Jose Mariano Moreno Pimentel

Effects of Noise on a Speaker-Adaptive Statistical Speech Synthesis System

School of Electrical Engineering

Espoo 02.04.2014

Project supervisor:

Prof. Mikko Kurimo

Project advisor:

M.Sc. (Tech.) Reima Karhila

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF
THE FINAL
PROJECT

Author: Jose Mariano Moreno Pimentel

Title: Effects of Noise on a Speaker-Adaptive Statistical Speech Synthesis System

Date: 02.04.2014 Language: English Number of pages:9+56

Department of Signal Processing and Acoustics

Professorship: Speech and Language Processing Code: S-89

Supervisor: Prof. Mikko Kurimo

Advisor: M.Sc. (Tech.) Reima Karhila

In this project we study the effects of noise on a speaker-adaptive HMM-based synthetic system based on the GlottHMM vocoder. The average voice model is trained with clean data, but it is adapted to the target speaker using speech samples that have been corrupted by artificially adding background noise to simulate low quality recordings. The synthesized speech played without background noise should not compromise the intelligibility or naturalness.
A comparison is made to system based on the STRAIGHT vocoder when the background noise is babble noise. Both objective and subjective evaluation methods were conducted. GlottHMM is found to be less robust against severe noise. When the noise is less intrusive, the used objective measures gave contradictory results and no preference to either vocoder was shown in the listening tests. In the preference of moderate noise levels, GlottHMM performs as well as the STRAIGHT vocoder.

Keywords: speech synthesis, synthetic speech, TTS, HMM, noise robustness, TTS adaptation, vocoding, glottal inverse filtering, GlottHMM, STRAIGHT

Acknowledgments

This final project has been carried out at the Department of Signal Processing and Acoustics at Aalto University, supported by the Simple4All project. The work has also been contributed by the Speech Technology Group at the ETSI. Telecomunicación, UPM.

I would like to thank both groups and my respective supervisors in each group during the project, Mikko Kurimo, who was crazy enough to accept me in the group without knowing me, and Juan M. Montero for his help before and during the project.

Special thanks must be given to Ruben San-Segundo for introducing me in the speech world, for his selfless help, support and advice during these last years, and Roberto Barra for his crusade against spelling mistakes in my Spanish reports, his paternal lectures and last but not least, his amazing selfless help every time I asked him for.

I cannot miss the opportunity to thank Reima Karhila, my advisor in this project. Although being on the cover is such an indescribable honor, I want to thank him for his patience, for reading this project and sending me the corrections, although he might have been a little bit fussy in this task, for his help, his plotting skills with both Gnuplot and Matlab and for being less Finnish during my stay.

Finally, on a personal level I want to thank Arturo, my lab partner, whose complains have been very supporting during our stay in Finland, and my family, who are thanked as a group to avoid jealousy, for their support, help and love, without which I could have never done this project.

Otaniemi, 02.04.2014

Jose M. Moreno

Contents

Abstract ii

Acknowledgments iii

Contents iv

Symbols and Abbreviations ix

1 Introduction 1

2 History of Speech Synthesis 3
2.1 Acoustical-Mechanical Speech Machines . . . 3
2.2 Electrical Synthesizers: The Vocoder . . . 5

3 Speech Synthesis Systems 7
3.1 TTS Architecture . . . 7
3.2 Speech Synthesis Methods . . . 8
3.2.1 Formant Synthesis . . . 8
3.2.2 Articulatory Synthesis . . . 8
3.2.3 Concatenative Synthesis . . . 9
3.2.4 LPC-Based Synthesis . . . 9
3.2.5 HMM-Based Synthesis . . . 9

4 HMM-Based Speech Synthesis 11
4.1 Hidden Markov Models . . . 11
4.2 HMM-Based Speech Synthesis System . . . 13
4.2.1 System Overview . . . 13
4.2.2 Speech Parametrization . . . 14
4.2.3 Training of HMM . . . 14
4.2.4 Adaptation . . . 15
4.2.5 Synthesis . . . 18

5 Vocoders 19
5.1 Basics . . . 19
5.2 GlottHMM . . . 19
5.2.1 Analysis . . . 20
5.2.2 Synthesis . . . 20
5.2.3 GlottHMM with Pulse Library Technique . . . 22
5.3 STRAIGHT . . . 22
5.3.1 Analysis . . . 23
5.3.2 Synthesis . . . 23

6 Effects of Noise on Speaker Adaptation 25

7 Experiments **28**
7.1 Initial Experiments . . . 28
7.2 Feature Extraction . . . 35
7.3 Average Voice Model . . . 36
7.4 Adaptation . . . 36
7.5 Synthesis . . . 37

8 Evaluation **38**
8.1 Objective Evaluation . . . 38
8.2 Subjective Evaluation . . . 38

9 Results **40**
9.1 Objective Results . . . 40
9.2 Subjective Results . . . 41

10 Discussion and Conclusion **45**
10.1 Discussion . . . 45
10.2 Conclusion . . . 45

References **47**

Appendices

A GlottHMM Configuration **51**
A.1 GlottHMM configuration file . . . 51
A.2 Noise Reduction Parameters . . . 55

B Questions of the Listening Test **56**

List of Figures

1 Reconstruction of von Kempelen's speech machine made by Wheatstone [1] . . . 4
2 VODER synthesizer [2] . . . 5
3 General block diagram of a TTS system [3] . . . 7
4 6-state HMM structure. the states are denoted with numbered circles. State transitions probability form state i to state j are denoted by a_{ij}. Output probability densities of state i are denoted b_i and the observation generated at time instant t is o_t [4] . . . 12
5 Overview of an HMM-based speech synthesis system [5] . . . 13
6 Overview of an HMM-based speaker-adaptive speech synthesis system [6] . . . 16
7 On the left, CSMAPLR and its related algorithms, and on the right an illustration of a combined algorithm of the linear regression and MAP adaptation [6] . . . 17
8 Flow chart of the analysis made by GlottHMM [3] . . . 21
9 Synthesis block diagram of GlottHMM [7] . . . 21
10 Block diagram of the synthesis process made by STRAIGHT [7] . . . 24
11 Spectra for GlottHMM LSF (left), STRAIGHT MCEP components (middle) and FFT MCEP components (right) of a male speaker's vowel frame, with added babble (top) or band-limited Gaussian noise in the 300-700 Hz frequency band (bottom), shown in the figures in grey [8] . . . 26
12 Natural speech FFT spectra of clean speech, speech with babble noise, factory noise and machine gun noise . . . 28
13 Synthetic speech FFT spectra of clean speech, speech with babble noise, factory noise and machine gun noise after analysis and resynthesis with GlottHMM . . . 29
14 Histogram of the F_0 values of individual frames from the voices composing the average voice model, extracted with no lower or upper bounds . . . 29
15 SNR measures with $NOISE\_REDUCTION\_LIMIT = 4.5$ fixed and $NOISE\_REDUCTION\_DB$ from 5 to 50 . . . 30
16 MCD measures with $NOISE\_REDUCTION\_LIMIT = 4.5$ fixed and $NOISE\_REDUCTION\_DB$ from 5 to 50 . . . 31
17 SNR measures with $NOISE\_REDUCTION\_DB = 35$ fixed and $NOISE\_REDUCTION\_LIMIT$ from 0.5 to 6 . . . 31
18 MCD measures with $NOISE\_REDUCTION\_DB = 35$ fixed and $NOISE\_REDUCTION\_LIMIT$ from 0.5 to 6 . . . 32
19 Frame by frame representation of the natural speech with a babble background noise level of 10dB, resynthesized speech after analysis with GlottHMM not using the noise reduction module values in Appendix A.2 (set to true), resynthesized speech using the noise reduction module and SNR and MCD measures for the last synthetic sample 33

20 Frame by frame representation of the natural speech with a babble background noise level of 20dB, resynthesized speech after analysis with GlottHMM not using the noise reduction module values in Appendix A.2 (set to true), resynthesized speech using the noise reduction module and SNR and MCD measures for the synthetic samples . 33
21 SNR and MCD measures of a resynthesized sample with babble 10dB background noise using and not using the noise reduction module (values in Appendix A.2, set to true) 34
22 SNR and MCD measures of a resynthesized sample with babble 20dB background noise using and not using the noise reduction module (values in Appendix A.2, set to true) 34
23 Results of the AB test comparing different adapted voices obtained with the GlottHMM-based system 42
24 Results for the AB test comparing the performance of the GlottHMM-based system against the STRAIGHT-based one 43
25 Mean opinion scores (MOS) for the second part of the listening test. Median is denoted by the red line, boxes cover 25th and 75th percent percentiles, whiskers cover the data not considered outliers. The notches mark the 95% confidence interval for the median 44

List of Tables

1 Averaged fwSNRseg and MCD measures for 3 speakers. For the GlottHMM vocoder in clean conditions two results are shown: the below one uses the noise reduction system. All noise-affected systems use the noise reduction mechanism. The STRAIGHT values were calculated in [9] 26
2 Objective scores for the adapted test data using the F_0 calculated for each case with the GlottHMM-based system 40
3 Objective scores for the adapted test data using an external in the feature extraction F_0 calculated from the clean data with the GlottHMM-based system 41
4 Objective scores comparing GlottHMM and STRAIGHT 42
B1 Questions used in the subjective evaluation AB test 56
B2 Questions used in the subjective evaluation MOS test 57

Symbols and Abbreviations

Symbols

λ	Hidden Markov model
F_0	Fundamental frequency
$\mathbf{O}$	Observation sequence vector
P	Probability
$\mathbf{Q}$	State sequence vector

Abbreviations

CMLLR	Constrained Maximum-Likelihood Linear Regression
CSMAPLR	Constrained Structural Maximum A Posteriori Linear Regression
EM	Expectation-Maximization
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
HNR	Harmonic-to-Noise Ratio
LP	Linear Prediction
LPC	Linear Predictive Coding
LSF	Line Spectral Frequency
LSP	Line Spectral Pair
MAP	Maximum A Posteriori
MBE	Mixed multi-Band Excitation
MCD	Mel-Cepstral Distortion
MLSA	Mel Log Spectrum Approximation
MFCC	Mel-Frequency Cepstral Coefficient
MOS	Mean Opinion Score
MSD-HSMM	Multi-Space Distribution Hidden Markov Models
NSW	Non-Standard Word
PSOLA	Pitch-Synchronous OverLap-Add
SAT	Speaker-Adaptive Training
SMAP	Structural Maximum A Posteriori
SNR	Signal-to-Noise Ratio
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum
TEMPO	Time-domain Excitation extractor using Minimum Pertubatin Operator
TTS	Text-To-Speech

1 Introduction

There are many different kind of speech synthesis systems, and all of them pursued the same goal: produce natural sounding speech, which is the main goal of speech synthesis. As an extra requirement to this main goal, TTS systems aim to create the speech from arbitrary texts given as inputs, increasing the difficulty. It is easy to assume that a considerably amount of data is needed in order to cover all the possible sounds combinations in a given text. Moreover, the current trend in TTS aims towards generating different speaking styles with different speaker characteristics and emotions expressed with our voice, enlarging the spectrum of the characteristics of the voice to take into account and its differences depending on the context, increasing the amount of data needed to develop the final system.

It must be pointed out that among all the different techniques used nowadays to synthesize speech, some are not focused in maximum naturalness but they focus in intelligibility or high-speed synthesized speech. Although naturalness still a main issue, the final target, e.g. helping impaired people to navigate computers using a screen reader, forces to prioritize some other characteristics before naturalness.

Among the synthesis techniques, when talking about fulfilling the general requirements presented so far: naturalness, speaker characteristics, emotions, style, etc., unit selection technique and Hidden Markov Model (HMM) approaches stand out. Although unit selection synthesis provides the greatest naturalness, it does not allow an easy adaptation of a TTS system to other speakers or speaking styles, requiring a large amount of data due to the selection and concatenation used in this kind of synthesis, making this technique not suitable for example to embedded systems. On the other hand, HMM-based systems make easier to use adaptation techniques and require less memory, making them very popular nowadays.

We can find various vocoders currently being used in HMM-based systems, but the Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum (STRAIGHT) vocoder is the most commonly used and the most established one. However, due to the degradation in naturalness suffered in HMM-based systems, a new vocoder is being developed trying to solve this issue: the GlottHMM vocoder, which estimates a physically motivated model of the glottal signal and the vocal tract associated to it, producing a more natural voice.

So far memory requirements and the amount of data needed to build the system have been pointed as some of the weak points in speech synthesis systems. The amount of data is particularly important in unit selection synthesis systems. Sadly, collecting data is not an easy task since speech synthesis systems need high quality recordings covering different contexts. Moreover, when using speaker-adaptive systems, where an average voice model is built from several speakers to adapt it later to a new target speaker, certain amount of audio recordings will be needed from a substantial number of speakers. Adapting an average voice model, made out from high quality recorded audio of different speakers, with non high quality recordings would facilitate the access to a bigger number of target voices.

Noisy conditions were explored in speech recognition systems before being tested in synthesis system. Speech recognition is highly related to statistically speech syn-

thesis, specially HMM-based systems. For example, the analysis done to the audio recordings is the same in both cases, thus the same concepts used in recognition can be applied to speech synthesis systems. Nevertheless, speech recognition techniques under noisy conditions cannot satisfy all the needs of speech synthesis, so further research should be done in the future.

In this project the possibility of synthesizing speech from a model trained with noisy data will be explored. The aim is to adapt an average voice model made from high-quality training data, recorded in studio conditions, with noisy data, which is easier to obtain. HMM-based speech paradigm has been found to be quite robust on Mel-Cepstrum [9, 10] and Mel-LSP-based vocoders [11], but different adaptation techniques, vocoding techniques and noise present in the adaptation data can reduce quality, naturalness and speaker similarity and also add some background noise to the synthesized speech compared to the adaptation made from clean data.

A similar approach to this problem has been carried out in [9] using STRAIGHT vocoder. As GlottHMM targets on obtaining more natural voices, in this project we will study the effects of different types of noise present in adaptation data, using objective measures and subjective tests to evaluate the results. Besides, we will compare the performance made by GlottHMM vocoder with the one made by STRAIGHT vocoder in [9], trying to established which conditions benefit each vocoder against the other and learn about the level of acceptance of the synthesized voices observed in the subjective tests. To make the comparison as fair as possible, we will be working in Finnish with the same training and adaptation data.

2 History of Speech Synthesis

Speech synthesis is not a recent ambition in history of mankind. The earliest attempts to synthesize speech are only legends starring Gerbert d'Aurillac (died 1003 A.D.), also known as Pope Sylvester II. The pretended system used by him was a brazen head: a legendary automaton imitating the anatomy of a human head and capable to answer any question. Back in those days, the brazen heads were said to be owned by wizards. Following Pope Sylvester II, some important characters in mankind history were reputed to have one of these heads, such as Albertus Magnus or Roger Bacon [12].

During the 18th century, Christian Kratzenstein, a German-born doctor, physicist and engineer working at the Russian Academy of Sciences, was able to built acoustics resonators similar to the human vocal tract. He activated the resonators with vibrating reeds producing the the five long vowels: /a/, /e/, /i/, /o/ and /u/ [13].

Almost at the end of the 18th century, in 1791, Wolfgang von Kempelen presented his Acoustic-Mechanical Speech Machine [14], which was able to produce single sounds and some combinations. During the first half of the 19th century, Charles Wheatstone built his improved and more complicated version of Kempelen's Acoustic-Mechanical Speech Machine, capable of producing vowels, almost all the consonants, sound combinations and even some words.

In the late 1800's, Alexander Graham Bell also built a speaking machine and did some questionable experiments changing with his hands the vocal tract of his dog and making the dog bark in order to produce speech-like sounds [15, 13].

Before World War II, Bell labs developed the vocoder, which analyzed and extracted fundamentals tone and frequency from speech. In the 1950's, the first computer based speech synthesis systems were created and in 1968 the first general English text-to-speech (TTS) system was developed at the Electrotechnical Laboratory, Japan [2]. From that time on, the main branch of speech synthesis development has been focused on the investigation and development of electronic systems, but research conducted on mechanical synthesizers has not been abandoned [16, 17].

Speech synthesis can be defined as the artificial generation of speech. Nowadays the process has been facilitated due to the improvements made during the last 70 years in computer technology, making the computer-based speech synthesis systems lead the way supported by their flexibility and their easier access compared to mechanical systems. However, after the first resonators built by Kratzenstein, the fist speaking machine was built and presented to the world in 1791, and was obviously mechanic.

2.1 Acoustical-Mechanical Speech Machines

The speech machine developed by von Kempelen incorporated models of the lips and the tong, enabling it to produce some consonants as well as vowels. Although Kratzenstein presented his resonators before von Kempelen presented his speech machine, von Kempelen started his work quite before, publishing a book where he

described the studies made on human speech production and the experiments he made with his speech machine over 20 years of work [14].

The machine was composed by a pressure chamber, acting as lungs, a vibrating reeds in charge of the functions of the vocal cords and a leather tube that was manually manipulated in order to change its shape as the vocal tract does in an actual person, producing different vowel sounds. It had four separate constricted passages, controlled by the fingers, to generate consonants. Von Kempelen also included in his machine a model of the vocal tract with a hinged tongue and movable lips so as to create plosive sounds [15, 13, 18].

