision trees within the tilt intonation model to predict F0 contours," in *Proceedings of Eurospeech*, Budapest, Hungary, 1999.
- [22] T. Dutoit, *An Introduction to Text-to-Speech Synthesis*. Dordrecht, the Netherlands: Kluwer Academic Publisher, 1997.
- [23] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken, "The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes," in *Proceedings of ICSLP*, Philadelphia, USA, 1996.

- [24] G. Fant, J. Liljencrants, and Q. G. Lin, "A four-parameter model of glottal flow," Royal Institute of Technology, Stockholm, Sweden, Tech. Rep., 1985.
- [25] J. Flanagan, *Speech Analysis, Synthesis and Perception*. Springer-Verlag, 1972.
- [26] J. Flanagan and L. Rabiner, *Speech Synthesis*. Dowden, Hutchinson & Ross, 1973.
- [27] J. Fraser and G. Gilbert, "Simulating speech systems," *Computer Speech and Language*, pp. 81–99, 1991.
- [28] H. Fujisaki, "From information to intonation," in *Proceedings of the International Symposium on Spoken Dialogue*, 1993, pp. 7–18.
- [29] H. Fujisaki and S. Ohno, "Analysis and modeling of fundamental frequency contours of English utterances," in *Proceedings of Eurospeech*, vol. 2, Madrid, Spain, 1995, pp. 985–988.
- [30] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proceedings of ICASSP*, vol. 1, 1992, pp. 137–140.
- [31] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, pp. 254–272, 1981.
- [32] ———, "Flexible speech recognition," in *Proceedings of Eurospeech*, 1995, pp. 1595–1603.
- [33] M. Gales, "Model-based techniques for noise robust speech recognition," Ph.D. dissertation, Cambridge University, 1996.
- [34] M. Gales and S. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Transactions on Speech and Audio Processing*, 4(5):352–359, September 1996.
- [35] J. Gauvain and C. Lee, "Maximum a-posteriori estimation for multivariate Gaussian mixture observations of Markov chains," 1994, *IEEE Transactions on Speech and Audio Processing*, 2:291–298.
- [36] M. Goldstein, "Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener," *Speech Communication*, vol. 16, pp. 225–244, 1995.

- [37] W. I. Hallahan, "DECtalk software: Text-to-speech technology and implementation," *Digital Technical Journal of Digital Equipment Corporation*, vol. 7, no. 4, pp. 5–19, Mar. 1996.
- [38] S. R. Hertz, "The Delta programming language: An integrated approach to non-linear phonology, phonetics and speech synthesis," in *Papers in Laboratory Phonology I*, 1990.
- [39] S. R. Hertz, R. J. Younes, and N. Zinovieva, "Language-universal and language-specific components in the multi-language ETI-eloquence text-to-speech system," in *Proceedings of XIV International Congress of Phonetic Sciences*, vol. 3, San Francisco, CA, 1999, pp. 2283–2286.
- [40] J. M. Hillenbrand and T. M. Nearey, "Identification of resynthesized /hVd/ utterances: Effects of formant contour," *J. Acoust. Soc. Amer.*, vol. 105(6), pp. 3509–3523, 1999.
- [41] J. Holmes and W. Holmes, *Speech Synthesis and Recognition*. New York, USA: Taylor & Francis, 2001.
- [42] A. S. House, C. E. Williams, M. H. L. Hecker, and K. D. Kryter, "Articulation testing methods: Consonantal differentiation with a closed-response set," *Journal of the Acoustical Society of America, JASA*, vol. 37, pp. 158–166, 1965.
- [43] A. Hunt and A. Black, "Unit selection in concatenative speech synthesis system using a large speech database," in *Proceedings of ICASSP*, 1996, pp. 373–376.
- [44] "IPA (International Phonetic Association) homepage," <http://www.arts.gla.ac.uk/IPA/ipa.html>.
- [45] R. Ishihara, "Enhancing TTS performance," in *Proceedings of Telephony Voice User Interface Conference*, San Diego, 2003.
- [46] J. Iso-Sipilä, O. Viikki, and M. Moberg, "Multi-lingual speaker-independent voice user interface for mobile devices," in *Proceedings of ICASSP*, Toulouse, France, 2006.
- [47] J. A. Jacko, H. S. Vitense, and I. U. Scott, "Perceptual impairments and computing technologies," in *The Human Computer Interaction Handbook*. New Jersey, USA: Lawrence Erlbaum Associates, Inc., 2003.
- [48] U. Jekosch, "Speech quality assessment and evaluation," in *Proceedings of Eurospeech*, vol. 2, 1974, pp. 1387–1394.

- [49] F. Jelinek, "Markov source modeling of text generation," in *The Impact of Processing Techniques on Communications, volume E91 of NATO ASI Series*, 1985, pp. 569–598.
- [50] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 2000.
- [51] C.-M. Karat, J. Vergo, and D. Nahamo, "Conversational interface technologies," in *The Human Computer Interaction Handbook*. New Jersey, USA: Lawrence Erlbaum Associates, Inc., 2003.
- [52] M. Karjalainen, T. Altosaar, and M. Vainio, "Speech synthesis using warped linear prediction and neural networks," in *Proceedings of ICASSP*, 1998.
- [53] E. Keller, *Fundamentals of Speech Synthesis and Speech Recognition*. West Sussex, England: John Wiley & Sons Ltd, 1994.
- [54] K. Kirchhoff, "Language characteristics," in *Multilingual Speech Processing*. Elsevier, 2006.
- [55] E. Klabbers and K. Stöber, "Creation of speech corpora for the multilingual Bonn open synthesis system," in *Proceedings of 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Pitlochery, Scotland, 2001, pp. 23–27.
- [56] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America, JASA*, vol. 67(3), p. 971, 1980.
- [57] —, "Review of text-to-speech conversion for English," *Journal of the Acoustical Society of America, JASA*, vol. 82(3), pp. 737–793, 1987.
- [58] D. H. Klatt and L. C. Klatt, "Analysis, synthesis and perception of voice quality variations among female and male talkers," *Journal of the Acoustical Society of America, JASA*, vol. 87(2), p. 820, 1990.
- [59] H. Klaus, H. Klix, J. Sotscheck, and K. Fellbaum, "An evaluation system for ascertaining the quality of synthetic speech -based on subjective category rating tests," in *Proceedings of Eurospeech*, vol. 3, 1993, pp. 1679–1682.
- [60] W. Kleijn and K. Paliwall, *Speech Coding and Synthesis*. Lausanne: Elsevier, 1995.
- [61] U. Laine, M. Karjalainen, and T. Altosaar, "Warped linear prediction (WLP) in speech synthesis and audio processing," in *Proceedings of ICASSP*, vol. 3, 1994, pp. 349–352.

- [62] Y. Laprie, “Snorri, a software for speech sciences,” *MATISSE (Method and Tool Innovations for Speech Science Education)*, pp. 89–92, London, 1999.
- [63] J. Latorre, K. Iwano, and S. Furui, “Speaker adaptable multilingual synthesis,” in *Proceedings of Symposium on Large Scale Knowledge Resources (LKR 2005)*, 2005, pp. 235–238.
- [64] K. Laurila, “Robust speech recognition for voice dialing,” Ph.D. dissertation, Tampere University of Technology, Department of Information Technology, 2000.
- [65] C. Leggetter and P. Woodland, “Flexible speaker adaptation using maximum likelihood linear regression,” in *Proceedings of Eurospeech*, Madrid, Spain, 1995, pp. 1155–1158.
- [66] W. Li, “Random texts exhibit Zipf’s-law-like word frequency distribution,” vol. 38(6), pp. 1842–1845, 1998.
- [67] M. Y. Liberman and K. W. Church, “Text analysis and word pronunciation in text-to-speech synthesis,” in *Advances in Speech Signal Processing*, Dekker, 1992, pp. 791–831.
- [68] S. Love, *Understanding Mobile Human-Computer Interaction*. Great Britain: Elsevier Ltd., 2005.
- [69] J. Markel and A. G. Jr., *Linear Prediction of Speech*. Springer-Verlag, 1976.
- [70] M. McTear, *Spoken Dialogue Technology: Toward the Conversational User Interface*. Springer Verlag, 2004.
- [71] A. Mikheev, “Periods, capitalized words, etc.” *Computational Linguistics*, vol. 28(3), pp. 289–318, 2002.
- [72] Y. Minami and S. Furui, “An HMM adaptation method for noise and distortion by maximizing likelihood,” pp. 1–9, 1998.
- [73] H. Mixdorff, “A novel approach to the fully automatic extraction of Fujisaki model parameters,” in *Proceedings of ICASSP*, vol. 3, Istanbul, Turkey, 2000, pp. 1281–1284.
- [74] H. Mixdorff and H. Fujisaki, “A scheme for a model-based synthesis by rule of F0 contours of German utterances,” in *Proceedings of Eurospeech*, vol. 3, Madrid, Spain, 1995, pp. 1823–1826.
- [75] M. Moberg and K. Pärssinen, “Comparing CART and Fujisaki intonation models for synthesis of US-English names,” in *Proceedings of Speech Prosody*, Nara, Japan, 2004, pp. 439–442.