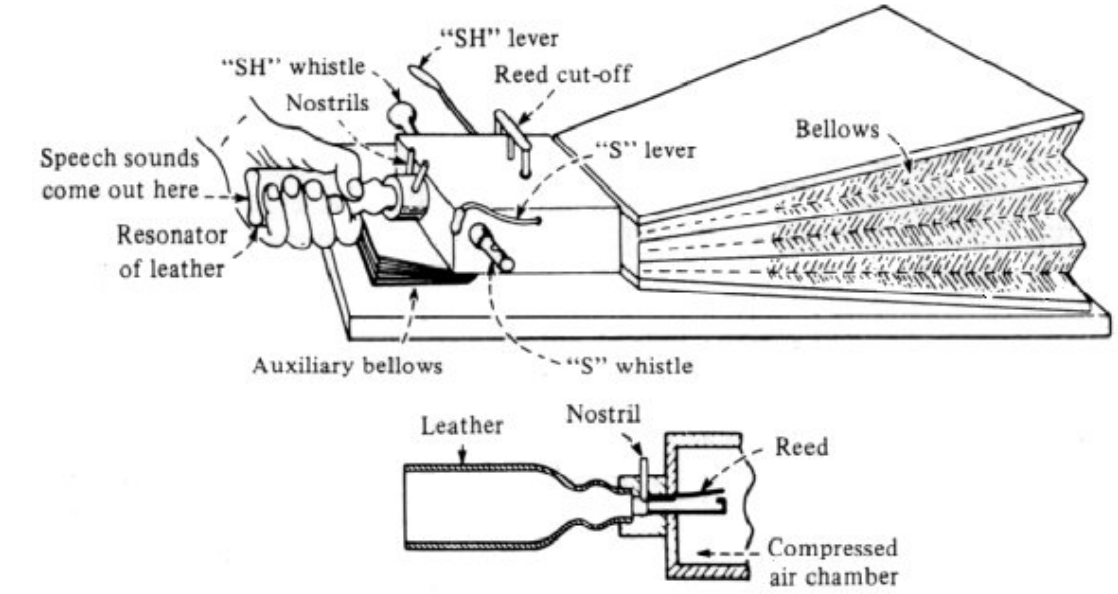

Figure 1: Reconstruction of von Kempelen's speech machine made by Wheatstone [1]

Inspired by von Kempelen, Charles Wheatstone built an improved version of the speech machine, capable of producing vowels, consonants, some combinations and even some words. In Figure 1 a scheme of the machine constructed by Wheatstone is presented. Alexader Graham Bell saw the reconstruction built by Wheatstone at an exposition and, encouraged and helped by his father, made his own speaking machine, starting his way towards the contribution in the invention of the telephone.

The research with mechanical items modelling the vocal system did not give any significant improvement during the following decades, leaving the door open to alternative systems to take the lead: the electrical synthesizers with a major breakthrough: the vocoder.

2.2 Electrical Synthesizers: The Vocoder

The first electrical device was presented to the world by Stewart in 1922 [2]. It consisted of a buzzer acting as the excitation followed by two resonant circuits modelling the vocal tract. The device was able to create single static vowel sounds with two lowest formants but not any consonant nor connected sounds. A similar type of synthesizer was built by Wagner [1], consisting on four parallel electrical resonators and excited by a buzz, capable of generating the vowel spectra when the proper combination of the outputs of the four resonators was made.

In New York's World fair 1939 [1, 2, 18], Homer Dudley presented what was consider the first full electrical synthesis device: the VODER. It was inspired by the vocoder developed at Bell Laboratoies some years earlier, which analyzed the speech into slowly varying acoustics parameters that drove the synthesizer to produce a an approximation of the speech signal. The VODER consisted of wrist bar for selecting a voicing or noise source and a foot pedal to control the fundamental frequency. The source signal was routed through ten band-pass filters controlling their output levels with the fingers [13]. In Figure 2 the VODER structure is graphically described. As you can imagine, it was not an easy task to synthesize a sentence on this device and the speech quality and intelligibility were far from acceptable, but he demonstrated the potential to produce synthetic speech.

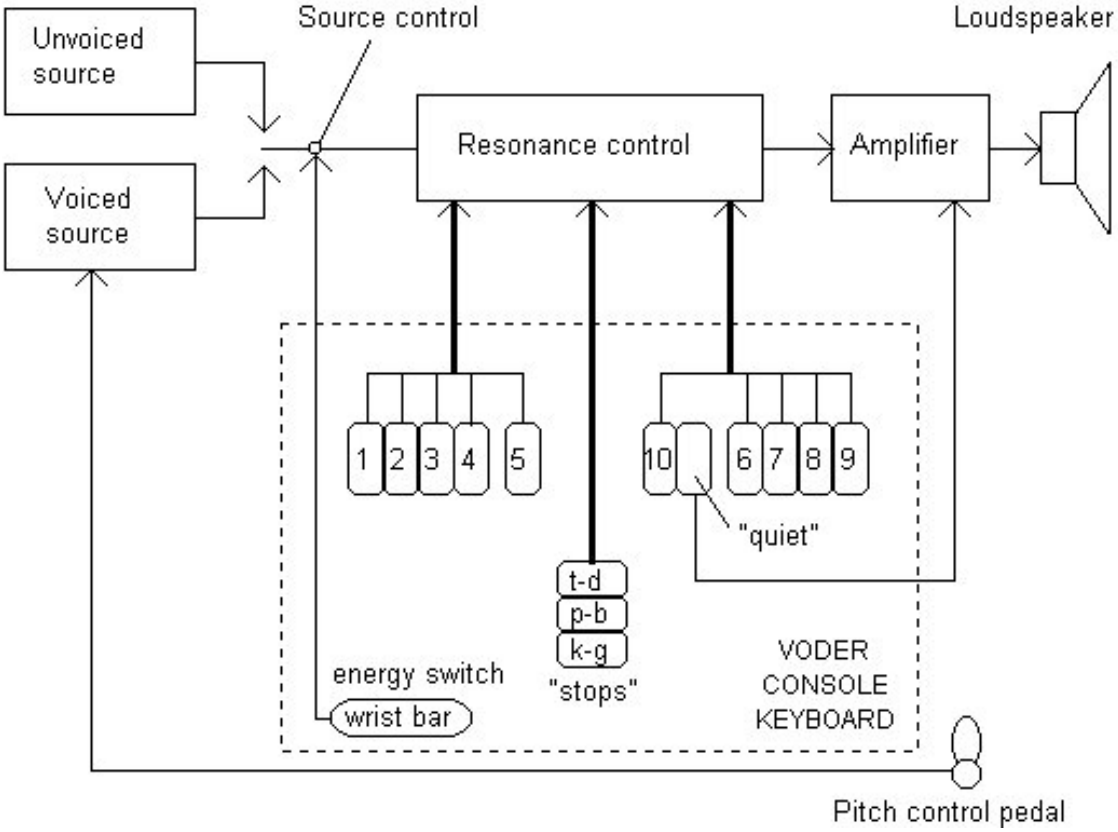

Figure 2: VODER synthesizer [2]

The demonstration of the VODER stimulated the scientific community and more people become interested in artificial speech generation. In 1951, Franklin Cooper lead the development of a Pattern Playback synthesizer [2, 18]. The device developed at the Haskins Laboratories used optically recorded spectrogram patterns on a transparent belt to regenerate the audio signal.

Walter Lawrence introduced in 1953 his Parametric Artificial Talker (PAT), the first formant synthesizer [2]. It consisted of three parallel electronic resonators excited by a buzz or noise and a moving glass slide converted painted patterns into six different time functions to control the three formant frequencies, voicing amplitude, noise amplitude and the fundamental frequency.

Simultaneously, the OVE I was introduced as the first cascade formant synthesizer. As its name suggest, the resonators in the OVE I were connected in cascade. A new version of this synthesizer was aired ten years later. The OVE II consisted on separate parts modelling the vocal tract to differentiate between vowels, nasals and obstruent consonants. It was excited by voicing, aspiration noise and fricative noise.

PAT and OVE developers engaged in a discussion about whether the transfer function of the acoustic tube should be modelled in parallel or in cascade. After a few years studying both systems, John Holmes presented his parallel formant synthesizer [2], obtaining a good quality in the synthesized voice.

Linear Predictive Coding (LPC) was first used in some experiments in the mid 1960's [15] and it was used in low-cost systems in 1980. The method was modified and nowadays is very useful and it can be found in many systems.

Different TTS systems appeared during the following years. Probably, the most remarkable one was the system developed by Dennis Klatt, the Klattalk, using a new sophisticated voicing source [2], forming along MITalk, developed at the M.I.T., the basis for many systems that came after them and also many ones used nowadays [13].

The modern technology used in speech synthesis involve quite sophisticated algorithms. As said in Section 1, HMM-based systems are very popular. Actually, HMMs have been used in speech recognition for more than 30 years. In Section 4 a detailed description of these systems is given, as is the technique used in this project.

HMM-based systems need to extract some features or parameters from the voice, and at that point is where the vocoder comes into action. Originally, the vocoder was developed to compress the speech in telecommunication systems in order to save bandwidth by transmitting the parameters of a model in stead of the speech, as they change quite slowly compared to a speech waveform. Despite its original objective, vocoders are the interface between the audio and the speech synthesis systems, extracting the features needed to model the system and synthesizing speech from the features generated by the system. In this project we will compare two vocoders, STRAIGHT and GlottHMM. They are both described in Section 5.

3 Speech Synthesis Systems

In this project we will use a HMM-based TTS system, but there are many different speech synthesis systems with their own advantages and disadvantages. In this section we will introduce the general architecture of a TTS system and diverse synthesis methods.

3.1 TTS Architecture

The main goal of a TTS system is to synthesize utterances from an arbitrary text. It is easy to notice that synthesizing from a text gives an extra flexibility to a synthesis system by allowing any reasonable input, in comparison to limited output systems such as GPS (Global Positional System) devices, but also an extra work has to be done to transform that text into the phonetic units required as inputs by the synthesizer. A general diagram of a TTS system is shown in Figure 3.

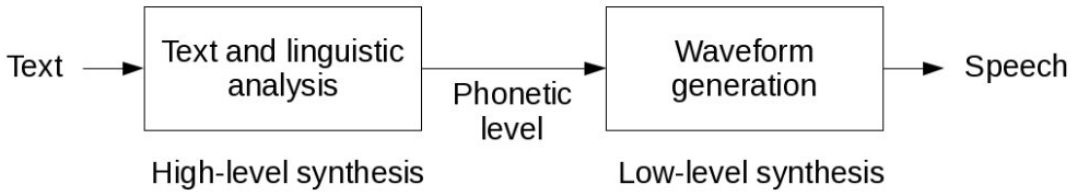

Figure 3: General block diagram of a TTS system [3]

The block representing the text and linguistic analysis is what differences a TTS system from other speech synthesis systems. The analysis made to the text has to generate the phonetic representation needed by the next component and predicting the desired prosody. Defining a larger set of goals for the speech synthesis system implies a more complex text and linguistic analysis. For example, trying to imitate the speaking style used by sports broadcaster in stead of synthesizing speech in a neutral style needs an extra function aiming to figure out the style of the input text, besides having constructed the corresponding model capable of producing speech mimicking the target style.

The main path followed by the text analysis includes a mandatory text normalization module. It is very important to normalize the text before trying to obtain its phonetic representation, to transform numbers, dates, acronyms and all the particularities that a language admit into a standardized form, called full-context labels representing the utterance on a phonetic-unit level based on the relations between phonemes, stress of each word, etc., accepted by the system. Also, this module is in charged of defining how similar spelled words are pronounced, e.g. the verb read has to different pronunciations whether is in the present tense or in the past tense. As it can be seen, text normalization is a complex problem that many researchers are

looking for a solution to. An interesting approach to convert non-standard words (NSWs) into pronounceable words based on a taxonomy built from several text types is discussed in [19].

Once the text is normalized, i.e. converted to plain letters, the structural properties of the text are analyzed and it is converted to a phonetic level. This last conversion is called the letter-to-sound conversion [20].

When the input text has gone through the first block represented in Figure 3, the low-level block generates predicts, based on the structural information and the prosodic analysis and tipically using statistical models, the fundamental frequency contour and phone durations. Finally, the speech waveform is generated by the vocoder.

3.2 Speech Synthesis Methods

The generation of the waveform can be carried out in several ways, thus, we can talk about different speech synthesis methods. As written in [3], the different methods can be divided in two categories attending to whether the speech is generated from parameter, i.e. completely artificial, or real speech samples are used during the process. From all the methods explained in this section, only concatenative synthesis uses real samples to synthesize speech.

3.2.1 Formant Synthesis

Formant synthesis is the most basic acoustic speech synthesis method. Based on the source-filter theory, which states that the speech signal can be represented in terms of source and filter characteristics [21], models the vocal tract with individually adjustable formant filters. The filters can be connected in serial, parallel or both. The different phonemes are generated by adjusting the center frequency, gain and bandwidth of each filter. Depending on the time intervals taken to do the adjustment, continuous speech can be generated. The source is modelled with voice pulses or noise.

Dennis Klatt's publication of the Klattalk synthesizer (see Section 2.2) was the biggest boost received by formant synthesis. However, nowadays the quality given by this kind of synthesizers is lower than other newer methods, such as concatenative systems. Even so, formant synthesis is used in many applications such as reading machines for blind people, thanks to its intelligibility [20].

3.2.2 Articulatory Synthesis

The aim of articulatory synthesis is to model the human articulatory system as accurately as possible, using computational physical models. Therefore, this is theoretically the best method in order to achieve high-quality synthetic voices. However, modelling as accurately as possible raises the difficulty. The main setbacks are the difficult implementation needed in an articulatory speech synthesis system and the computational load, limiting this technique nowadays. Despite its currently limita-

tions, articulatory models are being steadily developed and computational resources are still increasing, revealing a promising future.

3.2.3 Concatenative Synthesis

Concatenative methods use prerecorded samples of real speech to generate the synthetic speech. It is easy to deduce that concatenative synthesis stands out from other methods of synthesis in terms of naturalness of individual segments. There are several unit lengths, such as word, syllable, phoneme, diphone, etc, that are smoothly combined to obtain the speech according to the input text.

The main problem when using concatenative synthesis are the memory requirements. It is almost impossible to store all the necessary data for various speakers and contexts, making this technique the best one to imitate one specific speaker with one voice quality, but also makes it less flexible. It is difficult to implement adaptation techniques to obtain a different speaking style or a different speaker in concatenative speech. Apart from the storage problem, that thanks to the decrease in cost of digital storage and database techniques is becoming less serious, the discontinuities found in the joining points may cause some distortion even though the use of smoothing algorithms.

Concatenative systems may be the most widely used nowadays, but due to the limitations before commented, above all the flexibility problem, they might not be the best solution.

3.2.4 LPC-Based Synthesis

As in formant synthesis, in LPC-based synthesis utilizes source-filter theory of speech production. However, in this case the filter coefficients are estimated automatically from a short frame of speech, while in formant synthesis the different parameters are found for individual formant filters. Depending on the segment to be synthesized, the excitation needed is either a periodic signal, when synthesizing voiced segments, or noise, in case the segment is unvoiced.

Linear Prediction (LP) has been applied in many different fields for a long time and was first used in speech analysis and synthesis in 1967. The idea is to predict a sample data by a linear combination of the previous samples. However, LPC targets not to predict any samples, but to represent the spectral envelope of the speech signal.

Though the quality of basic LPC vocoder is consider poor, the more sophisticated LPC-based methods can produce high quality synthetic speech. The type of excitation is very important in LPC-based systems [3], but the strength of this method lays on its accuracy estimating the speech parameters and a relatively fast computational speed.

3.2.5 HMM-Based Synthesis

The use of HMMs in speech synthesis is becoming more popular. HMM-synthesis uses a statical model for describing speech parameters extracted from a speech

database. Once the statistical models are built, they can be use to generate parameters according a text input that will be use for synthesizing.

HMM-based synthesizers are able to produce different speaking styles, different speakers and even emotional speech. Other benefits are a smaller memory requirement and better adaptability. This last benefit is very interesting to us. While working with noisy data, limiting the amount of corrupted data used to train the system will probably affect positively to the quality of the synthetic speech obtained. Thus, constructing a high-quality average model and then taking profit of the adaptability of these systems to use the noisy data to train the adaptation transforms seems the correct approach. The data needed to train the adaptation transforms is always much lower than the training data used to built the average voice model.

On the other hand, naturalness is usually lower in HMM-based systems. But it must be said that these systems are improving very fast the quality of the synthetic speech obtained in terms of naturalness.

As in this project we will be using HMM-based TTS systems, they are going to be described with more detail in Section 4.

4 HMM-Based Speech Synthesis

Statistical parametric speech synthesis has grown in the last decade thanks to the advantages mentioned in Section 3.2.5: adaptability and memory requirements. In this section HMM-Based Speech Synthesis and HMM-based systems are explained.

4.1 Hidden Markov Models

HMMs can be applied to modelling different kinds of sequential data. They were first described in publications during the 1960s and the 1970s, but it was not until the 1980s when the theory of HMMs was widely understood and started to be applied in speech recognition and synthesis. Nowadays, HMMs are widely used along different fields and its popularity is still increasing.

As the name suggests, HMM-Based systems consist of statistical Markov models, where phenomena are modelled assuming they are Markov processes, i.e. stochastic processes that satisfy the Markov property. This Markov property can be described as a memoryless property: the next sample can be predicted from the current state of the system and the current sample, without using the past samples in the prediction.

Formally, HMMs are a doubly stochastic process formed by an underlying stochastic process that is not observable, i.e hidden, but can be observed through another set of stochastic processes that produce an observation sequence. Thus, the stochastic function of HMMs is a result of two processes, the underlying one is a hidden Markov chain with a finite number of states and the observable one consists on a set of random processes associated with each state.

An HMM can be defined as a finite state machine generating a sequence of time observations. Each time observation is generated by first making a decision to which state to proceed, and then generating the observation according to the probability density function of the current state. At any given discrete time instant, the process is assumed to be at some state. The current state generates an observation according to its stochastic process and the underlying Markov chain changes states with time according to the state transition probability matrix. In principle, the number of states, or order, of the underlying Markov chain is not bounded.

In Figure 4 a 6-state HMM structure in which at every time instant the state index can increase or stay the same, never decrease. A left-to-right structure is generally used for modelling systems whose properties evolve in a successive manner, as is the case of speech signal.

An N-state HMM is defined by a state transition probability distribution matrix, an output probability distribution for each state and an initial state probability distribution: $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{N}$, $\mathbf{B} = \{b_j(\mathbf{o})\}_{j=1}^{N}$ and $\Pi = \{\pi_i\}_{i=1}^{N}$ respectively. a_{ij} represents the state transition probability from state q_i to state q_j and $\mathbf{o}$ is the observation vector. A more compact notation for the model is: $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$.

There are three main problems associated to HMMs:

1. Finding an efficient way to calculate the probability of the observation sequence, $P(\mathbf{O}|\lambda)$, given an observation sequence $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T)$ and a model $\Pi = \{\pi_i\}_{i=1}^{N}$

2. How to choose an optimal state sequence $\mathbf{Q} = (q_1, q_2, ..., q_T)$ given the model and the observation sequence

3. How to maximize $P(\mathbf{O}|\lambda)$ by adjusting the model parameters

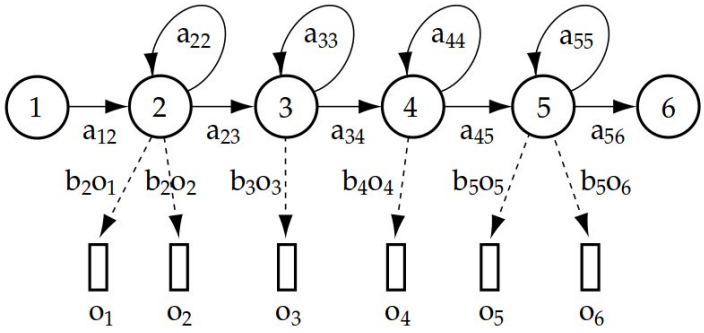

Figure 4: 6-state HMM structure. the states are denoted with numbered circles. State transitions probability form state i to state j are denoted by a_{ij}. Output probability densities of state i are denoted b_i and the observation generated at time instant t is o_t [4]

Finding the probability that the observed sequence was produced by the given model causes the first problem, but it can be used to score different models based on how well they match the given observation sequence. This probability is calculated by the equation:

$$P(\mathbf{O}|\lambda) = \sum_{all\ Q} P(\mathbf{O}|\mathbf{Q}, \lambda) \cdot P(\mathbf{Q}|\lambda) \tag{1}$$

Although the calculation of $P(\mathbf{O}|\lambda)$ is straightforward, it involves on the order of $2 \cdot T \cdot N^T$ calculations, which is far from being efficient. To reduce the computational cost of this calculation, this problem is usually evaluated with the Forward-Backward algorithm (see [22]), requiring $N^2 \cdot T$ calculations.

To solve the second problem we need to find the single best state sequence for a given observation sequence and a given model, i.e. we need to find $Q* = argmax_Q P(\mathbf{Q}|\mathbf{O}, \lambda)$. This is usually solved using the Viterbi-algorithm [23].

The third problem listed before is the most difficult one to solve. Solving the model which maximizes the probability of the observation sequence has no known analytical solution. In stead, gradient based algorithms and iterative algorithms such as the Expectation-Maximization (EM) algorithm [24] are being used for maximizing $P(\mathbf{O}|\lambda)$.

HMMs have the possibility of being extended with various features, increasing the versatility and efficiency depending on the needs of the user. For example, state tying, state duration densities and inclusion of null transitions are among the extensions proposed. More information about HMMs can be found in [22] and [25].

4.2 HMM-Based Speech Synthesis System

In this project an HMM-based speaker-adaptive synthesis system will be used to synthesize speech with different speaker styles. In [5] a general overview of speech synthesis based in HMMs can be found.

4.2.1 System Overview

The general overview of a HMM-based synthesis system is ilustrated in Figure 5.

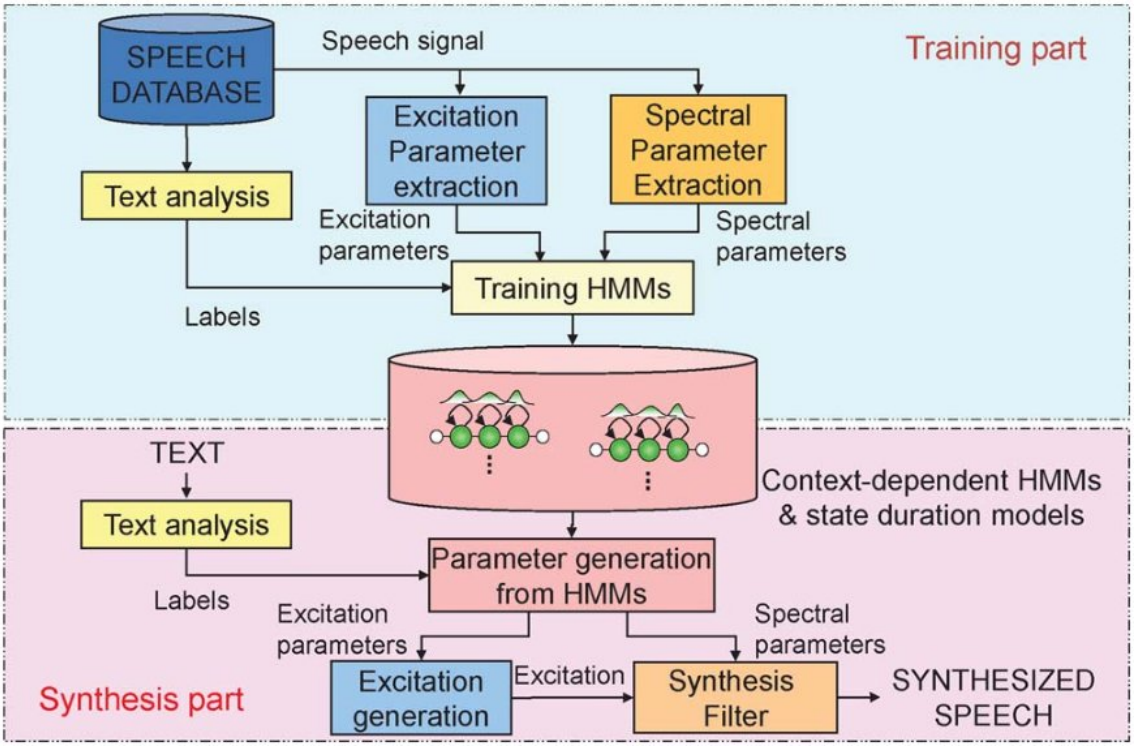

Figure 5: Overview of an HMM-based speech synthesis system [5]

An HMM-based system can be divided in two major parts: training and synthesis. In the training part, the vocoder extracts the speech parameters of every sample in the speech database and the labels containing the translation to the phonetic unit used, as explained in Section 3.1. Then, the obtained parameters are modeled in the framework of the HMM. The goal of the synthesis part is to produce a speech waveform according to the text input. This process begins with the analysis of the text, as in the training part, in order to concatenate the required HMMs for that particular sentence and generate the parameters to feed the synthesis module and generate the speech waveform.

In this project we will be using a speaker-adaptive system. Thus, there is an extra part not represented in the general overview of an HMM-based system shown in Figure 5: adaptation. Before the parameter generation a transformation is applied to the context-dependent HMMs and the state duration models, aiming to convert them into models of the target speaker. Adaptation makes synthesis with little data

from a specific speaker possible, but it must be done from a good average voice model, built out from several speakers, and the differences between the average voice model and the target speaker will highly affect the similarity between the real speaker and the synthetic voice [26]. In Section 4.2.4 an overview of a speaker-adaptive system is given and the adaptation technique used is explained.

The next sections explain the different steps that are done while constructing the HMM-based speech synthesis system.

4.2.2 Speech Parametrization

The first step of the training part is to extract from the speech signal a few parameters which function is to describe the essential characteristics of the speech signal as accurately as possible, compressing the original information. A very efficient way was found in separating the speech signal to source and filter [21], both represented by coefficients. Both, STRAIGHT and GlottHMM follow the source-filter theory, although it is not the only approach to this problem, it is a functional trade-off between the accurate but complex direct physical modelling and a reasonable analytic solution. This approach models the speech as a linear system where the ideal output is equivalent to the physical model, but the inner structure does not mimic the speech production physical structure.

In Section 5 the differences between the speech parametrization done by GlottHMM and STRAIGHT can be found, as they implement a different solution to this problem while following the same source-filter structure.

4.2.3 Training of HMM

Once the parametrization is done, the speech features obtained are used to train a voice model. During the training, maximum-likelihood estimation of the HMM parameters is performed.

The case of speech synthesis is a particular one. The F_0 values are not define in the unvoiced region, making the observation sequence of F_0 discontinuous. This observation sequence is composed of a 1-D continuous values representing the voiced regions and discrete values indicating the frames of the unvoiced regions. HMMs need to model both the excitation and spectral parameters at the same time, but applying both the conventional discrete and continuous HMMs to model F_0 cannot be done directly. Thus, to model the F_0 observation sequence, HMM-based speech systems use multispace probability distributions [27]. Typically, the multi-space distribution consists of a continuous distribution for the voiced frames an a discrete one for the unvoiced. Switching according to the space label associated with each observation makes possible to model variable dimensional vector sequences, in our case, the F_0 observation sequence. To keep synchronization between the spectral and the excitation parameters, they are simultaneously modelled by separate streams in a multistream HMM, which uses different output probability distributions depending on the features.

As shown in Figure 5, the training takes into account the duration and context to model the different HMMs. The duration modelling specifies for each HMM

a state-duration probability distribution. It models the temporal structure of the speech and it is in charge of the transitions between states, in stead of using fixed transition probabilities.

The context dependency of the HMMs is needed in speech synthesis to deal with the linguistic specifications. Different linguistics contexts, such as tone, pitch accent or speech stress among others, are used by HMM-based speech synthesis to build the HMMs. Spectral parameters are mainly affected by phoneme information, but prosodic and duration parameters are also affected by linguistic information. For example, within the contexts used in English, some of them are phoneme (current phoneme, position of the current phoneme within the current syllable, etc.), syllable or word contexts, such as the position of the current word within the current phrase [5].

Finally, it is important to note that there are too many contextual factors in relation with the amount of the speech data available. Increasing the speech data will increase the number of contextual factors and exponentially their combinations. Hence, limited amount of data will limit the accuracy and robustness of the HMMs estimation. To overcome this issue, tying techniques as state clustering and tying model parameters among several HMMs are used in order to obtain a more robust model parameters estimation. It must be noticed that spectral, excitation and duration parameters are clustered separately as they have different context dependency.

Once the HMMs are estimated regarding the considerations explained, the training part is finished and a model is built. If the model aims to reproduce one speaker, we would be talking about a speaker-dependent model. However, a speaker-adaptive system as the one used in this project aims to synthesize different speakers from one model as starting point. This model is called speaker-independent model, and the only difference with the speaker-dependent model so far in the HMM-based system construction is that the speech data is composed by several speakers to cover different speaker styles. However, when using speaker-independent models aiming to adapt to different speakers, a technique called speaker-adaptive training (SAT) is used to generate an average voice model by normalizing interspeaker acoustic variation [28, 29].

4.2.4 Adaptation

Figure 5 shows the overview of a general HMM-based speech synthesis system. In order to build a speaker-adaptive system, there is a third part that must be added to the structure before the synthesis: adaptation.

As commented previously, HMM-based systems are quite flexible, resulting in a good quality adaptive systems. Figure 6 illustrates a HMM-based speaker-adaptive system, hence, it shows the basic structure of both systems compared in this project.

The adaptation layer between the training and the synthesis part is the only difference between the structures of an adaptive and a non-adaptive system.

Many adaptation techniques are used in HMM-based speaker-adaptive systems, all of them targeting the same: transforming an average voice model to match a predefined target using a very small amount of speech data. Among the different

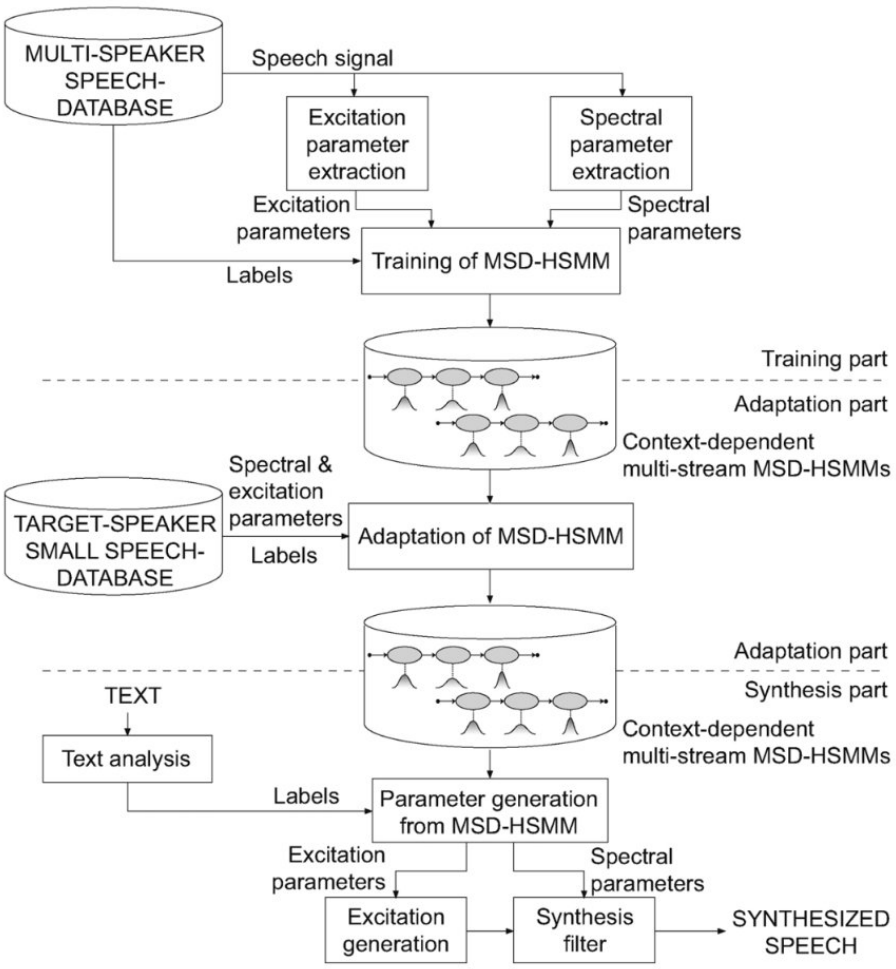

Figure 6: Overview of an HMM-based speaker-adaptive speech synthesis system [6]

targets we can find for example speaker adaptation or expressive speech. In [5] we can find several issues where adaptation techniques are helpful. Tree-based adaptation, where a decision tree is generated to estimate the transformation for each of the different units (e.g. for each phoneme), allows the use of several transforms in the adaptation algortihm.

Within the speaker-adaptive challenge, several techniques to approach a satisfying solution are available. [6] proposes an adaptation algorithm called constrained structural maximum a posteriori lineal regression (CSMAPLR) and compares several adaptation algorithms to figure out which one to use in which conditions.

The adaptations made during this project and in [9] use the CSMAPLR algorithm. This algorithm combines different adaptation algorithms in a defined order. The algorithms used are:

- Constrained maximum-likelihood liner regression (CMLLR)
- Maximum a posteriori (MAP)
- Structural maximum a posterior (SMAP)

When adapting in speech synthesis, it is important to adapt both the mean vectors and covariance matrices of the output and duration probability density functions, as the covariance is also an important factor affecting synthetic speech. This is the reason to use CMLLR in stead of the unconstrained version.

The CMLLR adaptation algorithm uses the maximum-likelihood criterion [30, 31] to estimate the transforms. The criterion works well when large amount of data is available. However, in the adaptation stage the amount of data is limited, a more robust criterion must be found: MAP. The basis of MAP algorithm are explained in [32] and an overview is given in [6].

In SMAP [33] the tree structures of the distributions effectively cope with the control of the hyperparameters. A global transform at the root node is estimated with all the adaptation data and then is propagated to the child nodes, whose transforms are estimated again using their adaptation data and the MAP criterion with the propagated hyperparameters. Finally, a recursive MAP-based estimation of the transforms from the root to the lower nodes is conducted.

CSMAPLR algorithm is obtained by applying the SMAP criterion to the CMLLR adaptation and using MAP criterion to estimate the transforms for simultaneously transforming the mean vectors and covariance matrices of state output and duration distributions. In Figure 7 this method is illustrated.

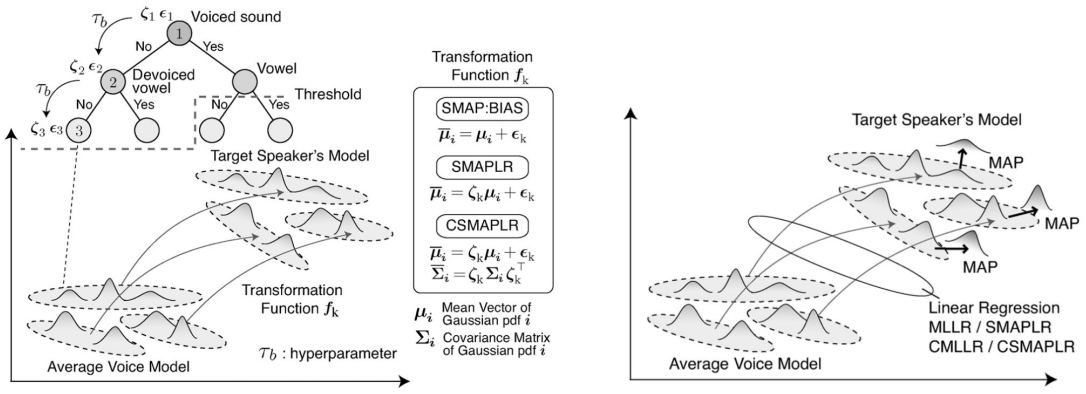

Figure 7: On the left, CSMAPLR and its related algorithms, and on the right an illustration of a combined algorithm of the linear regression and MAP adaptation [6]

Conclusions in [6] state that better and more stable adaptation performance from a small amount of data may be obtained by using gender-dependent average voice models and combining CSMAPLR adaptation with MAP adaptation, as shown in Figure 7.

In this project we make two rounds of CSMAPLR adaption followed by one round of MAP adaptation, in order to adapt the average voice model with noisy data. Each of the adaptations done generate models from which the parameters for synthesis can be generated. Based on the synthetic speech generated from every

different model, the unanimous conclusion is that the best quality is obtained when the three adaptation rounds are conducted.

4.2.5 Synthesis

The lower part of Figures 5 and 6 show the synthesis part of an HMM-based speech synthesis system. The first step is to convert the given text into a sequence of context dependent labels. Then, context-dependent HMMs are concatenated according to the labels calculated in the previous step, determining the duration of each state to maximize its probability based on its state duration probability distribution. Once the original sentence has been translated to context-dependent HMMs, a sequence of speech parameters is generated and using both the spectral and excitation parameters the speech waveform is produced by the correspondent vocoder.

5 Vocoders

The interface with both the natural speech and the synthesized speech is the vocoder. In this section, the fundamentals of the vocoder are presented and a detailed description of the two vocoders compared in this project is given.

5.1 Basics

The human speech is produced by regulating the air from the lungs through the throat, mouth and nose. The airflow from the lungs is modulated at the larynx by the vocal folds, creating the main excitation for voiced speech. The airflow is then filter by the vocal tract, formed by the pharynx and the oral and nasal cavities, acting as an acoustic time-varying filter by adjusting the dimensions and volume of the pharynx and the oral cavity.

The main functions of the vocoder are translating from natural speech to spectral and excitation parameters and from these features to synthetic speech. Thus, the vocoder should find a way to model the process involved in the human speech production in order to manage these features.

As established in Section 4.2.2, the source-filter theory is a functional trade-off behaving quite well in statistical speech synthesis. Hence, the basic vocoder could be the source-filter theory itself, modelling the source signal as a pulse train for voiced segments and white Gaussian noise for the unvoiced ones, i.e. impulse excitation vocoder.

The source-filter theory itself does not produce a high-quality synthetic speech. The very simple excitation modelling cannot correctly model some of the speech sounds. However, more complex vocoders as the compared in this project, GlottHMM and STRAIGHT, are also based on the source-filter theory, making the impulse excitation vocoder a standard to compare other vocoders with to test the quality. Apart from its benchmark functions, this simple vocoder has been historically significant for the development of statistical speech synthesis.

Among the different types of existing vocoders, in the following sections the two compared in this project are explained.

5.2 GlottHMM

GlottHMM is a glottal source modelling vocoder. The main characteristic of glottal source modelling vocoders is that they use estimated characteristics of a model of the glottal pulse in the determination of the exciting signal. GlottHMM was proposed by Tuomo Raitio in [3] and later improved [34].

The main idea in GlottHMM vocoder is to estimate a physically motivated model of the glottal pulse signal and the vocal tract filter associated with it. To achieve that, a method called Iterative Adaptive Inverse Filtering (IAIF) is used [35].

The advantage of the proposed method is that real glottal pulses can be used as the excitation signal when synthesizing, therefore providing a more natural synthetic

speech compared to pulse train excitation, making a quality improving. Moreover, the glottal flow spectrum can be easily adapted or modified.

A highly detailed description of GlottHMM can be found in [3] and [7]. In the next subsections an overview of the modules of GlottHMM is given, but it is not a deep description.

5.2.1 Analysis

During the analysis, GlottHMM first high-pass filters the speech signal from 70 Hz onwards. Then, the speech signal is windowed into fixed length rectangular frames, from which the log energy is calculated as a feature parameter.

Secondly, the IAIF algorithm is applied to each frame resulting in the LPC representation of the vocal tract spectrum and the waveform representation of the voice source. It calculates the LPC spectral envelope estimate of the voice source and along with the LPC estimate of the vocal tract is converted into a Line Spectral Frequency (LSF) representation [7]. The glottal waveform is used for the acquisition of the F_0 and the Harmonic-to-Noise Ratio (HNR) values for a predetermined number of frequency sub-bands.

The estimated glottal flow signal is used to produce the rest of the parameters. A voicing decision based on zero-crossings and low-band energy (less than 1 KHz) is made. For voiced frames, the F_0 value is calculated with an autocorrelation method. The HNR is calculated from the Fourier transform of the signal, evaluating the cepstrum of each frequency band. For each frequency band, the degree of harmonicity is determined by the strength of the cepstral peak (defined by F_0) in ratio to the averaged value of other quefrencies of the cepstrum. For unvoiced frames, the F_0 and HNR values are set to zero.

The feature vector extracted from the analysis made by GlottHMM is composed of:

- Excitation parameters: F_0, log energy, m HNR sub-bands and n order glottal source LSF
- Spectral parameters: p order vocal tract LSF

Usually 5 HNR sub-bands are used and the orders of the glottal source and vocal tract LSFs are around 10-20 and 20-30 respectively.

5.2.2 Synthesis

GlottHMM uses for the excitation generation a method based on the voiced/unvoiced decision in stead of using the traditional mixed excitation model for the excitation generation, as most of the state-of-the-art vocoders use. In Figure 5.2.2 the block diagram of the synthesis process of GlottHMM is shown.

For voiced frames, a fixed library pulse obtained by glottal inverse filtering a sustained vowel signal is interpolated to match the target F_0, using cubic spline interpolation, and its energy is set to match the target gain from the feature vector.

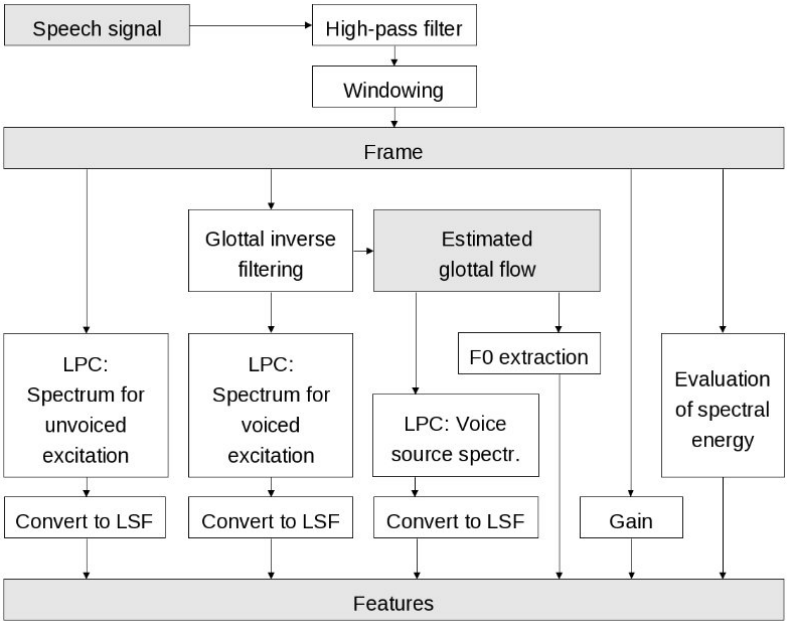

Figure 8: Flow chart of the analysis made by GlottHMM [3]

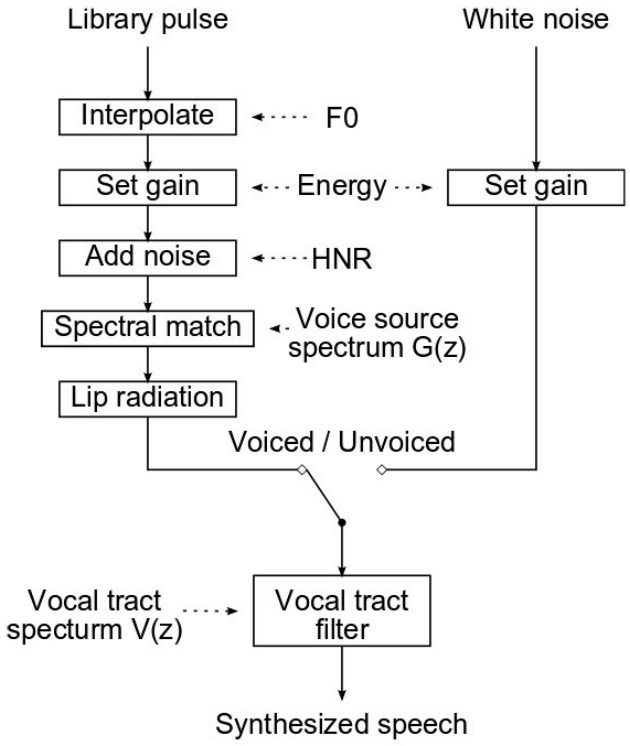

Figure 9: Synthesis block diagram of GlottHMM [7]

The next step is to conduct an HNR analysis similar to the one done in the analysis described in 8. Noise is added to the real and imaginary parts of the Fast

Fourier Transform (FFT) for every sub-band, according to the differences between the obtain and target HNR values, acting similar to the voiced excitation for voiced frames.

The spectrum of the library pulse is matched to the target glottal pulse in the feature vector. LPC analysis is performed for the spectral matching, post-filtering with the target synthesis filter. Finally, the lip radiation effect is added by filtering it with a fixed differentiator.

In the case of unvoiced frames, the excitation is generated with white Gaussian noise whose gain is set by the energy parameter in the feature vector.

The excitation is combined in the time domain by overlap-adding target frames and the final synthetic signal is generated by filtering the excitation with the vocal tract filter from the vocal tract LSFs in the feature vector.

5.2.3 GlottHMM with Pulse Library Technique

In [3] and [7] the pulse library for GlottHMM is described. This method was tested in the inital experiments to find out if it improves the quality over the single pulse excitation technique.

The results obtained with the pulse library method were clearly poorer than the ones using the single glottal pulse, therefore discarding the use of the pulse library. The causes of these poorer results when using the pulse library were not further investigated, as the technique presented code problems at the beginning that needed to be debugged.

5.3 STRAIGHT

STRAIGHT is a mixed multi-band excitation (MBE) vocoder. MBE vocoders use additional parameters with the F_0 to generate an accurate excitation signal reducing buzziness. The common characteristic in this kind of vocoders is that the parameters are extracted in a uniform way without case-specific adaptation.

STRAIGHT is the most established of the sophisticated vocoding methods. It was proposed by Hideki Kawahara [36] and has gone through extensive research and development [37]. Due to the tweaks and development done to STRAIGHT, several versions have been available throughout the time. The exact STRAIGHT configuration followed to built the system we compare the GlottHMM-based system to can be found in [9]. A general overview of the STRAIGHT vocoder is given in this section.

STRAIGHT was originally designed as a tool for speech transformation and accurate spectral envelope representation. Its original parameters are represented as Fourier transform magnitudes and their correspondent aperiodicity measurements. They cannot be used in HMM synthesis because of the high dimensionality. This issue was first overcome with the HMM-modified version of STRAIGHT proposed in [38], representing the spectral envelope as mel-frequency cepstral coefficients and averaging the corresponding aperiodicity measurements over five frequency sub-bands.

A detailed description of STRAIGHT can be found in [36], [38] and [7].

5.3.1 Analysis

STRAIGHT aims to an extraction of a smoothed spectral envelope, minimizing the effect of periodicity interference in the analysis frames, i.e. the STRAIGHT spectral envelope is essentially independent of the excitation.

To extract the spectral envelope the signal is windowed using two complementary F_0-adaptive windows with equivalent temporal and spectral solutions. Then, the original and complimentary magnitude spectrograms are calculated using both windows' functions and combined into a final spectrogram.

This method introduces an over-smoothing problem. To solve it, the use of a quasi-optimal smoothing function is proposed.

The aperiodicity measurements estimate the amount of harmonic information in relation to non-harmonic information in the signal. This is ideally done by warping each frame according to the phase of its fundamental component, making the warped signal to have a regular harmonic structure and calculating the ratios between lower and upper spectral envelopes. The upper spectral envelopes connects the spectral peaks while the lower one does the same with the valleys.

However, ideal solutions are usually not feasible and actually the unwarped aperiodicity measures are obtained by performing a table lookup of the lower-upper ratio from a database of known aperiodicity measurements. Then, its weighted average in relation to the speech power spectrum is calculated, resulting in the aperiodicity measurement.

To extract the fundamental frequency trajectory, STRAIGHT uses a specific pith extraction algorithm called Time-domain Excitation extractor using Minimum Pertubatin Operator (TEMPO) [37], based on the concept of instantaneous frequency.

The instantaneous frequency is extracted by the means of an analyzing continuous wavelet transform, which has the smallest amount of AM and FM properties at the fundamental frequency.

The HMM-adapted version of STRAIGHT transforms the STRAIGHT spectrum into a mel-frequency cepstral representation for statistical modelling. The aperiodicity measures are also represented in a compressed form.

The feature vector obtained with STRAIGHT consists of:

- Excitation parameters: F_0 and n order aperiodicity features
- p order STRAIGHT Mel-Frequency Cepstral Coefficient (MFCC)

5.3.2 Synthesis

STRAIGHT synthesis is carried out in a frame-by-frame basis by creating a mixed excitation signal of the length of two pulse periods, based on the F_0 and aperiodicity measurements. The harmonic pulse train is all-pass filtered with a randomized group-delay filter, reducing buzziness in the synthetic speech. The acquired mixed excitation signal is convolved with the minimum phase Mel Log Approximation (MLSA) filter, which is derived from the spectral MFCCs. To end, the Pitch-Synchronous Overlap-Add (PSOLA) algorithm [39] is applied to get the synthetic speech. This process is illustrated in Figure 10.

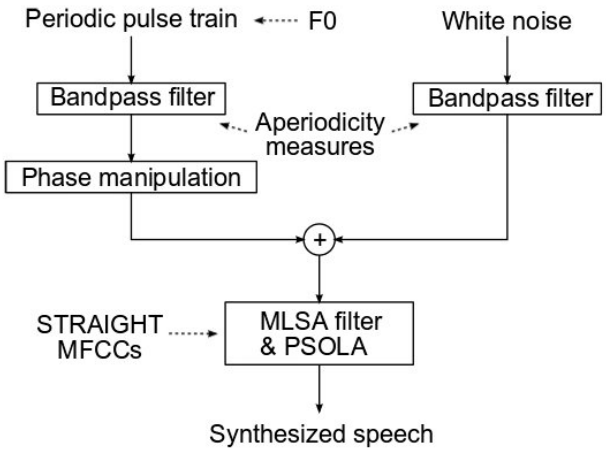

Figure 10: Block diagram of the synthesis process made by STRAIGHT [7]

The components for the mixed excitation are generated by sub-band filtering the voice and unvoiced parts, impulse train and white Gaussian noise respectively, in a separate way in the frequency domain.

The band-pass filters used are determined by the aperiodicity coefficients, having the resultant sub-bands the same average lower-to-upper envelope ratio as the correspondent aperiodicity coefficient.

The pulse train component is all-pass filtered to adjust the phase characteristics of the excitation.

The synthesis quality of STRAIGHT has a mean opinion score (MOS) of 3, while the impulse excitation vocoder (see 5.1) obtains a MOS of 2 [9].

6 Effects of Noise on Speaker Adaptation

Adapting from an average voice model is an efficient way of dealing with limited amount of data. Using the noisy data in the adaptation allows us to use a smaller amount of corrupted data than the needed if we use it in building a model starting from scratch. Therefore, less feature vectors corrupted by noise are used to estimate the final models.

Noise present in the adaptation data can add background noise to the synthetic speech or reduce its naturalness or similarity compared to adaptation done with clean data [9]. Depending on the vocoder used, the behavior displayed by the system may vary.

Previously, the speaker-adaptive HMM-based speech synthesis paradigm has been found to be quite robust on mel-cepstrum [9, 10] and Mel-LSP [11] based vocoders. In this work we focus on the GlottHMM vocoder and compare it to the STRAIGHT vocoder. Both vocoders model the excitation of the voice differently. The GlottHMM estimates glottal pulse via inverse filtering while STRAIGHT features pitch-adaptive extraction of spectrum (see Section 5).

Environmental noise can interfere with speech, as not all kind of noises can be filtered out of the signal without damaging the speech. Specially, noise signals that are time (and frequency) variant and occupy the same frequency bands provides the most difficult and challenging cases. When the noise cannot be removed the system must tolerate it. In this project three different noises are studied: babble, factory and machine gun noises. Among them, the most interesting one is the babble noise, because of its similar nature with speech and because it is the more common noise present when recording speech, as usually factories or environments involving gunfire are not chosen for, for example, interviewing people. As this is the most interesting case, the comparison between the GlottHMM and STRAIGHT vocoders is done for these cases.

Parametric vocoders represent the complete spectrum in some dozens of parameters. The ability to focus on the speech signal and smooth out the extra noise varies between vocoders.

Figure 11 shows how GlottHMM, STRAIGHT and the standard Fast Fourier Transform (FFT) react to vowel frames with added babble and Gaussian band-limited noise. It is obvious that all vocoders tolerate moderate amounts of Gaussian noise. Looking at the GlottHMM graphs, it can be seen that LSF models are locally affected by increased noise. The noise masks the speech signal locally reducing the accuracy of the LSF components in the area and even causing the formation of extra peaks. In the case of Gaussian noise, no effects are spotted out of the region where the noise was added.

If we look at the STRAIGHT graphs, the effects of the added Gaussian noise are more widely spread. The energy of the signal is redistribution with the decrease of the SNR, maintaining the shape of the spectral envelope in the higher frequency range although it is shifted. The effect of the noise is seen up to the 2000 Hz region.

As comparison, in the FFT spectra the effects of the Gaussian noise can be spotted up to the 1500 Hz region. However, the rest of the spectrum remains largely

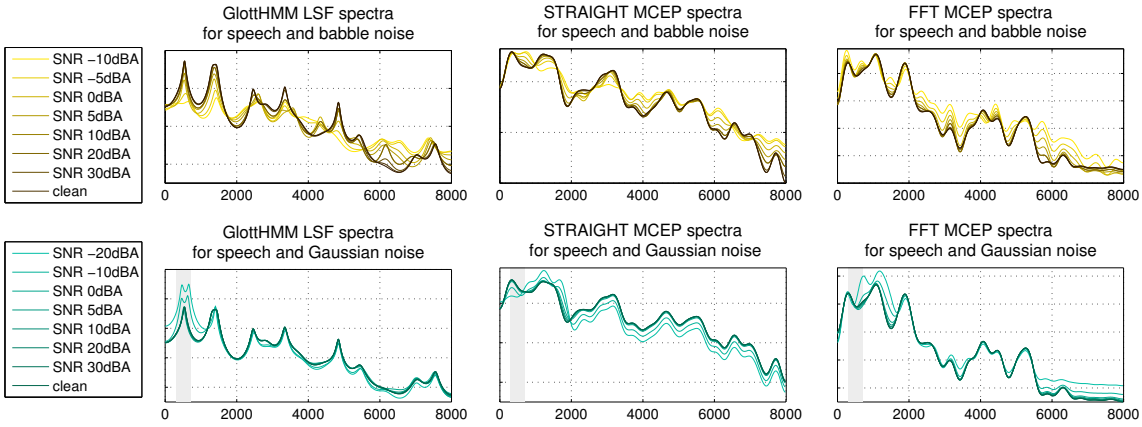

Figure 11: Spectra for GlottHMM LSF (left), STRAIGHT MCEP components (middle) and FFT MCEP components (right) of a male speaker's vowel frame, with added babble (top) or band-limited Gaussian noise in the 300-700 Hz frequency band (bottom), shown in the figures in grey [8]

without effect, except the higher frequencies, where the envelope is raised due to the jitter of the Gaussian noise.

On the other hand, the effects due to babble noise are very clear for all the vocoders. Valleys and peaks are moved and specially the LSF spectra shows more severe effects than the other two, with the appearance of sharp new peaks in mid and high frequencies.

		Original training data		GlotHMM resynth. data		STRAIGHT resynt. data	
Noise	SNR	fwS	MCD	fwS	MCD	fwS	MCD
Clean	-	35.0	0.0	14.6 15.9	1.0 2.1	15.5	1.5
Babble	20	20.7	1.1	15.6	2.3	14.0	2.0
	10	12.9	2.0	10.3	2.1	10.7	3.0
	5	9.5	2.5	8.3	2.5	8.4	3.4
Enhanced Babble	20	20.7	1.1	15.7	2.3	14.1	2.0
	10	13.3	1.8	11.3	2.1	11.0	2.6
	5	10.1	2.2	8.8	2.2	9.1	3.1

Table 1: Averaged fwSNRseg and MCD measures for 3 speakers. For the GlottHMM vocoder in clean conditions two results are shown: the below one uses the noise reduction system. All noise-affected systems use the noise reduction mechanism. The STRAIGHT values were calculated in [9]

The effects of babble noise are illustrated in Table 1, where the fwSNRseg and MCD results (see Section 8) for three test speakers' data analyzed and resynthe-

sized with both GlottHMM and STRAIGHT vocoders are shown. The analysis and resynthesis is done by synthesizing speech directly from the features obtained from the analysis of the samples, using only the different modules of the vocoders. For reference, the results of the original audio recordings are also shown. For GlottHMM in clean conditions two different results are shown: the first one does not use the noise reduction module (see Appendix A.2) and the second one is obtained using it. The SNR measures show that GlottHMM is able to reproduce the speech signal when the SNR is high enough, but the quality drops rapidly when the SNR of the signals gets low. It is very interesting that the MCD scores for GlottHMM do not behave as they are expected to. A noisier case (babble 10dB) is judge to be better than a cleaner one (babble 20dB). The performance obtained with STRAIGHT is the expected. Noisier cases are rated as lower quality cases by the two different measures.

The performance of the GlottHMM vocoder according to different values of the noise reduction parameters is investigated in Section 7.1.

7 Experiments

In mobile voice manipulation applications and in found data cases, it is mandatory to use audio recorded far from the ideal studio conditions, with the possibility of finding background noise. Nevertheless, in some vocoding and adaptation techniques noise present in the adaptation data can add background noise and produce distortion in the synthetic speech signal.

A GlottHMM-based speaker-adaptive statistical speech synthesis is built in this project, testing the effects of using noisy data in the adaptation. The different noises included in the adaptation data are: babble noise, factory noise and machine gun noise, with different signal-to-noise ratio (SNR). These noises where artificially added into clean data. The results will be compared to the ones obtained with the STRAIGHT-based system in [9].

From now on, we will refer to noise reduction configuration as the one set by the parameter values shown in Appendix A.2 (set to true).

7.1 Initial Experiments

The use of glottal pulses for HMM-synthesis was originally proposed due to the buzzy voice quality caused by simple excitation [40]. However, a proper modelling of the glottal pulse shape improves the quality in the case of lower fundamental frequency speakers. On the other hand, speakers with higher F_0, such as women, do not benefit from the pulse and the impulse excitation may be adequate for them. This particular behavior is the reason of working with a male average voice model, knowing that glottal inverse filtering approach flaws when synthesizing high-pitched voices.

The first step is testing the performance in analysis-resynthesis of GlottHMM when noisy data is used, in order to see if the performance of GlottHMM degrades when background noise is present in the samples. This is done using the analysis and synthesis modules of GlottHMM.

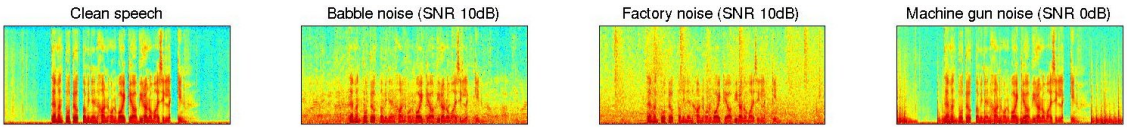

Figure 12: Natural speech FFT spectra of clean speech, speech with babble noise, factory noise and machine gun noise

Figure 12 shows the spectra of a natural speech sample in different environmental conditions while Figure 13 shows the spectra of the same samples resynthesized with GlottHMM.

As it can be seen, both the natural and the synthetic spectra have little differences between them. For example, the babble and machine gun noises have a bluer background, indicating less energy in low frequencies during the utterance. This is usually seen in synthetic speech, where the frequencies generated are bounded. After

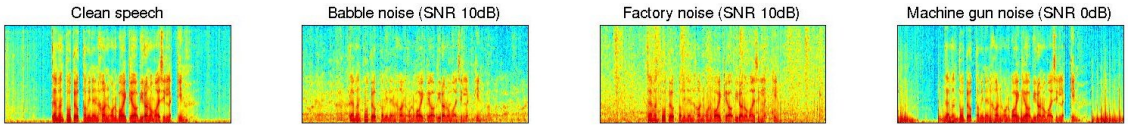

Figure 13: Synthetic speech FFT spectra of clean speech, speech with babble noise, factory noise and machine gun noise after analysis and resynthesis with GlottHMM

listening to the samples we could conclude that noise was not influencing the regular performance GlottHMM more than it influenced the performance of STRAIGHT in the experiments carried out in [9].

Another important issue is finding a correct configuration for GlottHMM. In Appendix A.1, the configuration file needed by GlottHMM can be found. As it can be seen, this file has a great amount of options to configure. However, thanks to previous experiments conducted and the advice of Tuomo Raitio, who developed GlottHMM, the tweaks to make in the configuration file are focused in noise robustness and some voice characteristics.

Some low F_0 problems were noticed during the first rounds of experiments. This problems consisted of frames where the voice sounded funny. To find out the details of this issue, a simple F_0 histogram plotting was made. Figure 14 presents the histograms of the voices used to build the average voice model, where low-frequency peaks can be pointed out in some of the voices.

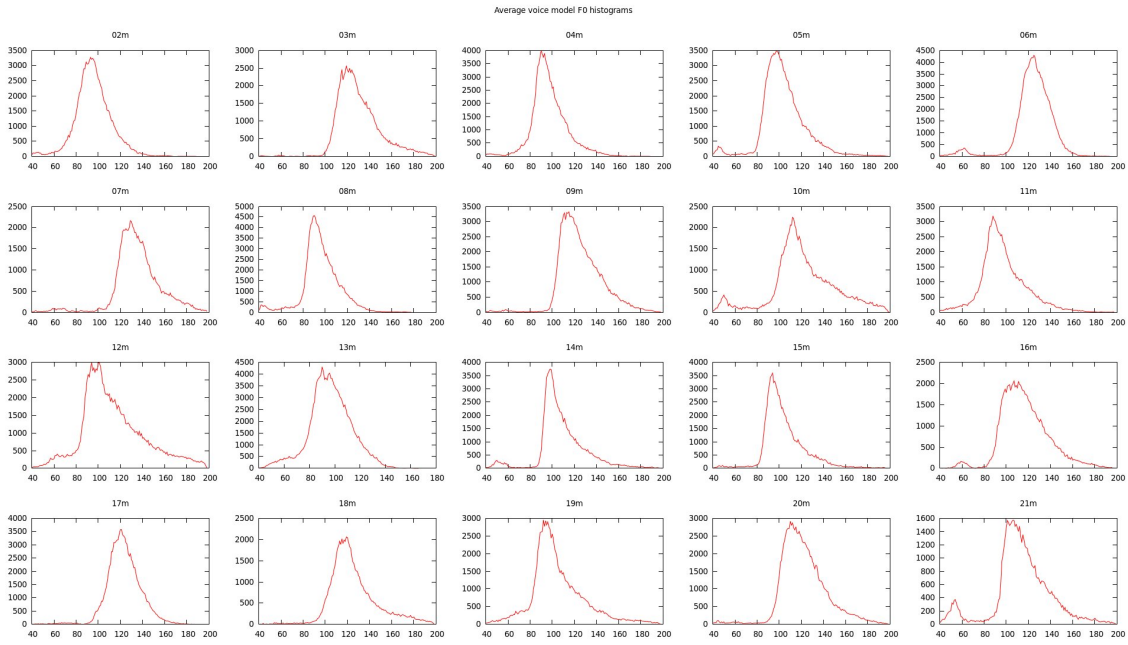

Figure 14: Histogram of the F_0 values of individual frames from the voices composing the average voice model, extracted with no lower or upper bounds

Solving this problem only required to extract again the features for the voices with low-frequency peaks. These peaks were found around 40-60 Hz in the training

data, used in the average voice model, and in the adaptation data. To eliminate them, in the configuration file the F_0 lower-limit was set to 65 Hz.

The last round of initial experiments conducted aim to find the best combination of the noise reduction parameters shown in Appendix A.2. These experiments consist on analysis and resynthesis of the noisy data varying the parameters in Appendix A.2 and carrying out the objective measures described in 8.1.

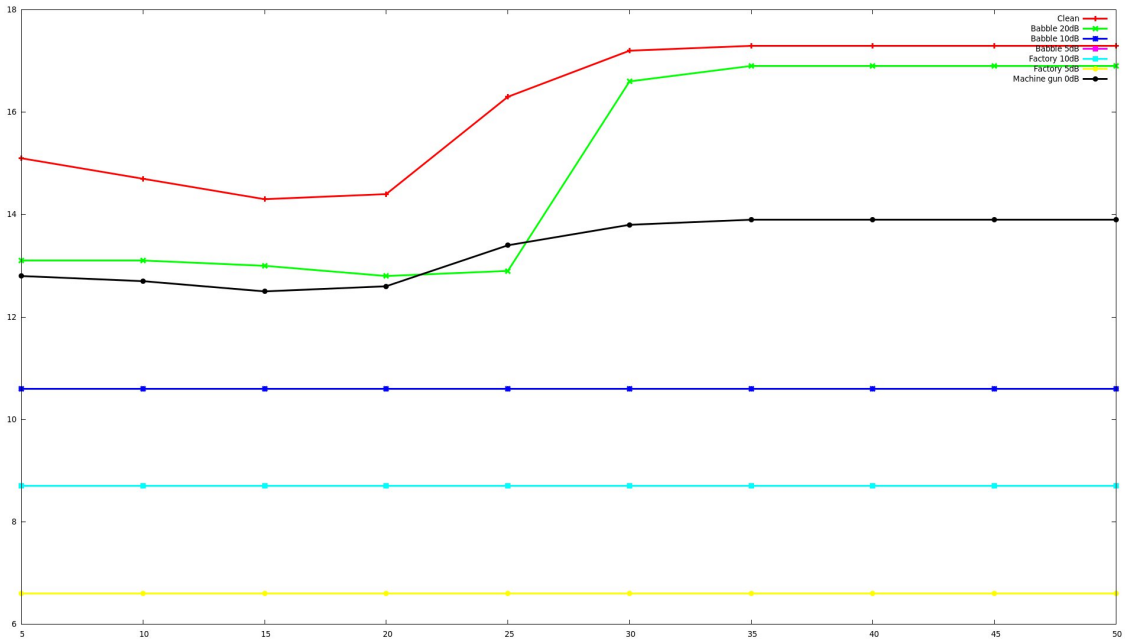

Figure 15: SNR measures with $NOISE\_REDUCTION\_LIMIT = 4.5$ fixed and $NOISE\_REDUCTION\_DB$ from 5 to 50

Figures 15, 16, 17 and 18 illustrate the evolution of the SNR and MCD measures when varying NOISE_REDUCTION_DB with a fixed NOISE_REDUCTION_LIMIT and vice versa. As it can be seen in Figure 15, for a fixed noise reduction limit the SNR measures increase significantly (higher SNR scores mean better quality) between 20 and 35dB noise reduction for the cases of clean and babble 20dB samples, reaching a limit. Also, there is a slight improvement in the case of machine gun 0dB noise, but in the rest of the cases no improvement is seen. As SNR scores increase MCD follows a similar patter (higher MCD scores mean worse quality). In Figure 16 it can be seen that the cases where the SNR was increasing are the ones with an increase of the MCD scores. The ones with steady SNR scores have no changes in the MCD either.

The case where NOISE_REDUCTION_DB is fixed and the variation is made in the NOISE_REDUCTION_LIMIT is presented in Figures 17 and 18. When increasing the NOISE_REDUCTION_LIMIT a steady increase of the SNR scores is only appreciable in the case of babble 20dB background samples. All the other cases remain stable, although a very small increase, not significant, can be spot for clean and machine gun 0dB cases. MCD scores follow the same pattern explained before. Babble 20dB has both increases in SNR and MCD. Not so big as in babble

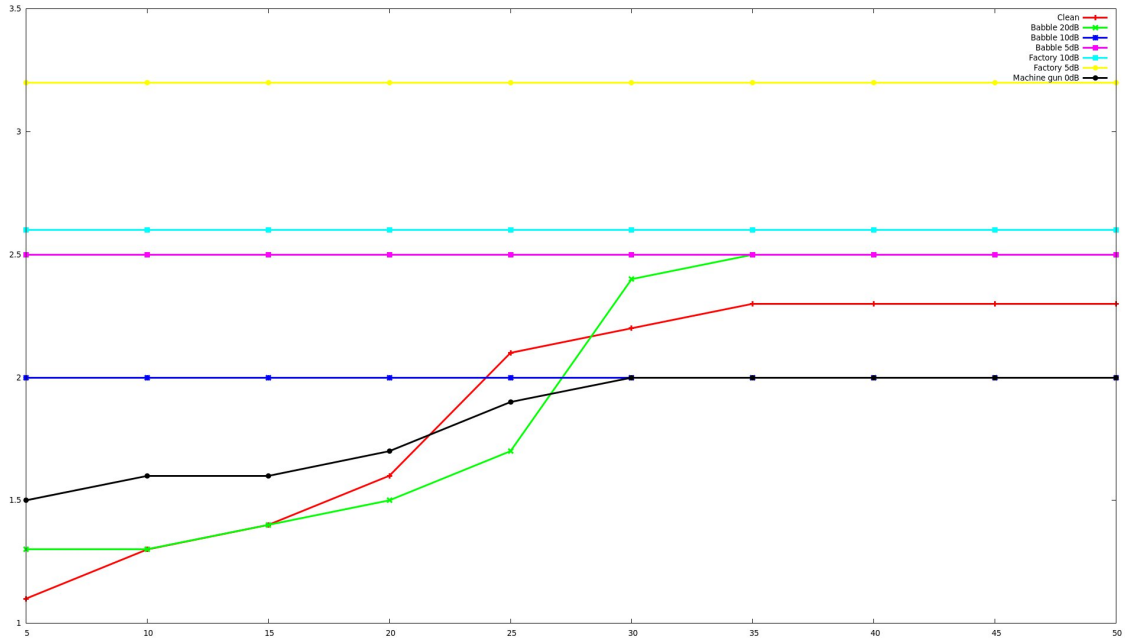

Figure 16: MCD measures with $NOISE\_REDUCTION\_LIMIT = 4.5$ fixed and $NOISE\_REDUCTION\_DB$ from 5 to 50

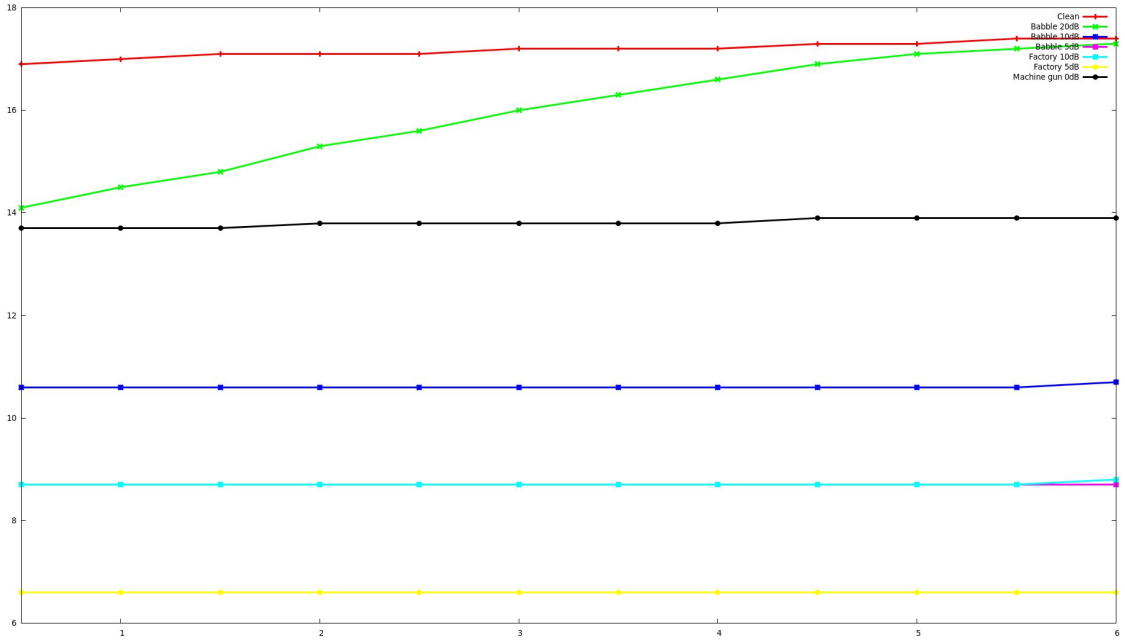

Figure 17: SNR measures with $NOISE\_REDUCTION\_DB = 35$ fixed and $NOISE\_REDUCTION\_LIMIT$ from 0.5 to 6

20dB case increases in MCD can be found also for the clean and machine gun cases, probably due to the small improvement appreciated in their SNR.

A frame by frame representation of the natural waveform, resynthesized wave-

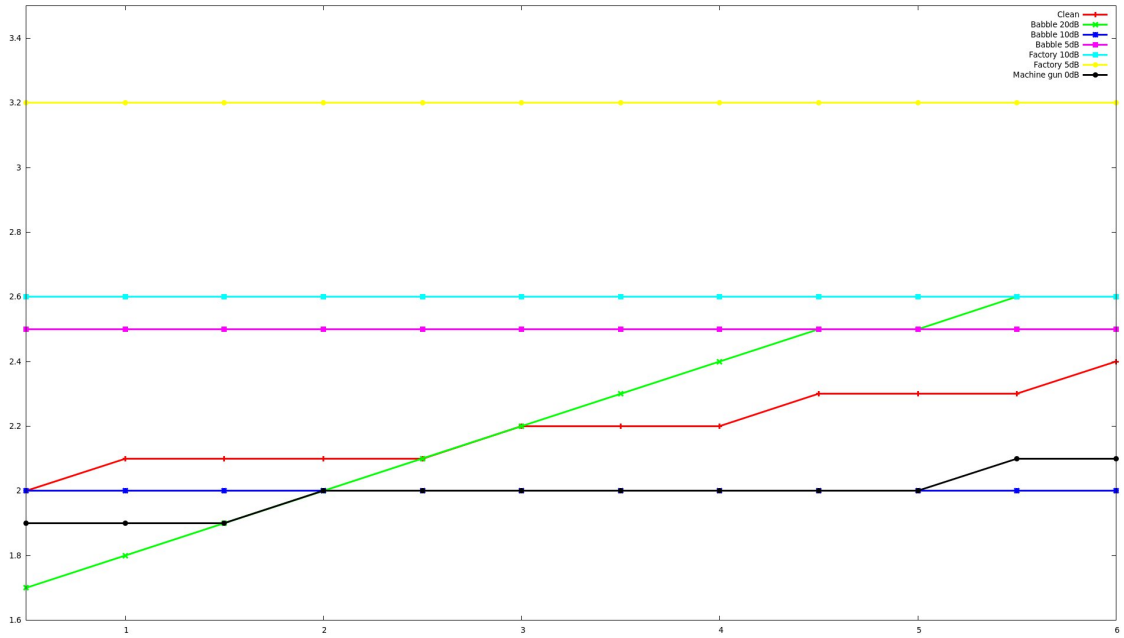

Figure 18: MCD measures with $NOISE\_REDUCTION\_DB = 35$ fixed and $NOISE\_REDUCTION\_LIMIT$ from 0.5 to 6

forms and SNR and MCD measures for the cases of babble 10dB and 20 dB background noise is shown in Figures 19 and 20.

In the case of the waveforms, we can see no difference when the data is corrupted with a 10dB level babble noise, while when the background noise level is 20dB the noise reduction module used by GlottHMM results in an improvement of the quality attending to cleaner synthetic waveform in speech silences and a better SNR score in some silences, visible for example when the SNR reaches its maximum value (35 dB) in the middle of the utterance.

A comparison between the measures done when resynthesizing using the noise reduction module and without using it is shown in Figures 21 and 22. No significant difference can be spotted when talking about the babble 10dB case. Nevertheless, in the case of having a babble 20dB background noise, we can point out different frames where the noise reduction module is clearly improving the SNR quality (careful with the different scales in the graph). Some frames reach the maximum SNR value (35 dB) while when not using the noise reduction module the same frames form a valley in the SNR graph. Other examples of this behavior can be found at the graph.

However, the MCD measures are contradictory. While the SNR increases, meaning a quality improvement, the MCD also increases creating the opposite effect, a quality decrease. This behavior has been noticed in the silences present along the utterance.

From all these initial experiments it can be said that the noise reduction included in the GlottHMM vocoder is not capable of carrying out its function when severe noise conditions are found. However, when the noise conditions are reasonable to record audio, these first experiments show that GlottHMM gets along quite well

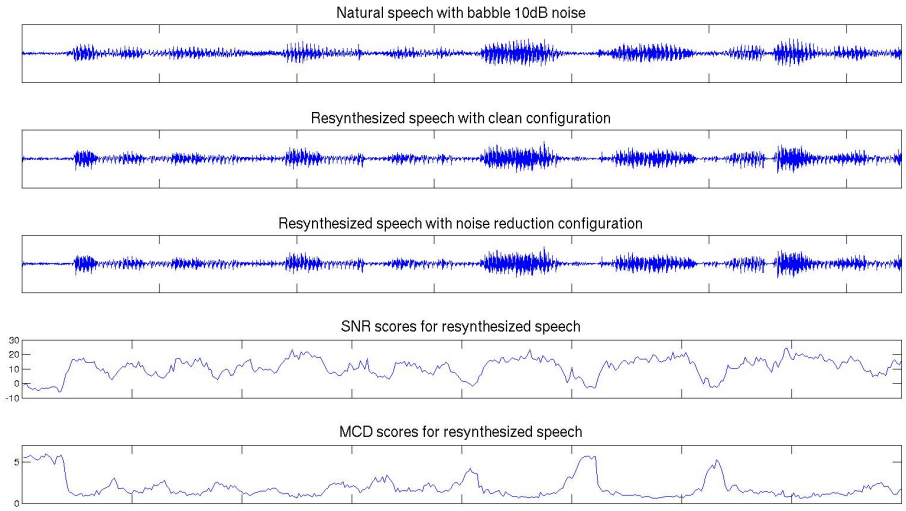

Figure 19: Frame by frame representation of the natural speech with a babble background noise level of 10dB, resynthesized speech after analysis with GlottHMM not using the noise reduction module values in Appendix A.2 (set to true), resynthesized speech using the noise reduction module and SNR and MCD measures for the last synthetic sample

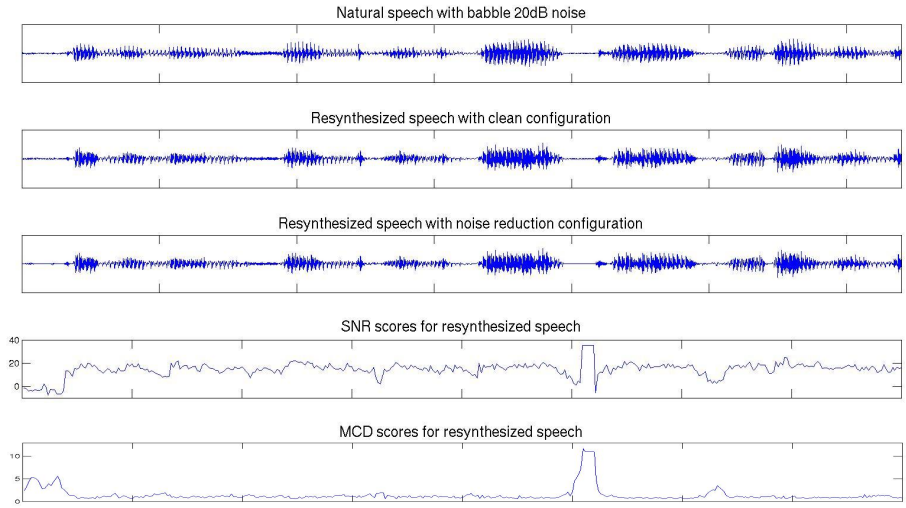

Figure 20: Frame by frame representation of the natural speech with a babble background noise level of 20dB, resynthesized speech after analysis with GlottHMM not using the noise reduction module values in Appendix A.2 (set to true), resynthesized speech using the noise reduction module and SNR and MCD measures for the synthetic samples

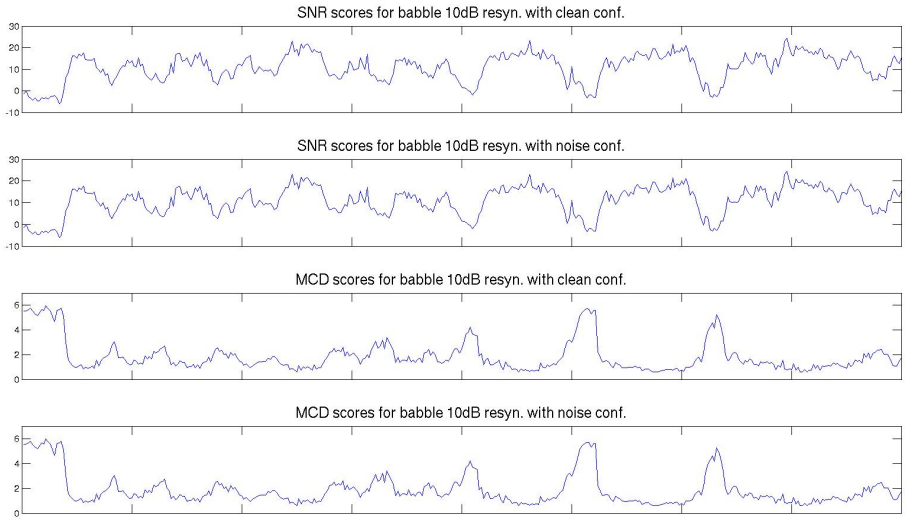

Figure 21: SNR and MCD measures of a resynthesized sample with babble 10dB background noise using and not using the noise reduction module (values in Appendix A.2, set to true)

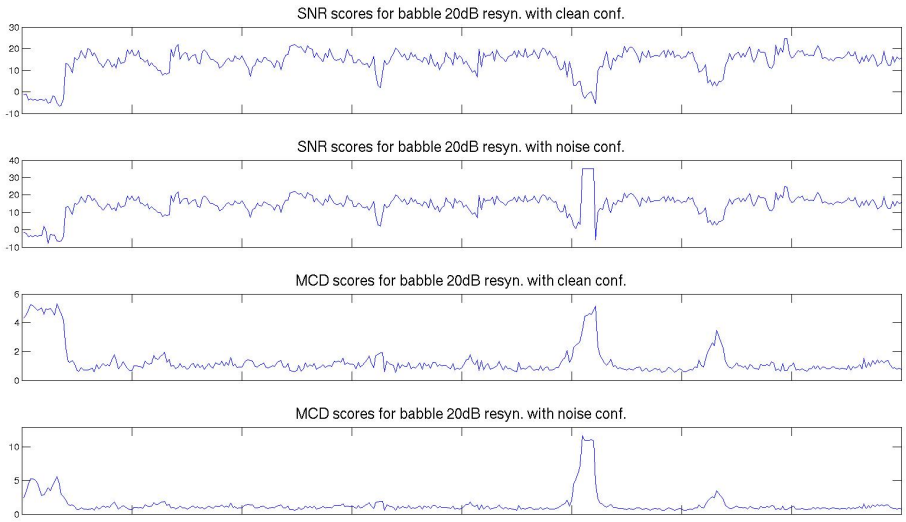

Figure 22: SNR and MCD measures of a resynthesized sample with babble 20dB background noise using and not using the noise reduction module (values in Appendix A.2, set to true)

in analysis and resynthesys, potentially improving the quality when adapting an HMM-based system.

7.2 Feature Extraction

The GlottHMM vocoder includes an analysis module that extract the features from audio files. However, the features extracted with this module are only the static features. Other features have to be calculated, such as the dynamic features, and in our case, noise robust features could be helpful in the adaptation process. These noise robust features are the Aurora features [41], calculated with the ETSI advanced front-end. For calculating the rest of the features, the dynamic ones and the Global Variance (GV) [42], used to solve over-smoothing problems, the scripts developed by Antti Suni, from the University of Helsinki, are used.

The final feature vector used in this system is a 183-dimension vector composed by:

- 30 LSF components + 1 energy
- 10 LSF source
- 5 harmonic-to-noise ratio (HNR)
- 1 F_0
- 14 Aurora components

From all the features except for the Aurora features the scripts used calculate the dynamic features (delta and delta-delta). The dynamic features for the Aurora are calculated with a snippet of the scripts used that does the delta-calculation. The addition of all the static and dynamic features gives the 183-dimension of the final feature vector.

It must be pointed out that during the training the Aurora features have no function, but they were investigated in [9] together with STRAIGHT features finding out that they improve the alignment when adaptation was not used. They were tested during this project in the training of the adaptation transforms, but regardless the results obtained in [9], our experiments show no improvement over the transforms trained with LSF features. Frequently, the alignment obtained with the Aurora features was poorer and noticeable by listening to the synthesized samples. Unfortunately, no improvement was obtained over the LSF features in the case of the GlottHMM-based system. The system learns about them during the training to be able to adapt according to the noise robustness provided by these features. However, during the experiments it was discovered that adapting with the Aurora features in stead of the LSFs not only does not produce any improvements but the results were clearly poorer. This was obviously noticed after adapting. Therefore, although the Aurora features are not used at all, they must be calculated in the adaptation data because of the construction of the average voice model, which embodies them in its feature vector composition.

7.3 Average Voice Model

As a proper modelling of the glottal pulse has been shown to improve the quality of low F_0 voices while higher F_0 ones do not benefit as much (impulse excitation can be adequate for them), we are focusing on male voices.

The average voice model of the speaker-adaptive speech synthesis system built in this work is trained on speech data from the Finnish PERSO corpus, with the features obtained as described in Section 7.2. 20 male voices from this corpus were used.

To train the model, a modified version of the EMIME 2010 Blizzard Entry [43] was used, using SAT and 3 reclustering iterations, using the configuration for GlottHMM attached in Appendix A.1, where the noise reduction module is not used: clean configuration. The third iteration gives us the multi-space distribution hidden semi-Markov models (MSD-HSMM) forming the average voice model.

The STRAIGHT voice is trained identically to [9].

7.4 Adaptation

Once the average voice model trained with high-quality data is ready, we can use the noisy data to adapt to different target speakers.

Three different types of noise were used in this project: Babble, factory and machine gun noise, with different SNR levels. The noisy samples were obtained adding noise from the NOISEX-92 corpus [44] to utterances from the EMIME corpus [45]. 105 utterances from each of the three target speakers conform the training of the adaptation transforms. This setting is similar to the one used in [9]. The only difference in the process is the vocoder used, as we want to compare GlottHMM against STRAIGHT.

The adaptation transforms are calculated using a combined algorithm with linear regression and MAP adaptation (see 4.2.4). Two rounds of CSMAPLR followed by one round of MAP adaptation conform the combined algorithm.

The regression trees were limited to 64 leaf nodes to obtain robust adaptation transforms, after discovering by trial and error that higher values carried a decrease in the quality of the synthetic speech. Due to the noise present in the data, more data is needed in one node to strengthen the transforms. Moreover, to correct some mistaken phoneme's durations a realignment of the test labels using the average voice model was done prior to the synthesis.

Finally, speech enhancement based on non-negative matrix factorization [46] was carried out to test the benefits, if any, obtained. The adaptation procedure with the enhanced data is similar to the noisy cases.

The values of the noise reduction parameters shown in Appendix A.2 are the values used in the noise reduction configuration of GlottHMM. When the energy of the frame is below 4.5dB (noise reduction limit) the gain of the frame is reduced by 35dB (noise reduction).

In the adaptation of all the noisy cases the noise reduction configuration is used. Also, adaptation using clean data is done using both the clean configuration and the

noise reduction configuration. Using both configurations is done to compare due to the results obtained in Section 7.1.

7.5 Synthesis

The voices of three different male speakers under different noise conditions were synthesized during this work. The clean and all the noisy cases' features were generated based on the models obtained by adapting with the F_0 estimations corresponding to each case. However, the problems with the estimation are clearly audible when comparing to the synthesis done from models adapted using an external F_0 extracted from the clean data of each speaker. The comparison to the STRAIGHT synthesized speech is done using the models adapted with the external F_0 synthetic samples obtained with the GlottHMM-based system, focusing on the effects of noise in the spectral components. While the subjective evaluation only includes the external F_0 case, the objective evaluation is conducted for both cases.

Both systems where constructed using the HTK toolkit [47]. The GlottHMM-based system uses a single, generic pulse was used to produce speech in stead of a pulse library. The parameter trajectories for synthesizing are generated with the HMGenS tool from the HTK toolkit according to the models obtained after the adaptation and the labels generated from the input. These labels are realigned before the feature generation. The realignment is done with the models from the average voice, as problems with the durations of some phonemes where noticed when realigning according to the adapted models. Once the features are produced, the synthesis modules of both GlottHMM and STRAIGHT vocoder used them as the input to generate the synthetic speech.

8 Evaluation

To evaluate the different synthetic voices we must implement to different types of measures: objective and subjective. The objective measures aim to provide quantitative scores indicating the quality of the analyzed samples according to defined parameters. Meanwhile, the subjective tests conducted for speech synthesis provide the listeners' opinions and preferences.

8.1 Objective Evaluation

Two different measures are used in this project.

The first one is a common measure for objective evaluation of speech synthesis quality: the mel-cepstral distortion (MCD) calculated between natural speech sentences and the corresponding synthesized sentences [48]. It is calculated for D-dimensional features as

$$MCD = \frac{1}{M}\sum_{m=1}^{M}\sqrt{2\sum_{d=0}^{D-1}(c(d,m)-\hat{c}(d,m))^2}, \tag{2}$$

where $\hat{c}(d,m))$ and $c(d,m)$ are the dth coefficient in test and reference mel-cepstra in time frame m, and M denotes the number of frames.

As in [9], in this project the MCD is used in conjunction with a Frequency Weighted Segmental SNR (fwSNRseg) based on the energy of the segments [49], calculated as

$$fwS = \frac{10}{M}\sum_{m=1}^{M}\frac{\sum_{j=1}^{K}W(j,m)log_{10}\frac{X(j,m)^2}{(X(j,m)-\hat{X}(j,m))^2}}{\sum_{j=1}^{K}W(j,m)}, \tag{3}$$

where $\hat{X}(j,m)$ is the test signal value in the jth mel filter channel in time frame m, $X(j,m)$ is the reference signal value in the same mel filter channel, and $W(j,m) = X(j,m)^{\gamma}$ with $\gamma = 0.2$ as in [49].

These measures were calculated based on a two-second sample extracted from the middle of each utterances. Unfortunately, synthetic speech may introduce excess frames, worsening the alignment between the synthetic and natural speech used in the measures. Therefore, the comparisons between the test and reference samples is done with a variable frame delay ([-10,...,10]). The best result is assumed to be the best aligned and reported.

8.2 Subjective Evaluation

In [50] a test for noisy synthesis cases is proposed based loosely on the ITU-T recommendation for noise-suppression algorithms [51], when background noise added

to the synthetic speech can mask some artifacts. The test framework consists on two parts implemented as a web interface that presented the questions in Appendix B. The first part of the test is an AB test to find out with of the systems, GlottHMM-based or STRAIGHT-based, is preferred for everyday use. The statistical test used to verify the significance of the AB test results is the binomial test, where the two options are mutually exclusive with the same initial probability. However, they could sound the same to the listeners and a third option was added, indicating they do not find any difference between the samples. Statistically, every time someone chooses the no difference option it counts as half vote for both of the options in the question. Moreover, to find out significance among the results, the accumulative probability must be calculated and if the value of the less probable of the tested options is equal or less than 0.05 we can place in the results in the significance region of the binomial distribution [52].

In the second part, a mean opinion score (MOS) test where the listeners were asked to rate either the speech signal, the background quality or the similarity to the natural correspondent voice.

As the silences models could be affected with the adaptation procedures used in this work, i.e. synthesized noise can be introduced at the beginning and end of the synthetic utterances, allowing listeners to adapt to the background noise, the ITU-T recommendation [51] was followed and eliminating those noises.

9 Results

In this section we present both the objective and the subjective results obtained. Each result falling more than 2 standard deviations away from the mean is consider an outlier and taken out from the final calculations. In the objective measures the outliers are not shown. For the subjective results, the outliers can be found in the MOS score graph (Figure 25).

9.1 Objective Results

The results in this section are the average of the three male speakers.

Table 2 shows the objective scores obtained for the synthetic speech of the GlottHMM-based system without using the external F_0 in the feature extraction. The clean samples have been synthesized with the clean configuration while in all the noisy cases the noise reduction configuration was used.

On the other hand, in Table 3 the results for all the noisy cases adapted using the external F_0 are presented. In this case, there is no clean row, as the F_0 is calculated from the clean samples. All the adaptations in this table were done with the noise reduction configuration.

As it can be seen, there are no significant differences between both approaches, what could lead us to think we can use either of them and obtain the same results. However, when listening to the synthesized audio samples it becomes pretty clear that when not using an external F_0 there is a huge quality drop. Therefore, not using an external F_0 is been rejected from this point on.

Noise	SNR	fwS	MCD
Clean	-	9.0	1.8
Babble	20	10.6	3.0
	10	7.5	2.7
	5	6.3	2.6
Factory	10	6.8	3.0
	5	5.3	3.2
Machine gun	0	9.3	2.7
Enhanced Babble	20	10.8	3.0
	10	8.4	2.8
	5	6.9	2.8
Enhanced Factory	10	8.7	3.2
	5	7	3.3

Table 2: Objective scores for the adapted test data using the F_0 calculated for each case with the GlottHMM-based system

In Table 4 the comparison between the GlottHMM and STRAIGHT systems can be evaluated through the objective scores. The above result in the clean row is obtained with the clean configuration of GlottHMM, while the one below is obtained using the noise reduction configuration. The contradictory results in the case of the GlottHMM-based system still happening: when SNR shows an improvement the MCD shows a quality decrease. Also, the GlottHMM-based system is found to suffer more degradation under severe noise conditions than the STRAIGHT one. We can spot that the noise reduction system makes the MCD values to increase, as so as the SNR values, giving the contradictory results previously seen.

In all the objective scores, GlottHMM presents a significantly steeper drop in quality as the noise level increases. This drop is clearly audible for SNR 5dB.

Noise	SNR	fwS	MCD
Babble	20	10.7	3.0
	10	7.6	2.7
	5	6.4	2.7
Factory	10	6.9	2.9
	5	5.5	3.2
Machine gun	0	9.4	2.7
Enhanced Babble	20	10.6	3.0
	10	8.4	2.8
	5	6.8	2.7
Enhanced Factory	10	8.7	3.2
	5	7.1	3.3

Table 3: Objective scores for the adapted test data using an external in the feature extraction F_0 calculated from the clean data with the GlottHMM-based system

9.2 Subjective Results

In the listening test two male voices were evaluated for different vocoders adaptation and different GlottHMM configurations by 32 native speakers using a web-based test. The reason to use only two of the three speakers is simply to have a test short enough (around 20 minutes long), because a long test might discourage the listeners. The results for both speakers were similar enough to be grouped.

In the AB test two different comparison where made. The first one asked the listeners about their preferences between different synthetic speech obtained with the GlottHMM-based system. The cases faced were:

- Adapted speech from clean data using the clean configuration against adapted speech from clean data using the noise reduction configuration

Noise	SNR	Adapted GlotHMM synthesized test data fwS	Adapted GlotHMM synthesized test data MCD	Adapted STRAIGHT synthesized test data fwS	Adapted STRAIGHT synthesized test data MCD
Clean	-	9.0 10.6	1.8 2.9	7.5	2.1
Babble	20	10.7	3	8.0	2.0
	10	7.6	2.7	7.5	2.1
	5	6.4	2.7	7.3	2.2
Enhanced Babble	20	10.6	3.0	8.0	2.0
	10	8.4	2.8	7.5	2.1
	5	6.8	2.7	7.3	2.2

Table 4: Objective scores comparing GlottHMM and STRAIGHT

- Adapted speech from clean data against adapted speech from babble 20dB data
- Adapted speech from babble 10dB data against the adaptation made from its enhanced version

All of these comparisons were made to find out which of the options the listener preferred, as in Table 4 the scores obtained were very similar. Figure 23 present the results of this firs part of the AB test.

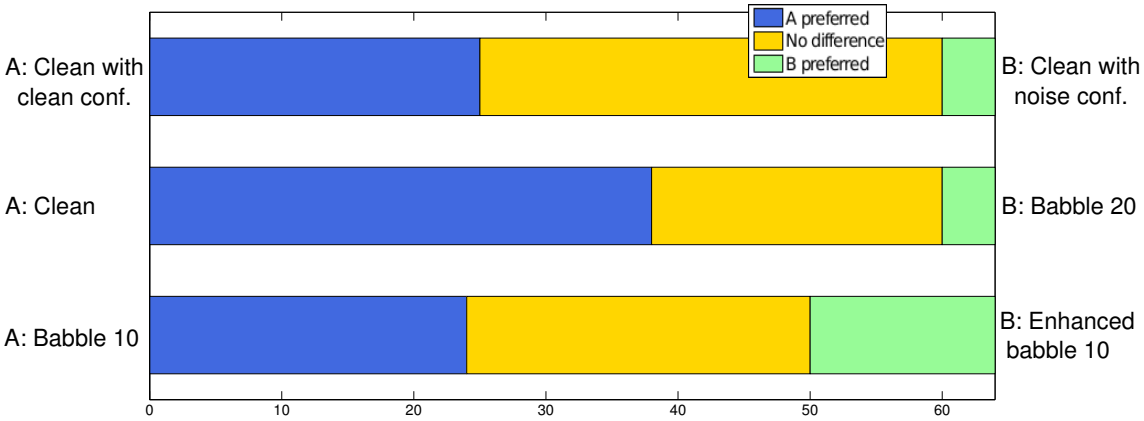

Figure 23: Results of the AB test comparing different adapted voices obtained with the GlottHMM-based system

In the first two cases, where the adapted voice using clean data and the clean configuration (both samples A) is compared to the one obtained using the noise

reduction configuration and babble 20dB data, the results are pretty clear and show a preference to speech using clean data and a clean configuration ($p = 0.043$ and $p \ll 0.0001$). For the third case, no significant conclusion can be formulated.

In Figure 24 the results of the AB test comparing GlottHMM-based system to the STRAIGHT-based are shown. The comparisons made in these tests are all for the cases where babble noise is found on the background, as is the most common noise you could find when recording speech and also one of the hardest ones to deal with, due to its similar nature with speech.

These results show a slight preference for the GlottHMM samples over the STRAIGHT ones in the case of SNR 10dB and 20dB babble noise in the adaptation data. The statistical significance test (binomial test) conducted shows that this preference is close to significant at best ($p = 0.0516$ for SNR 20dB and $p \gg 0.05$ for SNR 10dB). Nothing decisive can be said about the case where the adapted speech was obtained with the enhanced data.

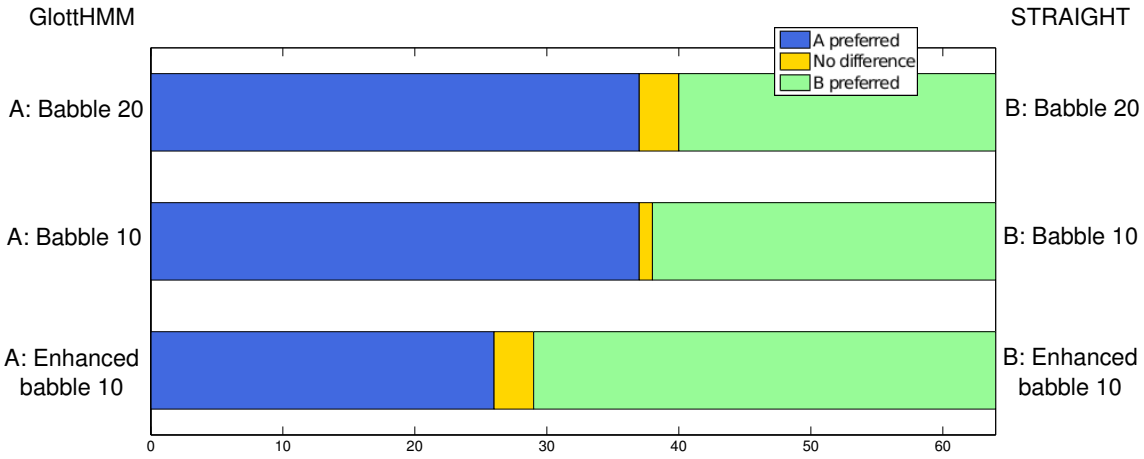

Figure 24: Results for the AB test comparing the performance of the GlottHMM-based system against the STRAIGHT-based one

Finally, Figure 25 presents the MOS scores for the listening test. The STRAIGHT system is rated slightly higher in naturalness than the GlottHMM-based system for almost all the noise cases. In similarity both systems are quite close, with STRAIGHT being slightly better in the case of babble at SNR 20dB. Background quality is rated very evenly for the STRAIGHT-based systems, whereas the GlottHMM is highly affected when the SNR drops from 20dB to 10dB.

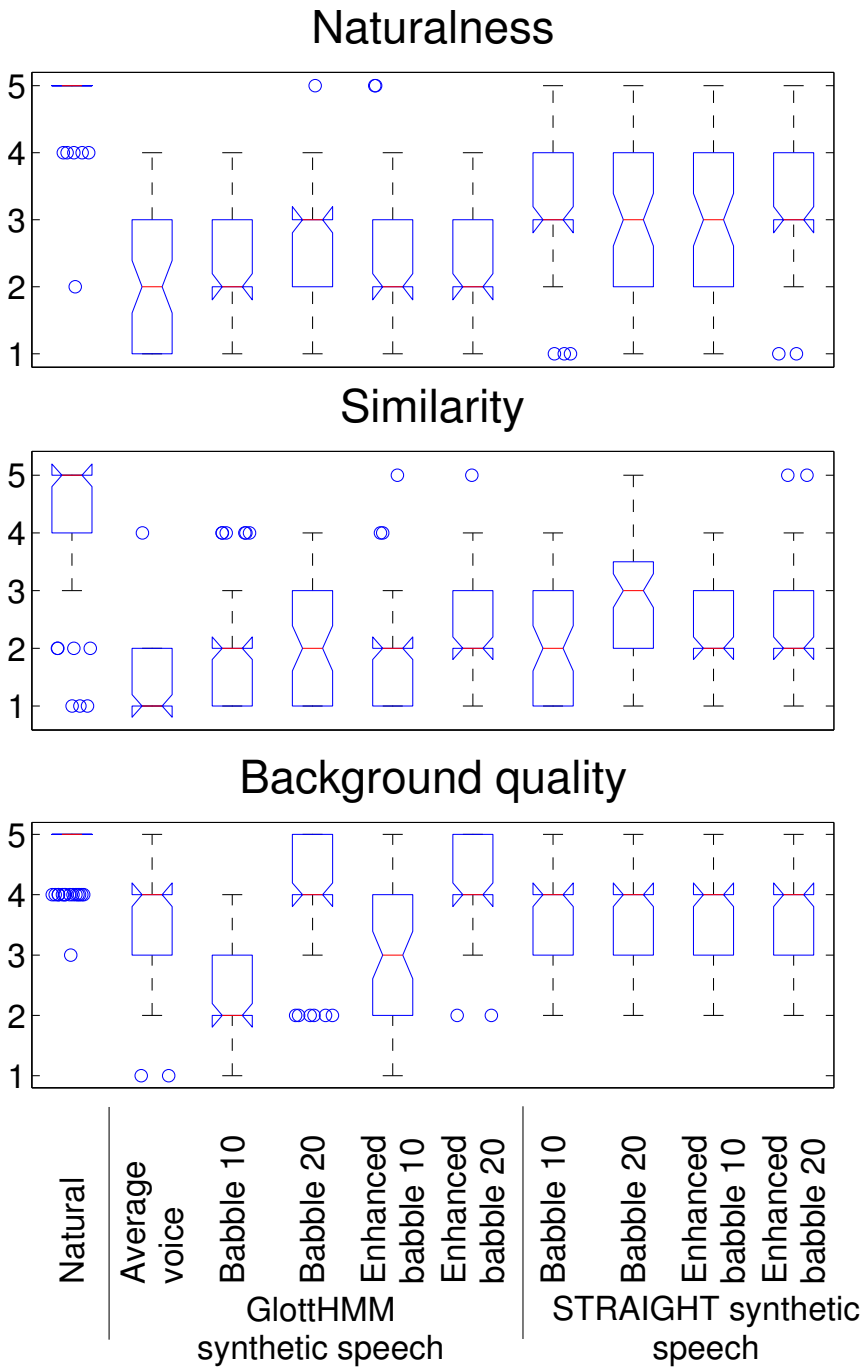

Figure 25: Mean opinion scores (MOS) for the second part of the listening test. Median is denoted by the red line, boxes cover 25th and 75th percent percentiles, whiskers cover the data not considered outliers. The notches mark the 95% confidence interval for the median

10 Discussion and Conclusion

10.1 Discussion

Both of the evaluated systems are capable of producing high quality synthetic speech. The STRAIGHT-based system is slightly better rated in some of the evaluated aspects when listening to single sentences in the MOS test. However, when comparing samples between two systems listeners displayed no preference for STRAIGHT-based over GlottHMM-based system. The listening test used in this work has previously been proved consistent in [9]. However, in [9] the tests were conducted on an identical framework for voice building, varying only the noise level, while in this project two different frameworks where used. When comparing different vocoders many factors could bias the test. For example, having different feature stream dimensions allow different clustering thresholds. Therefore, the listener might be very disciplined when rating samples based on the MOS test questions. In the listener's opinion smooth voices (STRAIGHT) sounds more natural and more similar to target speaker but a more varying voice (GlottHMM), even with some imperfections, might be preferred for an everyday use.

The results of the objective evaluation show the difficulties found to evaluate a vocoder in complex systems. The fwSNRseg and MCD scores for the two vocoders not only react very different to the increase of noise, but are contradicting to each other in quality assessment. Specially, in the case of GlottHMM it has been noticed that during the silences found within utterances each of the measures used leads to an opposite conclusion over the speech quality. To solve the problem spotted on the silences, the evaluation methods could take into account, for example, the differential energy of the speech signal, as in the silences, even in the presence of noise, a significant energy drop should be noticed. Besides, the STRAIGHT system have been trained with MCEP-formatted speech data while the GlottHMM emphasizes on formant modelling, which might be partial cause of the STRAIGHT system scoring far better in MCD and the GlottHMM doing the same with the perceptually motivated fwSNRseg. Thus, relaying on only one objective measure does not seem a good idea when technical choices must be done. Objective evaluation metrics should be developed looking to unify the evaluation of different technologies facilitating the comparisons.

10.2 Conclusion

A speaker-adaptive, based on the GlotHMM vocoder, speech system was built and compared to a STRAIGHT MCEP-based HMM-system built in [9]. Speaker adaptation data corrupted by different noises at varying SNR levels were used, but only babble noise was used in the comparison between systems. The system were first evaluated using objective methods that led to contradictory results that need further investigation. Formal listening tests showed that the STRAIGHT-based system was perceived as slightly higher than the GlottHMM-based one in terms of naturalness. Differences in similarity were very small. GlottHMM was found to be more suscep-

tible to degradation under more severe noise conditions in both the objective and subjective evaluations conducted. The preference tests did not show any significant differences between the systems.

GlottHMM has been shown to generate high-quality synthetic speech when the system is trained using noise-free data. The tests conducted in this project pointed out that GlottHMM is susceptible to severe noise present in the background, while STRAIGHT is more robust under the same severe noise conditions. If small amounts of noise are found in the background, GlottHMM works well.

References

[1] J. L. Flanagan, *Speech analysis; synthesis and perception, by James L. Flanagan.* Springer-Verlag Berlin, New York, 2nd ed. ed., 1972.

[2] D. H. Klatt, "Review of text-to-speech conversion for English," *Journal of Acoustic Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[3] T. Raitio, "Hidden Markov model based Finnish text-to-speech system utilizing glottal inverse filtering," Master's thesis, Helsinki University of Technology, 2008.

[4] M. Karjalainen, "Kommunikaatioakustiikka," 1999.

[5] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," in *Proceedings of the IEEE*, 2013.

[6] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 66–83, 2009.

[7] M. Airaksinen, "Analysis/synthesis comparison of vocoders utilized in statistical parametric speech synthesis," Master's thesis, Aalto University, 2012.

[8] J. M. Moreno, R. Karhila, T. Raitio, J. M. Montero, and M. Kurimo, "Effects of babble noise on a GlottHMM-based speaker-adaptive statistical speech synthesis system," in *Proceedings of INTERSPEECH 2014*. Under review.

[9] R. Karhila, U. Remes, and M. Kurimo, "Noise in HMM-based speech synthesis adaptation: Analysis, evaluation methods and experiments," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, no. 2, pp. 285–295, 2013.

[10] J. Yamagishi, Z. Ling, and S. King, "Robustness of HMM-based speech synthesis," in *Proc. Interspeech 2008*, 2008.

[11] K. Yanagisawa, J. Latorre, V. Wan, M. J. F. Gales, and S. King, "Noise robustness in HMM-TTS speaker adaptation," in *8th ISCA Workshop on Speech Synthesis*, (Barcelona, Spain), pp. 139–144, August 2013.

[12] E. M. Butler and E. M. Butler, *The myth of the magus.* Cambridge University Press, 1993.

[13] S. Lemmetty, "Review of speech synthesis technology," Master's thesis, Helsinki University of Technology, 1999.

[14] W. von Kempelen, *Mechanismus Der Menschlichen Sprache Nebst Beschreibung Seiner Sprechenden Maschine.* Stuttgart-Bad Cannstatt, 1970.

[15] M. R. Schroeder, "A brief history of synthetic speech," *Speech Commun.*, vol. 13, pp. 231–237, Oct. 1993.

[16] A. Takanishi, "Anthropomorphic talking robot waseda talker series." http://www.takanishi.mech.waseda.ac.jp/top/research/voice/index.htm. Accesed: 05-03-2014.

[17] K. Fukui, T. Kusano, Y. Mukaeda, Y. Suzuki, A. Takanishi, and M. Honda, "Speech robot mimicking human articulatory motion," in *Proceedings of INTERSPEECH 2010*, pp. 1021–1024, 2010.

[18] J. Flanagan and L. Rabiner, *Speech synthesis*. Dowden, Hutchinson & Ross, 1973.

[19] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer Speech & Language*, vol. 15, no. 3, pp. 287 – 333, 2001.

[20] J. Pickett, *The acoustics of speech communication: fundamentals, speech perception theory, and technology*. Allyn and Bacon, 1999.

[21] G. Fant, *Acoustic Theory of Speech Production*. Mouton De Gruyter, 1970.

[22] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, pp. 257–286, 1989.

[23] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theor.*, vol. 13, pp. 260–269, Sept. 2006.

[24] 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, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[25] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[26] J. Yamagishi, O. Watts, S. King, and B. Usabaev, "Roles of the average voice in speaker-adaptive HMM-based speech synthesis," in *Proceedings of INTERSPEECH 2010*, 2010.

[27] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE TRANSACTIONS on Information and Systems*, vol. 85, no. 3, pp. 455–464, 2002.

[28] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2, pp. 1137–1140, IEEE, 1996.

[29] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "A training method of average voice model for HMM-based speech synthesis," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 86, no. 8, pp. 1956–1963, 2003.

[30] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995.

[31] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[32] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains," *Speech and audio processing, ieee transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[33] K. Shinoda and C.-H. Lee, "A structural bayes approach to speaker adaptation," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 276–287, 2001.

[34] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 153–165, 2011.

[35] P. Alku, "Glottal wave analysis with pitch synchronous iterative adaptive inverse filtering," *Speech communication*, vol. 11, no. 2, pp. 109–118, 1992.

[36] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2, pp. 1303–1306, IEEE, 1997.

[37] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[38] Z. Heiga, T. Tomoki, M. Nakamura, and K. Tokuda, "Details of the nitech HMM-based speech synthesis system for the blizzard challenge 2005," *IEICE transactions on information and systems*, vol. 90, no. 1, pp. 325–333, 2007.

[39] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech communication*, vol. 9, no. 5, pp. 453–467, 1990.

[40] T. Raitio, A. Suni, H. Pulakka, M. Vainio, and P. Alku, "HMM-based Finnish text-to-speech system utilizing glottal inverse filtering.," in *Interspeech*, pp. 1881–1884, 2008.

[41] ETSI, *ES 202 050 V1.1.5 Speech processing, transmission and quality aspects (STQ), distributed speech recognition*, 2007.

[42] T. Toda and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis," in *Proceedings of INTERSPEECH 2005*, pp. 2801–2804, 2005.

[43] J. Yamagishi and O. Watts, "The CSTR/EMIME HTS system for Blizzard Challenge," in *Proc. Blizzard Challenge*, 2010.

[44] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: Ii. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247 – 251, 1993.

[45] M. Wester, "The EMIME Bilingual Database," Tech. Rep. EDI-INF-RR-1388, The University of Edinburgh, 2010.

[46] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. Interspeech*, 2010.

[47] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book*, vol. 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

[48] N. Kitawaki, H. Nagabuchi, and K. Itoh, "Objective quality evaluation for low-bit-rate speech coding systems," *Selected Areas in Communications, IEEE Journal on*, vol. 6, pp. 242 –248, feb 1988.

[49] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 229 –238, jan. 2008.

[50] R. Karhila, U. Remes, and M. Kurimo, "HMM-based speech synthesis adaptation using noisy data: Analysis and evaluation methods," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[51] ITU-T, *Recommendation P.835 (2003/11) Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm.*

[52] D. Howell, *Statistical methods for psychology.* Cengage Learning, 2012.

A GlottHMM Configuration

A.1 GlottHMM configuration file

Listing 1: Configuration file

```
##################################################################
#   Default configuration file for GlottHMM (v. 1.0.7)  #
##################################################################


# Analysis and Synthesis: Common parameters:
        SAMPLING_FREQUENCY =                16000;
        FRAME_LENGTH =                      25.0;
        UNVOICED_FRAME_LENGTH =             20.0;
        F0_FRAME_LENGTH =                   45.0;
        FRAME_SHIFT =                       5.0;
        LPC_ORDER =                         30;
        LPC_ORDER_SOURCE =                  10;
        WARPING_VT =                        0.0;
        WARPING_GL =                        0.0;
        HNR_CHANNELS =                      5;
        NUMBER_OF_HARMONICS =               10;
        SEPARATE_VU_SPECTRUM =              false;
        DIFFERENTIAL_LSF =                  false;
        LOG_F0 =                            false;
        DATA_FORMAT =                       "ASCII";        #
            Choose between "ASCII" / "BINARY"



# Noise reduction
        NOISE_REDUCTION_ANALYSIS =      false;
        NOISE_REDUCTION_SYNTHESIS =     false;
        NOISE_REDUCTION_LIMIT_DB =      4.5;
        NOISE_REDUCTION_DB =            35.0;



# Analysis:
        # Analysis: General parameters:
                PITCH_SYNCHRONOUS_ANALYSIS =    false;
                INVERT_SIGNAL =                 false;
                                # Remember to set true e.g. for
```

```
                MV voice (it's inverted)
39              HP_FILTERING =                  true;
40              HPFILTER_FILENAME =             "/home/
                    morenoj1/PFC/GlottHMM/hp_16khz";
41          # Analysis: Parameters for F0 estimation:
42              F0_MIN =                        65.0;
43              F0_MAX =                        200.0;
44              VOICING_THRESHOLD =             120.0;
45              ZCR_THRESHOLD =                 110.0;
46              USE_F0_POSTPROCESSING =         true;
47              RELATIVE_F0_THRESHOLD =         0.005;
48              F0_CHECK_RANGE =                10;
49              USE_EXTERNAL_F0 =               false;
50              EXTERNAL_F0_FILENAME =          "filename.F0
                    ";
51
52          # Analysis: Parameters for extracting pulse
                libraries:
53              MAX_NUMBER_OF_PULSES =          10000;
54              PULSEMAXLEN =                   45.0;
55              RESAMPLED_PULSELEN =            10.0;
56              WAVEFORM_SAMPLES =              10;
57              MAX_PULSE_LEN_DIFF =            0.05;
58              EXTRACT_ONLY_UNIQUE_PULSES =    true;
59              EXTRACT_ONE_PULSE_PER_FRAME =   true;
60
61                      # Analysis: Parameters for spectral
                            modeling:
62              USE_IAIF =                      true;
63              LPC_ORDER_GL_IAIF =             8;
                            # Order of the LPC analysis for
                    voice source in IAIF
64              USE_MOD_IAIF =                  false;
                            # Modified version of IAIF
65              LP_METHOD =                     "XLP";
                            # Select between "LPC" / "WLP" /
                    "XLP"
66              LP_STABILIZED =                 true;
67              LP_WEIGHTING =                  "STE";
                            # Select between "STE" / "GCI"
68              FORMANT_PRE_ENH_METHOD =        "NONE";
                            # Select between "NONE" / "LSF" /
                     "LPC"
69              FORMANT_PRE_ENH_COEFF =         0.5;
```

```
70                  FORMANT_PRE_ENH_LPC_DELTA =       20.0;
                                # Only for LPC-based method
71
72          # Analysis: Select parameters to be extracted:
73                  EXTRACT_F0 =                      true;
74                  EXTRACT_GAIN =                    true;
75                  EXTRACT_LSF =                     true;
76                  EXTRACT_LSFSOURCE =               true;
77                  EXTRACT_HNR =                     true;
78                  EXTRACT_HARMONICS =               false;
79                  EXTRACT_H1H2 =                    false;
80                  EXTRACT_NAQ =                     false;
81                  EXTRACT_WAVEFORM =                false;
82                  EXTRACT_INFOFILE =                true;
83                  EXTRACT_PULSELIB =                false;
84                  EXTRACT_SOURCE =                  false;
85
86
87
88
89  # Synthesis:
90          # Synthesis: General parameters:
91                  SYNTHESIZE_MULTIPLE_FILES =       false;
92                  SYNTHESIS_LIST =                  "
                        synthesis_list_filename";
93                  USE_HMM =                         true;
94
95          # Synthesis: Choose excitation technique and related
                parameters:
96                  USE_PULSE_LIBRARY =               false;
97                  GLOTTAL_PULSE_NAME =              "/home/
                        morenoj1/PFC/GlottHMM/pulse";
98                  PULSE_LIBRARY_NAME =              "/data/users
                        /morenoj1/data/perso_male/pulse_library/
                        pulse_libraries/perso_male_lib/
                        perso_male_lib";
99                  NORMALIZE_PULSELIB =              true;
100                 USE_PULSE_CLUSTERING =            false;
101                 USE_PULSE_INTERPOLATION =         true;
102                 AVERAGE_N_ADJACENT_PULSES =       0;
103                 ADD_NOISE_PULSELIB =              true;
104                 MAX_PULSES_IN_CLUSTER =           2000;
105                 NUMBER_OF_PULSE_CANDIDATES =      200;
106                 PULSE_ERROR_BIAS =                0.3;
107                 MELSPECTRUM_CH =                  100;
```

```
108                 CONCATENATION_COST =                1.0;
109                 TARGET_COST =                       1.0;
110                 PARAMETER_WEIGHTS =                 [0.0, 1.0,
                        1.0, 2.0, 3.0, 5.0, 1.0, 1.0, 1.0, 0.0];
111                 # Parameter names                  [LSF  SRC
                        HARM HNR  GAIN F0   WAV  H1H2 NAQ  PCA/
                        ICA]
112
113             # Synthesis: Select used parameters:
114                 # F0, Gain, and LSFs are always used
115                 USE_LSFSOURCE =                     true;
116                 USE_HNR =                           true;
117                 USE_HARMONICS =                     false;
118                 USE_H1H2 =                          false;
119                 USE_NAQ =                           false;
120                 USE_WAVEFORM =                      false;
121                 USE_MELSPECTRUM =                   false;
122                 USE_PULSE_PCA =                     false;
123
124             # Synthesis: Set level and band of voiced noise:
125                 NOISE_GAIN_VOICED =                 0.75;
126                 NOISE_LOW_FREQ_LIMIT =              2400.0;
                            # Hz
127
128             # Synthesis: Smoothing of parameters for analysis-
                    synthesis:
129                 LSF_SMOOTH_LEN =                    5;
130                 LSFSOURCE_SMOOTH_LEN =              3;
131                 GAIN_SMOOTH_LEN =                   5;
132                 HNR_SMOOTH_LEN =                    15;
133                 HARMONICS_SMOOTH_LEN =              5;
134
135             # Synthesis: Gain related parameters:
136                 GAIN_UNVOICED =                     1.0;
137                 NORM_GAIN_SMOOTH_V_LEN =            0;
138                 NORM_GAIN_SMOOTH_UV_LEN =           0;
139                 GAIN_VOICED_FRAME_LENGTH =          25.0;
140                 GAIN_UNVOICED_FRAME_LENGTH =        20.0;
141
142             # Synthesis: Postfiltering:
143                 POSTFILTER_METHOD =                 "LPC";  #
                        Select between "NONE" / "LSF" / "LPC"
144                 POSTFILTER_COEFFICIENT =            0.4;
145
146             # Synthesis: Utils:
```

```
147                 USE_HARMONIC_MODIFICATION =     false;
148                 HP_FILTER_F0 =                  false;
149                 FILTER_UPDATE_INTERVAL_VT =     0.3;
150                 FILTER_UPDATE_INTERVAL_GL =     0.05;
151                 WRITE_FFT_SPECTRA =             false;
152                 WRITE_EXCITATION_TO_WAV =       false;
153
154         # Synthesis: Voice adaptation:
155                 PITCH =                         1.0;
156                 SPEED =                         1.0;
157                 JITTER =                        0.0;
158                 ADAPT_TO_PULSELIB =             false;
159                 ADAPT_COEFF =                   1.0;
160                 USE_PULSELIB_LSF =              false;
161                 NOISE_ROBUST_SPEECH =           false;
162
163         # Synthesis: Pulse library PCA/ICA:
164                 USE_PULSELIB_PCA =              false;
165                 PCA_ORDER =                     12;
166                 PCA_ORDER_SYNTHESIS =           0;
167                 PCA_SPECTRAL_MATCHING =         false;
168                 PCA_PULSE_LENGTH =              800;
```

A.2 Noise Reduction Parameters

Listing 2: Noise reduction parameters in GlottHMM's configuration file

```
1  # Noise reduction
2          NOISE_REDUCTION_ANALYSIS =      false;
3          NOISE_REDUCTION_SYNTHESIS =     false;
4          NOISE_REDUCTION_LIMIT_DB =      4.5;
5          NOISE_REDUCTION_DB =            35.0;
```

B Questions of the Listening Test

▷ Natural reference speech sample
▷ Synthesized speech sample A
▷ Synthesized speech sample B

Play the reference sentence. Then play both sample sentences. Considering the OVERALL QUALITY of the signal, select the one you would prefer to represent the reference voice in applications like mobile devices, video games, audio books etc. Regarding the OVERALL QUALITY

A. First sample is better

B. Second sample is better

C. They sound exactly the same

Table B1: Questions used in the subjective evaluation AB test

▷ Synthesized speech sample
Play the sample and attending ONLY to the SPEECH SIGNAL, select the category which best describes the sample you just heard. The SPEECH SIGNAL in this signal was

5. Completely natural
4. Quite natural
3. Somewhat unnatural but acceptable
2. Quite unnatural
1. Completely unnatural

▷ Synthesized speech sample
Play the sample and attending ONLY to the BACKGROUND, select the category which best describes the sample you just heard. The BACKGROUND in this signal was

5. Clean
4. Quite clean
3. Somewhat noisy but not intrusive
2. Quite noisy and somewhat intrusive
1. Very noisy and very intrusive

▷ Natural reference speech sample
▷ Synthesized speech sample
Play both samples, and attending ONLY to the SPEECH SIGNAL, select the category which best describes the second sample to the reference sample.The voices in the SPEECH SIGNALS of the samples sounded

5. Exactly like the same person
4. Quite like the same person
3. Somewhat different but recognizable
2. Quite like a different person
1. Like a totally different person

Table B2: Questions used in the subjective evaluation MOS test