- [76] —, “Cross-lingual phoneme mapping for multilingual synthesis systems,” in *Proceedings of ICSLP*, Jeju Island, Korea, 2004, pp. 1029–1032.
- [77] —, “Integrated development environment for a multilingual data configurable synthesis system,” in *Proceedings of International Conference of Speech and Computer*, Greece, 2005, pp. 155–158.
- [78] —, “Multilingual rule-based approach to number expansion: Framework, extensions and application,” *International Journal of Speech Technology*, 2006.
- [79] —, “Using text-to-speech in mobile phones,” in *Proceedings of Phonetics Symposium*, Helsinki, Finland, 2006, pp. 125–133.
- [80] M. Moberg and O. Viikki, “Optimizing speech synthesizer memory footprint through phoneme set reduction,” in *Proceedings of IEEE Workshop on Speech Synthesis*, Santa Monica, USA, 2002.
- [81] E. Moulines and F. Charpentier, “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones,” *Speech Communication*, vol. 9, no. 5–6, 1990.
- [82] G. Möhler and A. Conkie, “Parametric modeling of intonation using vector quantization,” in *Proceedings of 3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 311–314.
- [83] E. Navas, I. Hernandez, A. Armenta, B. Etxebarria, and J. Salabarría, “Modeling Basque intonation using Fujisaki’s model and CARTs,” in *State of the Art in Speech Synthesis*, London, UK, 2000.
- [84] A. T. Neel, “Formant detail needed for vowel identification,” *Acoustic Research Letters Online (ARLO)*, vol. 5(4), pp. 125–131, 2004.
- [85] J. Nielsen, *Usability Engineering*. Academic Press, 1998.
- [86] P. W. Nye and J. H. Gaitenby, “The intelligibility of synthetic monosyllabic words in short, syntactically normal sentences,” 1974, Haskins Laboratories Status Report on Speech Research, 37/38:169–190.
- [87] J. J. Odell, “The use of context in large vocabulary speech recognition,” Ph.D. dissertation, Cambridge University, 1995.
- [88] S. Oviatt, “Multimodal interfaces,” in *The Human Computer Interaction Handbook*. New Jersey, USA: Lawrence Erlbaum Associates, Inc., 2003.
- [89] —, “Physical disabilities and computing technologies: An analysis of impairments,” in *The Human Computer Interaction Handbook*. New Jersey, USA: Lawrence Erlbaum Associates, Inc., 2003.

- [90] J. Pierrehumbert, "The phonology and phonetics of English intonation," Ph.D. dissertation, MIT, 1980.
- [91] D. B. Pisoni, "Perception of synthetic speech," in *Progress in Speech Synthesis*. New York, USA: Springer-Verlag, 1997.
- [92] K. Pärssinen and M. Moberg, "Evaluation of perceptual quality of control point reduction in rule-based synthesis," in *Proceedings of ICSLP*, Pittsburgh, Pennsylvania, 2006, pp. 2070–2073.
- [93] ———, "Multilingual data configurable text-to-speech system for embedded devices," in *Proceedings of Multiling*, Stellenbosch, South Africa, 2006.
- [94] K. Pärssinen, M. Moberg, and M. Gabbouj, "Reading text messages using a text-to-speech system in Nokia Series 60 mobile phones: Usability study and application," Tampere University of Technology, Tech. Rep., 2006.
- [95] K. Pärssinen, M. Moberg, M. Harju, and O. Viikki, "Development challenges of multilingual text-to-speech systems," Heraklion, Greece, 2006, internationalizing W3C's Speech Synthesis Markup Language, Workshop II.
- [96] K. Pärssinen, P. Salmela, M. Harju, and I. Kiss, "Comparing Jacobian adaptation with cepstral mean normalization and parallel model combination," in *Proceedings of ICASSP*, Orlando, Florida, 2002, pp. 193–196.
- [97] L. Rabiner, *Fundamentals of Speech Recognition*. Prentice-Hall Inc., 1993.
- [98] U. Reichel and F. Schiel, "Using morphology and phoneme history to improve grapheme-to-phoneme conversion," in *Proceedings of Eurospeech*, 2005, pp. 1937–1940.
- [99] M. Riley, "Some applications of tree-based modeling to speech and language indexing," in *Proceeding of the DARPA Speech and Natural Language Workshop*, 1989, pp. 339–352.
- [100] P. Rossi, F. Palmieri, and F. Cutugno, "A method for automatic extraction of Fujisaki-model parameters," in *Proceedings of Speech Prosody*, Aix-en-Provence, April 2002.
- [101] M. Schroeder, "A brief history of synthetic speech," *Speech Communication*, vol. 13, pp. 231–237, 1993.
- [102] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system Mary: A tool for research, development and teaching," in *Proceedings of 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, Scotland, 2001.

- [103] H. Sheikhzadeh, E. Cornu, R. Brennan, and T. Schneider, "Real-time speech synthesis on an ultra low-resource, programmable system," in *Proceedings of ICASSP*, Orlando, Florida, 2002.
- [104] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proceedings of Eurospeech*, 2002.
- [105] K. Shinoda and T. Watanabe, "Mdl-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn.(E)*, vol. 21(2), pp. 79–86, 2000.
- [106] K. Silverman, M. E. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: a standard for labelling English prosody," in *Proceedings of ICSLP*, vol. 2, 1992, pp. 867–870.
- [107] K. Sjölander and J. Beskow, "Wavesurfer- an open source speech tool," in *Proceedings of ICSLP*, Beijing, China, October 2000, <http://www.speech.kth.se/wavesurfer>.
- [108] R. Sproat, *Multilingual Text-to-Speech Synthesis - The Bell Labs Approach*. Kluwer Academic Publishers, 1998.
- [109] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer Speech & Language*, vol. 15(3), pp. 287–333, 2001.
- [110] R. W. Sproat and J. P. Olive, "A modular architecture for multilingual text-to-speech," in *Progress in Speech Synthesis*. New York, USA: Springer-Verlag, 1997.
- [111] K. N. Stevens, *Acoustic Phonetics*. Massachuttes: The MIT Press, 1998.
- [112] "SVOX Ag, SVOX Technical White Paper," <http://www.svox.com>, pp. 4–5, 2003.
- [113] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using MLLR," in *Proceedings of The Third ESCA/COCOSDA workshop on Speech Synthesis*, 1998, pp. 273–276.
- [114] ———, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proceedings of ICASSP*, 2001.
- [115] P. Taylor, "Concept-to-speech synthesis by phonological structure matching," pp. 1403–1416, 2000.

- [116] P. Taylor and A. Black, "The architecture of the Festival speech synthesis system," in *Proceedings of 3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 147–151.
- [117] P. Taylor and A. Isard, "SSML: A speech synthesis markup language," 1997, *Speech Communication*, no. 21, pp. 123–133.
- [118] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden markov models based on multi-space probability distribution for pitch pattern modeling," in *Proceedings of ICASSP*, 1999.
- [119] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to English," 2002.
- [120] Y. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of ICASSP*, Istanbul, Turkey, 2000, pp. 1315–1318.
- [121] M. Vainio, T. Altsaar, M. Karjalainen, R. Aulanko, and S. Werner, "Neural network models for Finnish prosody," in *Proceedings of the XIVth International Congress of Phonetic Sciences*, 1999, pp. 2347–2350.
- [122] M. Vainio, A. Suni, and P. Sirjola, "Developing a Finnish concept-to-speech system," in *Proceedings of 2nd Baltic conference on Human Language Technologies*, 2005, pp. 201–206.
- [123] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," *IEEE Transactions on Information Theory*, pp. 260–269, April 1967.
- [124] "VoiceXML (W3C Voice Browser Working Group) homepage," <http://www.w3.org/Voice/>.
- [125] I. Witten, *Principles of Computer Speech*. Academic Press Inc., 1982.
- [126] K. Wothke, "Morphologically based automatic phonetic transcription," *IBM Systems Journal*, vol. 32, pp. 485–511, 1993.
- [127] J. Yamagishi, "Average-voice-based speech synthesis," Ph.D. dissertation, Tokyo Institute of Technology, 2006.
- [128] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems," Ph.D. dissertation, Nagoya Institute of Technology, 2002.
- [129] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Proceedings of Eurospeech*, vol. 5, 1997, pp. 2523–2526.

- [130] —, “Duration modeling in HMM-based speech synthesis system,” in *Proceedings of ICSLP*, vol. 2, 1998, pp. 29–32.
- [131] —, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” in *Proceedings of Eurospeech*, September 1999, pp. 2374–2350.
- [132] F. Yvon, “Self-learning techniques for grapheme-to-phoneme conversion,” 1994.

Publications

Publication 1

Moberg, M., Pärssinen, K., "Comparing CART and Fujisaki Intonation Models for Synthesis of US-English Names", *Proceedings of Speech Prosody 2004*, pp. 439-442, 23-26 March, Nara, Japan.

© 2004 ISCA. Reprinted, with permission.

Publication 2

Moberg, M., Pärssinen, K. "Cross-Lingual Phoneme Mapping for Multilingual Synthesis Systems", *Proceedings of International Conference on Spoken Language Processing 2004*, pp. 1029-1032, 17-21 October, Jeju Island, Korea.

© 2004 ISCA. Reprinted, with permission.

Publication 3

Moberg, M., Pärssinen, K. "Integrated Development Environment for a Multilingual Data Configurable Synthesis System", *Proceedings of International Conference of Speech and Computer 2005*, pp. 155-158, 17-19 October, Patras, Greece.

© 2005 ISCA. Reprinted, with permission.

Publication 4

Pärssinen, K., Moberg, M. "Multilingual Data Configurable Text-to-Speech System for Embedded Devices", *Proceedings of Multiling 2006*, 9-11 April, Stellenbosch, South Africa.

Publication 5

Pärssinen, K., Moberg, M. "Evaluation of Perceptual Quality of Control Point Reduction in Rule-Based Synthesis", *Proceedings of International Conference on Spoken Language Processing 2006*, pp. 2070-2073, 17-21 September, Pittsburgh, Pennsylvania.

© 2006 ISCA. Reprinted, with permission.

Publication 6

Moberg, M., Pärssinen, K. "Multilingual Rule-Based Approach to Number Expansion: Framework, Extensions and Application", *To appear in International Journal of Speech Technology (accepted 2006)*, Springer.

© 2006 Springer. Re-printed, with permission.

Publication 7

Pärssinen, K., Moberg, M., Gabbouj, M. "Reading Text Messages Using a Text-to-Speech System in Nokia Series 60 Mobile Phones: Usability Study and Application", *Technical Report, Tampere University of Technology Report 2006:3*, Tampere, Finland.

Publication 8

Pärssinen, K., Salmela, P., Harju, M., Kiss, I. "Comparing Jacobian Adaptation with Cepstral Mean Normalization and Parallel Model Combination", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2002*, Vol. 1, pp. 193-196, 12-17 May, Orlando, Florida.

© 2002 IEEE. Reprinted, with permission.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland