Arturo Romero Blanco

Spanish Emotional Speech Synthesis

School of Electrical Engineering

Espoo 02.04.2014

Project supervisor:

Prof. Paavo Alku

Project advisor:

M.Sc. (Tech.) Tuomo Raitio

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE FINAL PROJECT

Author: Arturo Romero Blanco

Title: Spanish Emotional Speech Synthesis

Date: 02.04.2014 Language: English Number of pages:8+51

Department of Signal Processing and Acoustics

Professorship: Acoustic and audio signal processing Code: S-89

Supervisor: Prof. Paavo Alku

Advisor: M.Sc. (Tech.) Tuomo Raitio

In this project a text-to-speech (TTS) HMM-based speech system (HTS) has been used to create emotional synthetic speech in Spanish. Nowadays the synthetic voices have high quality, but this is not enough, they must be able to capture the natural expressiveness of the human speech. Giving this expressiveness to the synthetic voices will lead to a much more natural voice, that is the goal of these systems.

To achieve this, both male and female voices will be used and two different techniques will be applied: dependent models and average voice models with adaptation.

In this TTS system different vocoders can be used. For this project GlottHMM has been used and then three perceptual test have been carried out to compare it with STRAIGHT vocoder.

The results of the perceptual tests shows that STRAIGHT is very robust and that GlottHMM is not yet at its level regarding the emotional speech synthesis.

Keywords: emotional speech synthesis, synthetic speech, vocoder, HMM-based, GlottHMM, STRAIGHT

Acknowledgements

This project has been done as part of the Simple4All project in the Deparment of Signal Processing and Acoustics (SPA) at Aalto University School of Electrical Engineering in colaboration with the Speech Technology Group (GTH) of the ETSI Telecomunicación UPM.

First of all thanks to the GTH group (Juancho, Ruben, Roberto y Jaime), the SPA group (Paavo, Mikko, Tuomo, Reima, Lauri) and the Simple4All project for the great opportunity and for the help.

Also thanks to my family for the support and love they have been given to me and to Jose for be patient and stand all my nonsense.

And last but not least thanks to Julia for her love, support and for cheer me up in the worst moments.

Otaniemi, 02.04.2014

Arturo Romero Blanco

Contents

Appendices

List of Figures

List of Tables

Abbreviations

CMLLR	Constrained Maximum Likelihood Linear Regression
CSMAPLR	Constrained Structural Maximum A Posteriori Linear Regression
ES	Emotion Strength
F0	Fundamental Frequency
FFT	Fast Fourier Transform
GV	Global Variance
HNR	Harmonic to Noise Ratio
HMM	Hidden Markov Models
HTK	Hidden Markov Model Toolkit
HTS	HMM-based Speech Synthesis System
IAIF	Iterative Adaptive Inverse Filtering
LPC	Linear Predictive Coding
MAP	Maximum a Posteriori
MFCCs	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MLSA	Mel Log Spectrum Approximation
MOS	Mean Opinion Score
MSD-HMM	Multi-Space probability Distribution HMM
PSOLA	Pitch-Synchronous Overlap-Add
SAT	Speaker Adaptive Training
SEV	Spanish Emotional Voices
SMAP	Structural Maximum a Posteriori
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum
SQ	Speech Quality
TTS	Text to Speech

1 Introduction

Speech synthesis is the artificial production of human speech. One of the biggest challenges in this field is the production of naturally sounding synthetic voices. This means that is not enough that the synthetic voices have high quality, they must also be able to capture the natural expressiveness that the human speech has.

With the introduction of the human-machine interfaces, the interest in this field have grown up, because the speech synthesis play an important role on them. Different applications such as as telecommunication services, language education, help to people with disabilities, etc can be easily found. Thanks to this, studies on how to improve its quality, naturalness, expressiveness, etc. have been done.

Expressive speech synthesis is a sub-field of speech synthesis that has been drawing a lot of attention lately. So assign expressiveness (e.g. emotions or speaking styles [1]) to the synthetic voices will lead, if succeeded, to a much more natural voice, increasing the overall satisfaction of the end users of the applications.

One of the emotional speech synthesis main problems a few years ago was to find a data base with enough data to train a robust model because emotional speech is not easy to find, so it has to be recorded on purpose and in good conditions So techniques like transplanting the emotions or the styles [1] to another speakers have been tried in order to give emotions to speakers that have not an expressive database.

Of the two main speech synthesis techniques (unit selection [2] and HMM based) HMM based synthesis has been used in this project due to its parametric nature is much more adaptable and adaptations techniques can be applied on them, so a big amount of data is not required.

This project is focused on the production of emotional speech synthesis in Spanish, and it is focused on four emotions (anger, happiness, sadness and surprise) plus the neutral speech. This will be done using a text-to-speech (TTS) system, where the input is text with a special format (label) and the system generates the speech waveform. The TTS system is composed by a vocoder (analysis/synthesis tool like STRAIGHT or GlottHMM) and a training module.

This is not the first attempt to do such a thing, emotional speech synthesis has been tried before and with the STRAIGHT vocoder. So the goal in this project is the use of the GlottHMM vocoder developed in Helsinki, that has been proved to be good in expressive speech recognition [3] and in resynthesis [4] and compare it with STRAIGHT regarding the naturalness in emotional speech synthesis using two different techniques: dependent models and adaptation. In the first one each emotion will be treated separately to produce synthetic voice meanwhile in the second one all the data will be treated together to build and average model and then an adaptation with the desired output emotion will be carried out.

The project is organized as follows:
The history of the speech synthesis is presented in Section 2. To understand the complexity of a definition for emotion, emotional theory is explained in Section 3. Information about the theory used in this project is presented in Sections 4 to7. In Section 8 can be found the experiments that have been done and the steps followed

to accomplish them. In Section 9 the results of the test performed with the synthesis samples achieved in the experiment can be found and in Section 10 the discussion and conclusion of this project are exposed.

2 Speech Synthesis History

The earliest successful attempts to produce speech synthesis were made over two hundred years ago [5]. For example, in 1779 by Professor Kratzensteint build some apparatus that represented the human vocal tract to produce five long vowels due to the physiological differences between the vowels. The apparatus were acoustic resonators similar to the human vocal tract and he activated them with reeds like the one used in musical instruments.

The first recorded success in connected speech synthesis was achieved by Wolfgang von Kempelen in 1791 when he completed the construction of his "Acoustic-Mechanical Speech Machine" which was a ingenious pneumatic synthesizer (see Figure 1). The machine had a pressure chamber for the lungs, a vibrating reed to act as vocal cords and a leather bag for the vocal tract action. Changing the shape (by hand) of the leather bag different vowel sounds were produced. Constants were simulated by four separate constricted passages that were controlled by the fingers. There were also a couple of hiss whistles to allow the simulation of fricatives and a pair of openings to simulate the nostrils. For plosive sounds a model of a vocal tract that included a hinged tongue and movable lips was employed. To produce a sequence of sounds that seems like speech a lot of practice was needed.

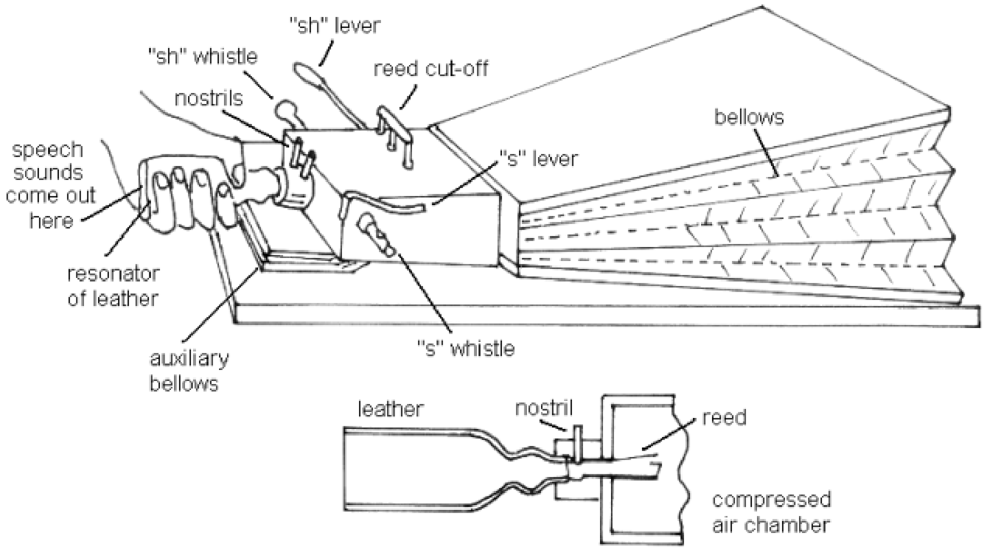

Figure 1: Kempelen Acoustic-Mechanical Speech Machine [6]

The connection between a specific vowel sound and the geometry of the vocal tract was found in 1838 by Willis, who synthesized different vowels with tube resonators and discovered that the quality of the vowel depended only on the length of the tube and not on its diameter. Also in the late 1800's Alexander Graham Bell constructed with his father same kind of speaking machine as the Wheastone's speaking machine that was a reproduction of the Kempelen speaking machine with a few changes.

With the 20th century came the development of electronics and later of electronic resonators. There were a few attempts early in the century to use electronic resonators in such a way that they could produce steady state vowels. An example of this is the electrical synthesis device created by Stewart in 1922. The synthesizer had a buzzer as excitation and two resonant circuits to model the acoustic resonances of the vocal tract. The machine was able to generate single static vowels sounds with two lowest formants, but not any consonants or connected utterances. Obata and Teshima discovered the third formant in vowels. It is considered that the three firs formants are enough for intelligible synthetic speech. It was finally in the late 1930's when the work of Homer Dudley at the Bell Laboratories produced the first electrical connected speech synthesizer.

Dudley developed two devices. One of them, the 'Voder' (Figure 2) was basically a parallel array of ten electronic resonators arranged as contiguous band-pass filters spanning the important frequencies of the speech spectrum. It consisted of a 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 whose output gain were controlled via keyboard. A considerable skill was needed to play a sentence on the device and the quality was not good, so in the end it was consider of little practical value, but after the demonstration of the Voder the scientific world became more and more interested in speech synthesis.

The other device Dudley made was called 'channel vocoder'. This channel vocoder and all subsequent vocoders are basically analysis/synthesis devices. They are divided into two halves, an analysis half and a synthesis half. The fist one analyses an incoming speech signal and obtains certain parameters from that natural signal. These parameters are passed as codes to the second half (synthesis) and there they are used to resynthesize a synthetic version of the incoming speech. The channel vocoder is the simplest of the vocoders. It is divided in two branches, one of them determines if the signal is voice or unvoiced and if voiced it determines the pitch. This information is used to produce a synthetic source. The other branch is a bank of electronic resonators acting like band-pass filters which measure the level of the signal in each frequency band at each point in time. With this information the synthetic source is produced (in the synthesis half of the vocoder) and is mixed with a spectral envelope reconstituted from the filter level values to produce synthetic version of the original signal.

The vocoders were originally developed at the Bell Telephone Labs as devices which allowed a signal to be coded more efficiently and thus allowed more conversations at the same time in the telephone network. More other vocoder configurations have been developed with simply filter banks and rely on complex mathematical transforms of the data (e.g Linear Prediction Coefficient vocoders) or on the detection of the formants in the speech signal.
In 1951 the pattern play-back machine (Figure 3) was developed by Cooper, Liberman and Borst. It reconverted recorded spectrogram patters into sounds, either original or modified form.

In 1953 Walter Lawrence introduced the first formant synthesizer, PAT, which

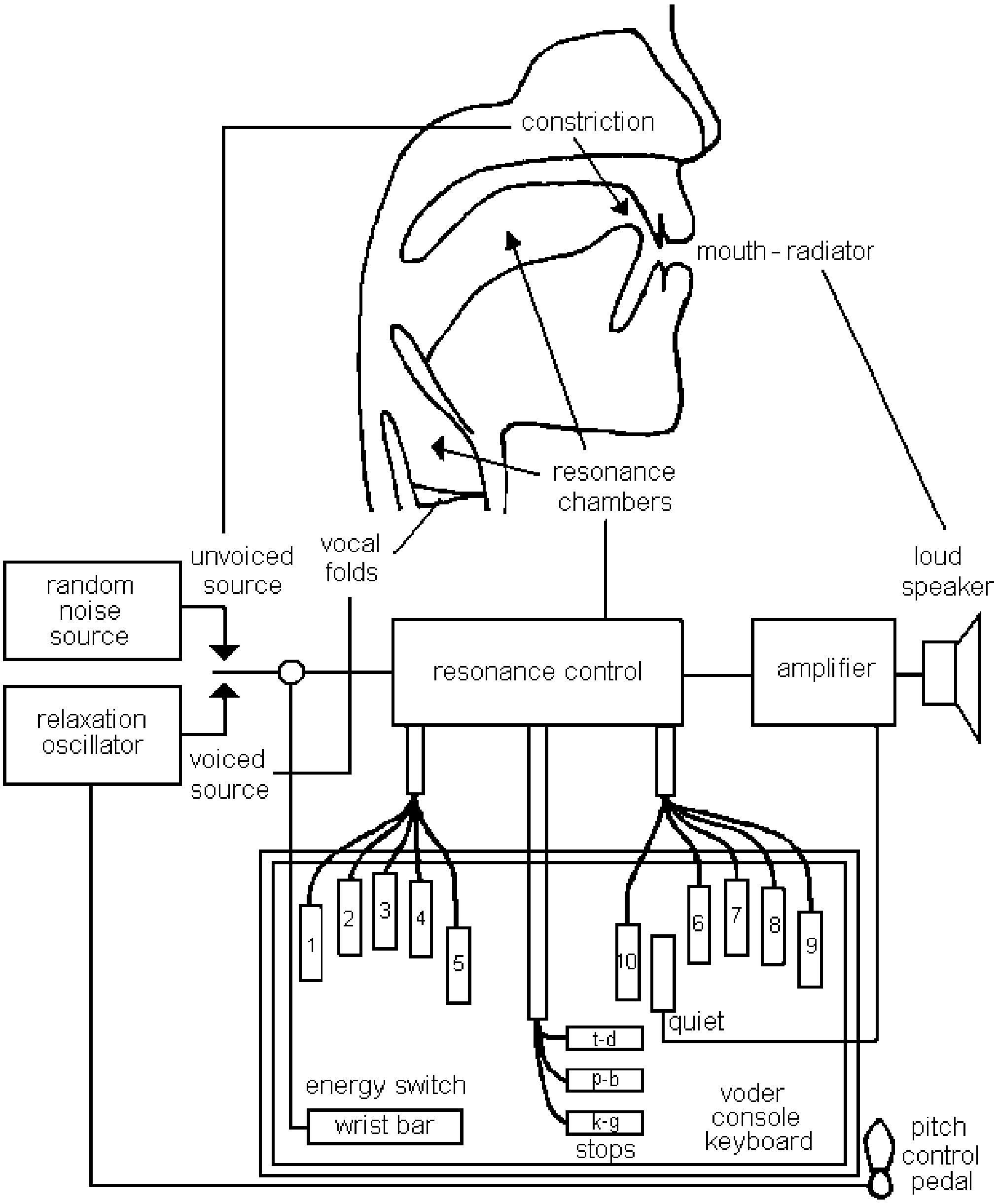

Figure 2: Dudley's Voder speech synthesizer [6]

looked similar to the pattern playback. It consisted of three electronic formant resonators connected in parallel and the input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency and noise amplitude. At that time Gunnar Fant introduced the first cascade formant synthesizer OVE I which consisted of formant resonators connected in cascade. Ten years later he introduced an improve (OVE II) with Martony, which consisted on

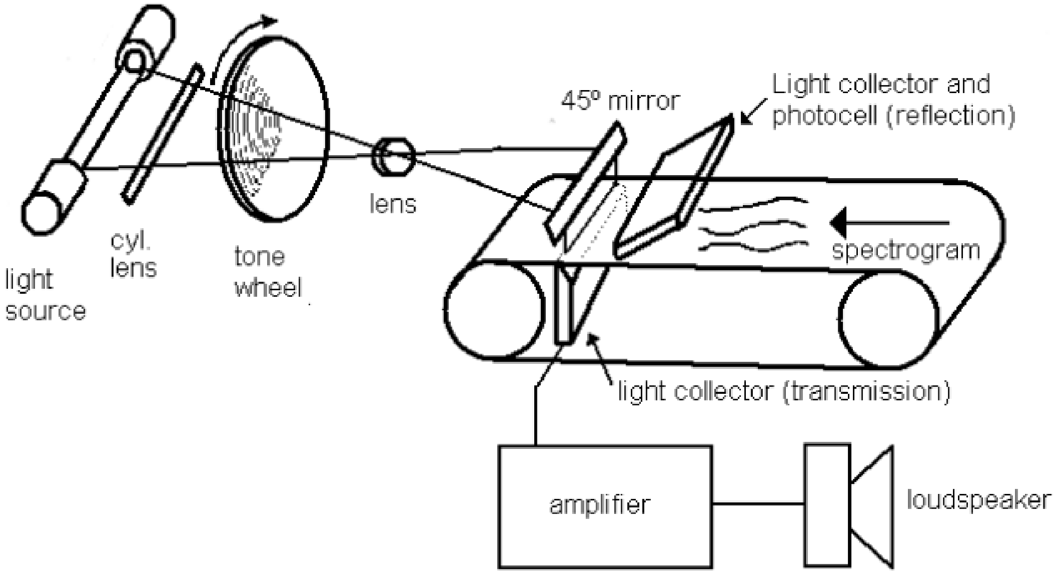

Figure 3: Pattern play-back machine [6]

separated parts to model the transfer function of the vocal tract for vowels, nasal, and obstruent consonants.
In 1958 the first articulatory synthesizer (The DAVO) was introduced at the Massachusetts Institute of Technology by George Rosen. In mid 1960's the first experiments with Linear Predictive Coding (LPC) were made, but it was first used in low-cost systems and its quality was poor. With some modifications this method has been found very useful.

In 1979 Allen, Hunnicutt and Klatt demonstrated the MITalk laboratory text to speech system. Two years later Klatt introduced his Klattalk system, which used a new sophisticated voicing source.

The first reading aid for blind people with an optical scanner was introduced in 1976 by Kurzweil. This system was capable to read quite well multiform written text.

In the late 1970's a lot of commercial TTS and speech synthesis products were introduced. The first integrated circuit was probably the Votrax chip which consisted of cascade formant synthesizer and simple low-pass smoothing circuits. In 1980 The LPC based Speak-n-Spell synthesizer based on low cost linear prediction synthesis chip was introduced by Texas Instruments and it was used for an electronic reading aid for children.

Modern speech synthesis technologies involve quite complicated and sophisticated methods and algorithms. One of the methods applied "recently" in speech synthesis is Hidden Markov Models (HMM, Section 4).

3 Emotions

One of the biggest problems found in research about speech is its variability. The intelligibility of the speech synthesizers is similar to the human one, but they do not have the variability of human speech which makes synthetic voice sound no natural.

The emotion is not a simple phenomenon, a lot of factors contribute to this. Emotions are experienced when something unexpected happens and the emotional effects start to have control in those moments. So emotion can be also described as the interface of the organism with the outside world, pointing three main emotion functions:

- Reflect the evaluation of the importance of a particular excitation in terms of the organism necessities, preferences, etc.
- Prepare physiologic and physically the organism for the appropriate action.
- Notify the state of the organism and its intentions to other organisms that surround it.

Emotion and mood are two different concepts, while emotions happen suddenly in response of a determined excitation and last seconds or minutes, the mood is more ambiguous in its nature and can last hours or days.

A lot of the words used to define emotions and its effects are necessary diffuse and are not clearly defined. This can be explained due to the difficulty for expressing with words abstract concepts that can not be quantified. For that reason, to describe the characteristic of the emotions a group of emotive words are used, but most of them are selected for personal choice.

The first researches about how the emotions affect to the behavior and the language of the animals were briefly described by Darwin in his book *The Expression of Emotion in Man and Animals* [7]. Lately, the effects of the emotions in speech have been studied by acoustic researchers that have analyzed the speech signal, by linguist, that have studied the lexical and prosody effects, and by psychologist. Thanks to them a lot of components present in emotions have been identified. The more important are: pitch, duration and voice quality.

The pitch (F0) is the fundamental frequency at which the vocal cords vibrates. The characteristic of the pitch are some of the main source of information about emotions. For example:

- The average value of F0 express the level of excitation of the speaker, so a high average of F0 means a higher level of excitement.
- The range of F0 is the distance between the maximum and minimum value of the F0. It also reflects the level of excitation of the speaker.
- Fluctuations in F0, defined as the speed of the fluctuation between high and low values and if they are blunt or soft.

The duration is the component of prosody described by the speed of the speech and the situation of the accents, and which effects are the rhythm and the speed. Emotions can be distinguish for some features as:

- Speech speed: usually an excited speaker will reduce the duration of syllables.
- Number of pauses and its duration: an excited speaker will tend to speak faster, with less and shorter pauses, while a depressed speaker will speak slower and with bigger pauses.
- Quotient between speak and pauses time.

The quality of the speech can be distinguish by:

- Intensity: is related with the perception of the volume.
- Voice irregularities: the speech jitter reflects the fluctuations of F0 of a glottal pulse to the other (like in angry emotion) or the disappearance of speech in some emotions (like sadness).
- The quotient between high and low frequencies: a big amount of energy in high frequencies is associated with the angry emotion, while low amount of energy is related with sadness.
- Breathiness and larynx effects reflects the characteristics of the vocal tract that are related with the customization of each voice.

Different classifications have been given to the emotion: The emotions can be divided into primary and secondary emotions [8].

- Primary emotions are those that are considered no acquired through experience but through evolutionary processes. In this group the happiness, sadness, fear, anger, surprise and disgust are found.
- Secondary emotions, are those that derive from previous ones through experience and cognitive modulation.

Joel Davitz and klaus Scherer classified the emotions and its effects using three edges of the semantic field:

- Power or Strength: corresponds to the attention or rejection, differentiating between emotions started by a subject to the ones that appear of the environment.
- Pleasure or evaluation: according to the pleasant or unpleasant of the emotion.
- Activity: presence or absence of energy or tension

Thank to some research it has been discovered that emotions with a same lever of activity are easier to confuse that the ones that have a similar level of strength or pleasure. So the activity is more related with simple hearing variables as tone or intensity.

Some researchers have divided the emotions into two groups, so an emotion can be:

- Active: which qualities are a low speech speed, low volume, low tone and a more resonant timbre.
- Passive: which qualities are a high speech speed, high volume, high tone and a "turned on" timbre.

Another classification more simple and natural is divide the emotion in positive or negative. Different levels inside this classification can be found.

More information about emotions like biological reasons can be found in [9].

4 HMM

The Hidden Markov Model (HMM) is one the statistical time series model most used in different fields. It has been used in speech recognition for years with great success and also TTS systems has made substantial progress in the last years using HMM.

A HMM is a finite state machine which generates a sequence of discrete time observations. At each time unit, the HMM change the state at Markov process with a state transition probability and the generates observational data in accordance with an output probability distribution of the current state.

A N-state HMM machine is defined by the state transition probability (A), the output probability distribution (B) and initial state probability (Π). Typical HMM structures can be seen in Figure 4.

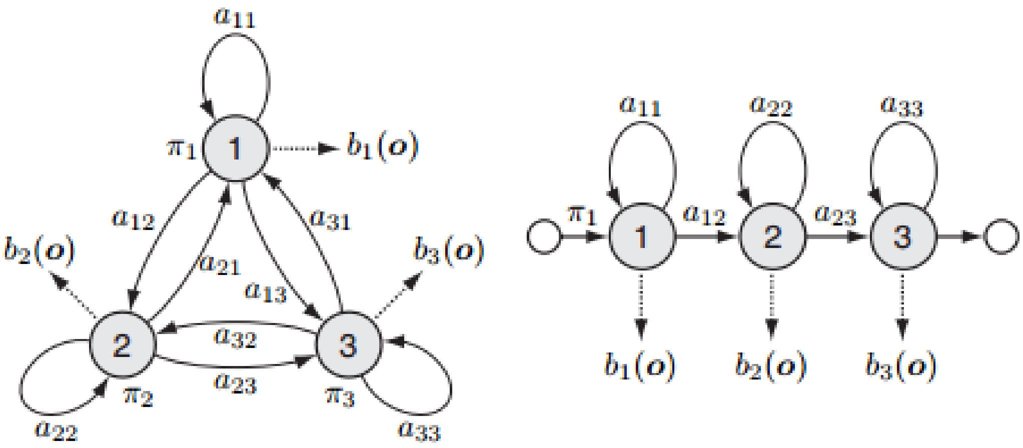

Figure 4: Typical HMM structures [10]

The structure on the left of Figure 4 is a 3-state ergodic model, in which all states can be reached by the others in a single transition. The structure on the right is a 3-state left to right model, in which the state index simply increases or stays depending on the time increment. This last model is often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successfully change.

5 HMM-Based Speech Synthesis

Here an HMM-based text-to-speech system is described. In the HMM-based speech synthesis, the speech parameters of a speech unit are statistically modeled and generated using HMMs based on maximum likelihood criterion [10].

The main goal of the TTS system is to produce natural synthetic speech sound including different types of speaking styles and emotions. In order to achieve this the system can be divided into two main parts: training and synthesis, as it is illustrated in Figure 5. The analysis is considered as part of the training and is where the features are extracted from the speech database. These features are then modeled by HMM. In the synthesis part, the HMMs are concatenated according to the analyzed input text (label) and speech parameters are generated from the HMM, then the synthesis module transforms them into a speech waveform.

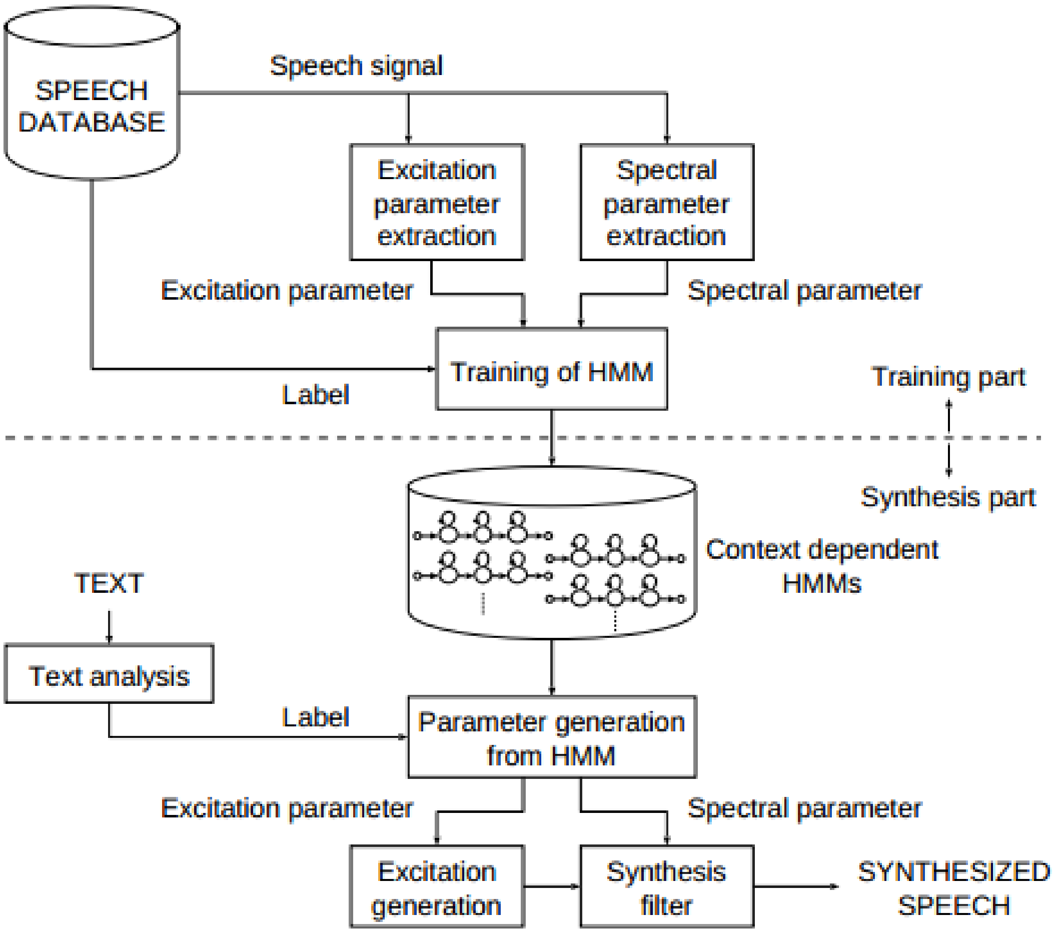

Figure 5: TTS overview [11]

5.1 Training Part

As it has been seen in Section 5, this training part is divided into two stages: the parametrization or feature extraction and the HMM training.

In the parametrization stage the input speech signal is compressed into a few parameters. These parameters have to describe the characteristics of the signal as accurately as possible. This stage is done in a different ways depending on the vocoder that is being used and will be explained in Section 6. For more detail see [12] [4].

In the HMM training stage the features obtained are modeled simultaneously by HMM. First of all monophone HMM models are trained in a 7-state left-to-right structure with 5 emitting states (similar to Figure 4). All the parameters except the F0 are modeled with continuous density HMMs by single Gaussian distributions with diagonal covariance matrix. F0 is modeled with by a multi-space probability distribution (MSD-HMM) [10] due to the conventional continuous or discrete HMMs models can not be applied to F0 pattern modeling because F0 consist of one-dimension continuous values and a discrete symbol that represents the unvoiced. The state duration for each HMM are modeled with multidimensional Gaussian distributions [13]. For GlottHMM each feature is modeled in an individual stream and for the F0 due to the MSD-HMM three streams are used, so the model has eight streams. In order to smooth transitions between states in parameter generation the delta and delta-delta coefficients of each feature are calculated, so the total feature order is 171.

After the training of the monophone HMMs, the monophone models are converted into context dependent models. As the number of contextual factor increase, their combination increase exponentially. This is a problem because with limited training data the model parameters can not be accurately estimated and it is impossible to cover all the combinations of contextual factors even with a prepared speech database. To solve this, the models for each feature are clustered independently by using a decision-tree based context clustering (Figure 6). In order to generate synthesis parameters for new observations vectors that are not included in the training data the clustering is also required.

5.2 Synthesis Part

In the synthesis part, the model created in the training part is used to generate speech parameters according to a text input (label). With these parameters the synthesis module is able to generate a speech waveform. So the synthesis part has two stages: the parameter generation and the synthesis as is illustrated in Figure 7.

In the parameter generation stage, the text input is first converted into to a context based label sequence by performing phonological and high level linguistic. According to the decision trees generated in the training stage and the label sequence, a sentence HMM is generated by concatenating the context dependent HMMs. The state durations of the sentence HMM are determined so that they maximize the likelihood of the state duration densities. With the sentence and the

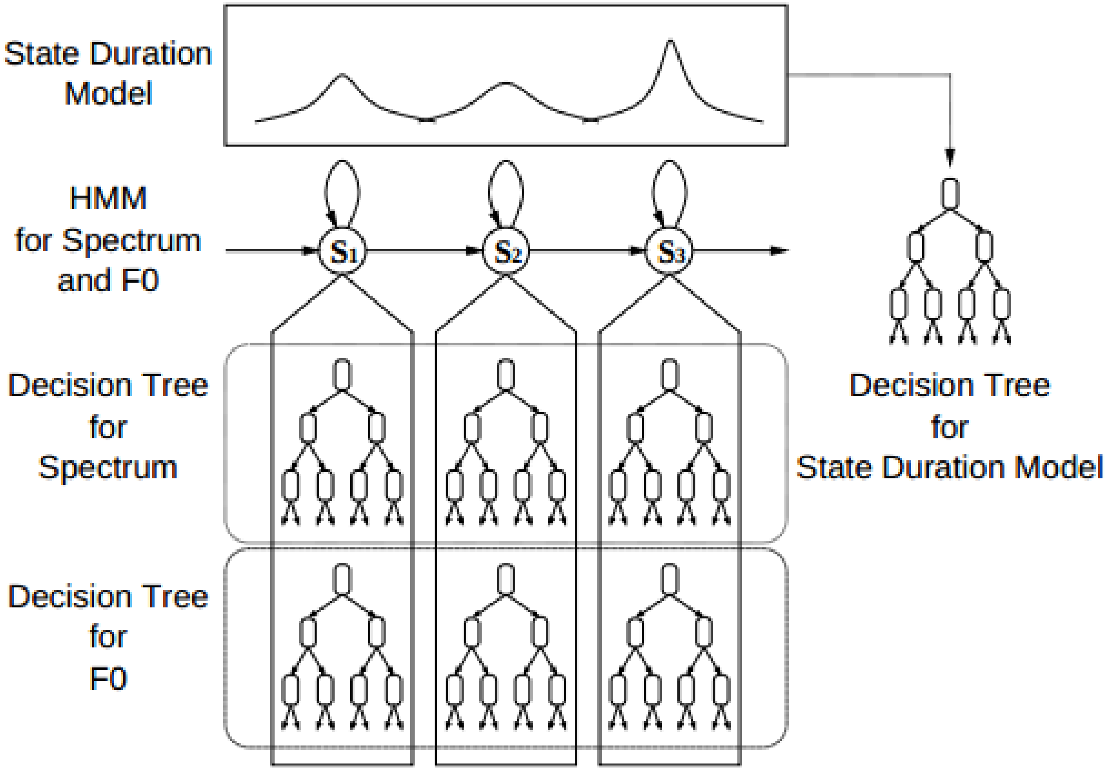

Figure 6: Example of decision-tree based context clustering for some features [11]

state durations, a sequence of speech features are generated and then used by the synthesis module to generate the speech waveform.

In the synthesis stage, as it has already been said, the speech waveform is generated according to the features generated in the first stage of the synthesis part.

The synthesis part also differs depending of the vocoder used, so it will be explained in Section 6.

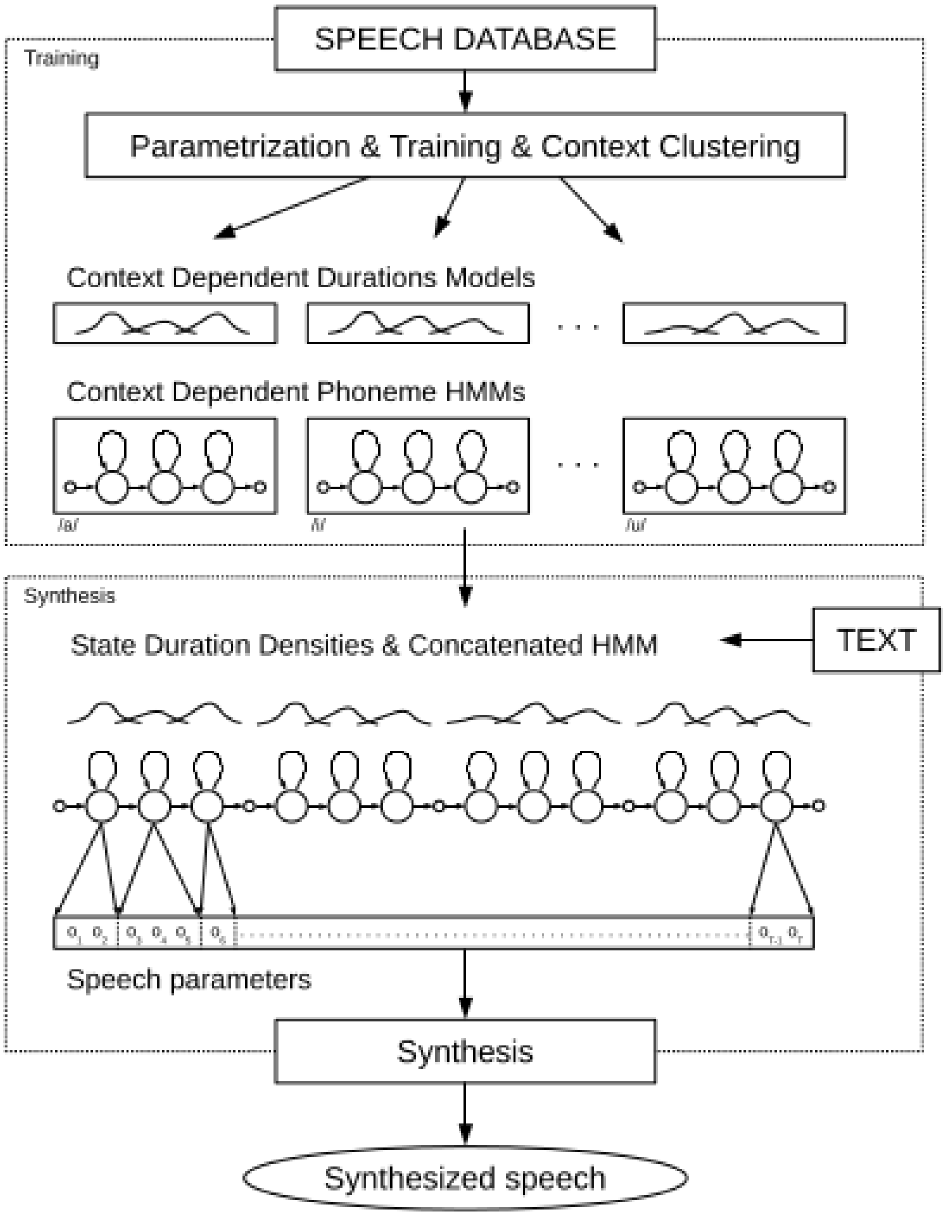
SPEECH DATABASE
Training
Parametrization & Training & Context Clustering
Context Dependent Durations Models
Context Dependent Phoneme HMMs
/a/
/i/
/u/
Synthesis
State Duration Densities & Concatenated HMM
TEXT
Speech parameters
Synthesis
Synthesized speech

Figure 7: HMM-based generation process of speech parameters [12]

6 Vocoders

Many different vocoders have been developed to be applied with HMM-based speech synthesis [4]. In this section two of them will be explained: GlottHMM and STRAIGHT due to that they are the ones that are being compared in this project.

6.1 GlottHMM

The GlottHMM was proposed by Tuomo Raitio in [12] and [14]. GlottHMM estimates the real glottal pulse signal G(z) an the vocal tract filter V(z) associated with it. So the speech signal can be represented as:

$$S(z) = G(z)V(z)L(z) \tag{1}$$

where L(z) represents the lip radiation. All parts are estimated of real physical properties. For example the glottal pulse signal can be divided into the source part E(z) an the filter containing the spectral envelope of the glottal pulse F$_G$(z):

$$G(z) = F_G(z)E(z) \tag{2}$$

and so the vocal tract filter can be expressed as:

$$V(z) = \frac{F(z)}{F_G(z)L(z)} \tag{3}$$

6.1.1 Analysis

To extract the parameters (analysis) of the speech signal GlottHMM follows this steps:

- First, the speech signal is high-pass filtered and windowed into fixed length rectangular frames, from which the signal log energy is calculated as a feature parameter.
- Second, the Iterative Adaptive Inverse Filtering (IAIF) algorithm illustrated in Figure 8 and explained in [4], is applied to each frame and results in the LPC representation of the vocal tract spectrum and and the waveform representation of the voice source.
- The LPC spectral envelope estimate of the voice source is calculated , and along with the LPC estimate of the vocal tract spectral envelope, is converted into LSF representation.
- The glottal flow waveform is used also for the acquisition of the F0 value as well as the Harmonic-to-Noise Ratio (HNR) values for a predetermined amount of sub-bands frequency.

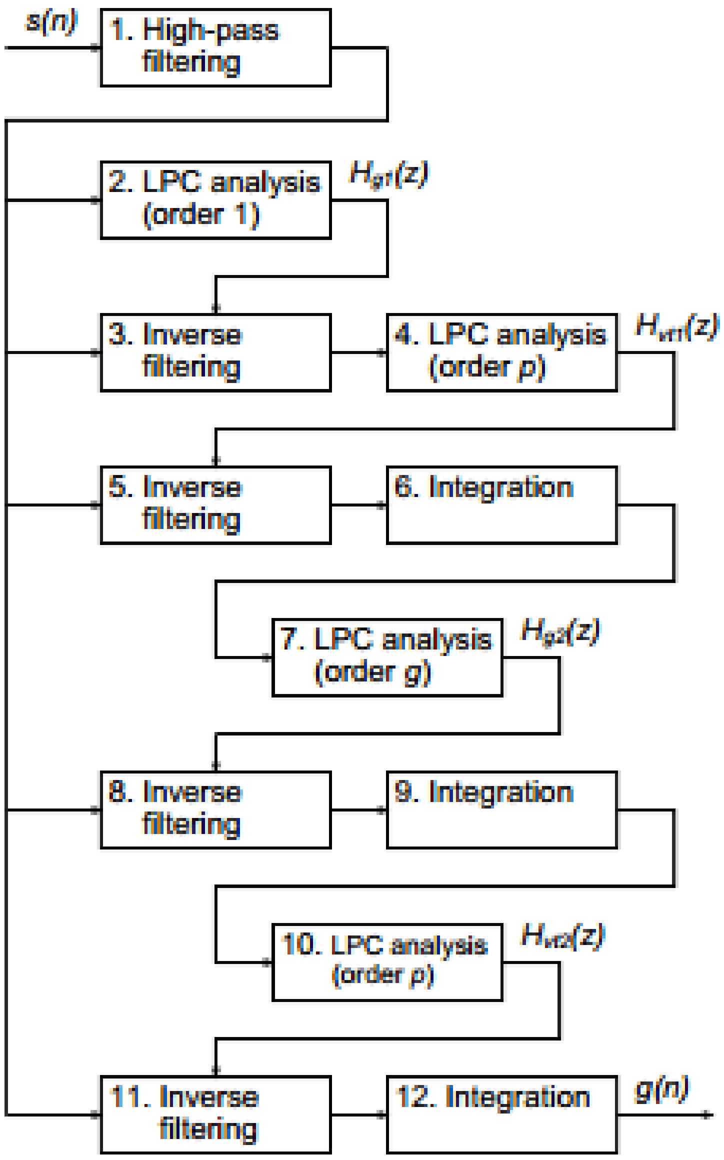

Figure 8: IAIF algorithm block diagram [12]

The output of the IAIF algorithm g(n) (estimated glottal flow signal) is used to generate the rest of the analysis parameters. A voicing decision is made based on the amount of zero-crossing and low-band energy. For voiced frames, the autocorrelation method is used to estimate the F0 value of the frame. The HNR is calculated from g(n). For unvoiced frames the HNR and F0 are set to zero. To model the excitation signal that is filtered by the vocal tract filter, the F0, HNR and the source LSF are used .

The final analysis vector of GlottHMM consists of single parameters for the F0 and log energy, around 5 parameters for HNR, 10-20 parameters for the glottal source LSF parameters and 20-30 parameters for the vocal tract LSF parameters.

6.1.2 Synthesis

To perform the synthesis, GlottHMM uses a method for the excitation generation based on the voiced/unvoiced decision instead of using a traditional mixed excitation model. The synthesis block diagram is illustrated in Figure 9.

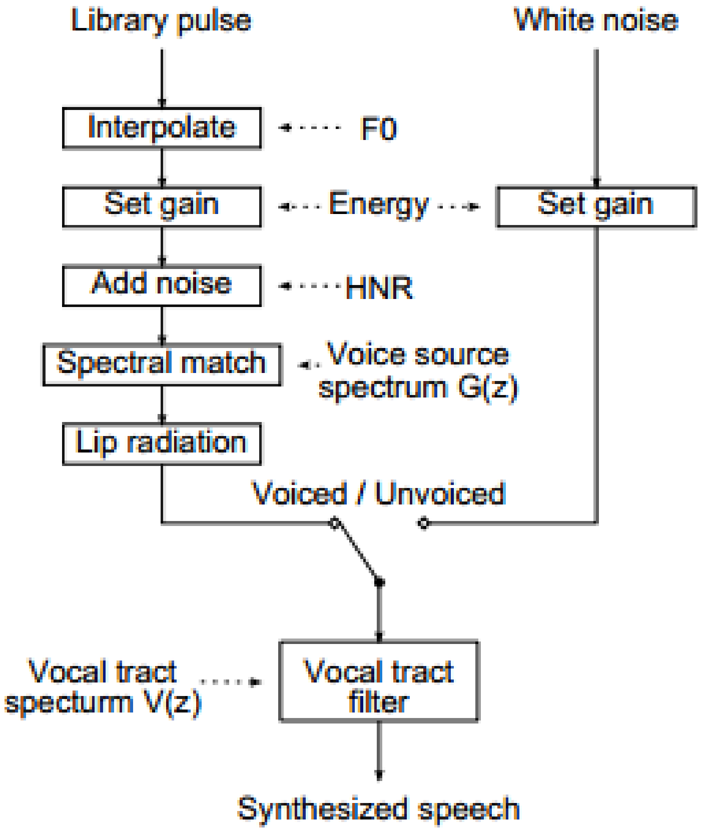

Figure 9: Synthesis block diagram for GlottHMM vocoder [4]

For the voiced frames, a fixed library pulse is obtained by glottal inverse filtering a sustained vowel signal. The library pulse is interpolated to match the target F0 value using cubic spline interpolation, and its energy is set to match the target gain obtained from the analysis vector.

Next, a HNR analysis is done to the library pulse. For each sub-band, noise is added to the real an imaginary parts of the FFT vector according to the differences between the obtained and the target HNR values.

The spectrum of the library pulse is matched to the spectrum of the target glottal pulse obtained from the analysis vector. The spectral matching is done by performing LPC analysis to the library pulse, and then filtering the obtained residual with the target synthesis filter. Finally, the lip radiation effect is added to the excitation by filtering it with a fixed differentiator.

For unvoiced frames, the excitation is generated as white Gaussian noise whose gain is set by the energy parameter of the analysis 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 derived from the vocal tract LSFs obtained from the analysis vector.

6.2 STRAIGHT

STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum) is the more established of the more sophisticated vocoding methods. Proposed by Kawahara in 1977, it has gone through extensive research and development since then. Is often the main reference to which other vocoders in HMM-based synthesis are compared, like in the case of this project.

For using HMM synthesis with STRAIGHT some modifications were made because of the high dimensionality of the parameters, and now the spectral envelope is represented as mel-frequency cepstral coefficients, and the corresponding aperiodicity measurements are averaged over five sub-bands frequency.

6.2.1 Analysis

In the parameter extraction (analysis) the main idea behind STRAIGHT is the extraction of a smoothed spectral envelope, which minimized the effect of periodicity interference in the analysis frames. So the spectral envelope is essentially independent of the speech excitation, which is a great feature with respect to speech transformation. The extraction of the spectral envelope can be found in [4].

The spectrum is represented as mel-frequency cepstral representation for the purpose of statistical modeling. The aperiodicity measurements are also transformed into a compressed representation.

The acquired analysis vector for STRAIGHT consists of the F0 value, 5 aperiodicity coefficients and 20-40 spectral MFC coefficients (MFCCs).

6.2.2 Synthesis

STRAIGHT synthesis is done in frame-by-frame basis by creating a mixed excitation signal of the length of two pulse periods based on the F0 and aperiodicity measurements. The harmonic pulse train is all-pass filtered with a randomized group-delay filter, which reduces the buzziness of the resultant synthesis. The acquired mixed excitation signal is convolved with the minimum phase Mel Log Spectrum Approximation (MLSA) filter derived from the frame's spectral MFCCs. Finally, the Pitch-Synchronous Overlap-Add (PSOLA) algorithm is applied to the synthesized frames to get the speech waveform signal [4].

As illustrated in Figure 10, the components for the mixed excitation are generated by sub-band filtering the voiced (impulse train) and unvoiced (white Gaussian

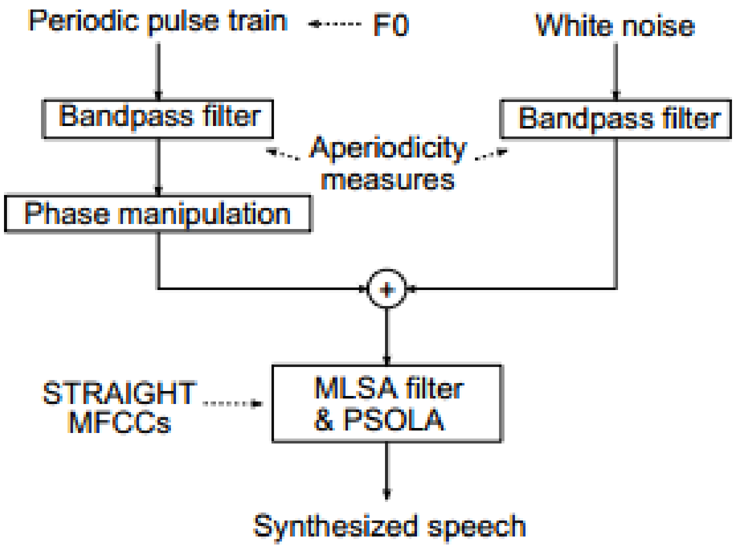

Figure 10: Synthesis block diagram for STRAIGHT vocoder [12]

noise) parts separately in the frequency domain. The band-pass filters used are determined by the aperiodicity coefficients so that the resultant sub-bands will have the same average lower-to-upper envelope ratio as the respective aperiodicity coefficient.

To adjust the phase characteristics of the excitation after the sub-band weighting, the pulse train component is all-pass filtered.

7 Adaptation

There are several styles of adaptation which affect both the possible application and the method of implementation. Firstly adaptation can be supervised in which case accurate transcriptions are available for all the adaptation data, or unsupervised in which case the required transcriptions must be hypothesis. Secondly, adaptation can be incremental, where adaptation data becomes available in stages or batch-mode, where all of the adaptation data is available from the start.

For cases where the adaptation data is limited, linear transform based schemes are currently the most effective form of adaptation. These approaches use the acoustic model parameters and require a transcription of the adaptation data.

7.1 Maximum Likelihood linear Regression

In maximum likelihood linear regression (MLLR), a set of linear transformations are used to map and existing model set such that the likelihood of the adaptation data is maximized.

There are two main variants of MLLR:

- Unconstrained MLLR: where separate transforms are trained for the means and variances. The effect of these transformations is to shift the component means and alter the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data.
- Constrained MLLR (CMLLR): where the transform for the mean and the variance is the same. The effect of these transformations is to shift the feature vector in the initial system so that each state in the HMM system is more likely to generate the adaptation data.

CMLLR is the form of linear transform most often used for adaptive training even with little amount of adaptation data[15]. For both forms of linear transformation, the matrix transformation may be full, block-diagonal, or diagonal. CMLLR is only implemented within HTK for diagonal covariance, continuous density HMMs due to computational reasons.

7.2 Regression Class Trees

A powerful feature of linear transform-based adaptation is that it allows all the acoustic models to be adapted using a variable number of transforms. When the amount of data is limited, a global transform is applied to all the Gaussian component in the model set, but as the amount of data increases, the HMM state components can be grouped into regression classes with each class having its own transform.

The number of transforms to use for any specific set of adaptation data can be determined automatically using regression class trees as illustrated in Figure 11. Each node represents a regression class (a set of Gaussian components that will

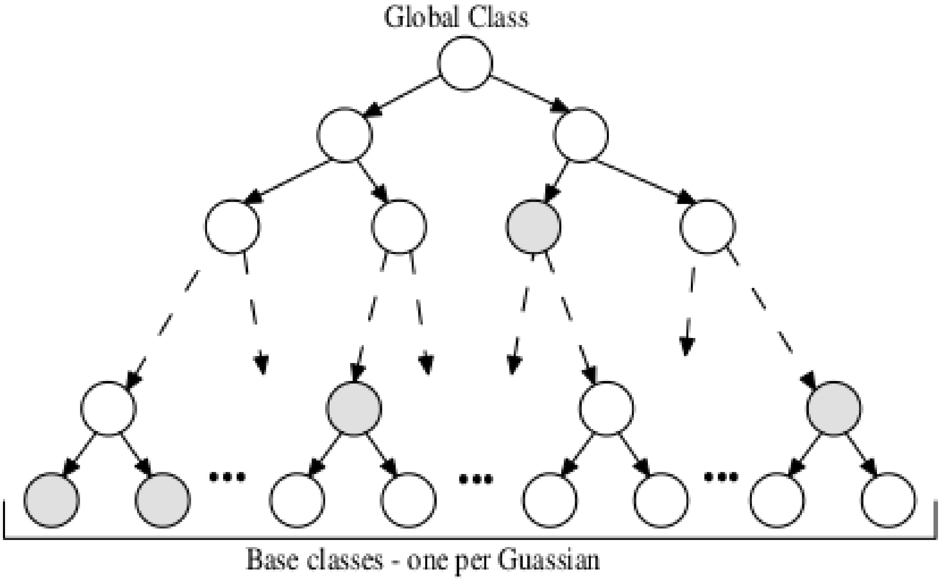

Figure 11: Regression class tree example [16]

share a single transform), the terminal nodes are called base classes. Then, for the given set of adaptation data, the tree is descended and the most specific set of nodes is selected for which there is enough data.

7.3 Maximum a Posteriori

It is possible to use standard statistical approaches to obtain robust parameter estimates rather than looking for a form of transformation to represent the differences between speakers. This is what maximum a posteriori (MAP) adaptation does. In MAP a prior over the model parameters is used to estimate the model parameters in addition to the adaptation data.

MAP adaptation effectively interpolates the original prior parameter values with those that would be obtained from the adaptation data alone. As the amount of adaptation data increases, the adaptation gets better and closer to the adaptation domain.

A variation of this technique exist and it is called Structural MAP (SMAP) [17]. It improves the MAP estimates obtained when the amount of adaptation data is small.

7.4 Adaptive Training

In the case of speaker independent, the training data includes large number of speakers. Therefore, training an acoustic model with different speakers "waste" a large number of parameters encoding the variability between speakers rather than the variability between spoken words which is the true aim. So what it can be done is to use adaptation transforms during the training step. This is known as speaker adaptive training (SAT).

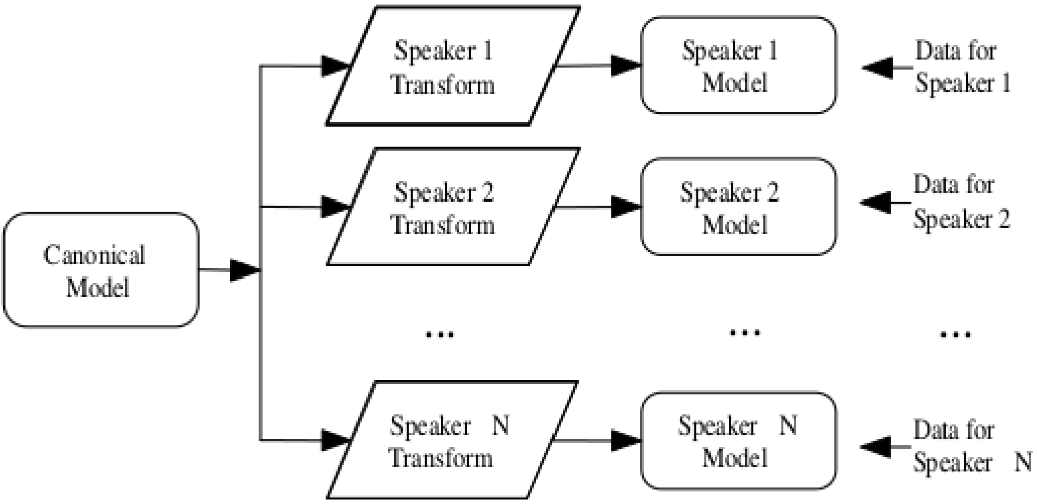

Figure 12: Speaker adaptive training example [16]

An example of this is illustrated in Figure 12. For each training speaker a transform is estimated and then the canonical model is estimated given all of these speaker transforms. The complexity of this method depend of the nature of the adaptation transform that can be split in three groups [16]:

- Model independent: These schemes do not make explicit use of any model information.
- Feature transformation: These transforms also act on the features but are derived, normally using ML estimation, using the current estimate of the model set.
- Model transformation: The model parameters, mean and possibly variances, are transformed.

The most common version of adaptive training uses CMLLR, since it is the simplest to implement.

8 Experiments

In this section, the work that has been done will be explained.

The experiments has been carried out with both male and female voices, and two different methods have been applied (Dependent models, Section 8.1, Adaptation, Section 8.2) in order to get the synthesized voices (Section 8.3).

Rergarding the vocoder, GlottHMM has been used and then the results have been compared with the ones obtained with STRAIGHT (see Section 9).

8.1 Dependent models

The first step in this project was to build dependent models for each emotion, but before some signal processing needed to be done, like sampling the audio files from 44KHz to 16KHz. Once this is done, the process for building the dependent models can be started.

In order to get these models the next steps were followed:

- Adjust GlottHMM configuration file (Section 8.1.1)
- Adjust HTS configuration file (Section 8.1.2)
- Extract features of the audio files (Section 8.1.3)
- Train the voice (Section 8.1.4)

8.1.1 Configuration File

GlottHMM use a configuration file to extract the features of an audio file (see Section 6.1), and it is very important have a good configuration to obtain good results after the training.

To try the configuration file, what it is done is to extract the features of a file and then synthesize it without any training, this is usually called resynthesis. The synthesized file must be very similar to the original one.

A configuration file has been created for each emotion and for some of them the result was better than with others, that is the reason why not all the emotions have the same final quality. For example, with the anger emotions the synthesized file is not as similar to the original as the sad one, and this is reflected in the final quality of the voice after the training. This is illustrated in Figures 13, 14.

Looking into the different configuration files for the different emotions, some little changes can be seen between them. These changes can be found in the f0 estimation of the analysis, where part of the emotion is located (see Section 3). The rest of the configuration file is the same for all the emotions and it can be also find the parameters that can be extracted in the analysis (can be true or false) or the ones that will be used in the synthesis. An example of a configuration file can be found in Appendix B.

In this f0 estimation some values can be tuned:

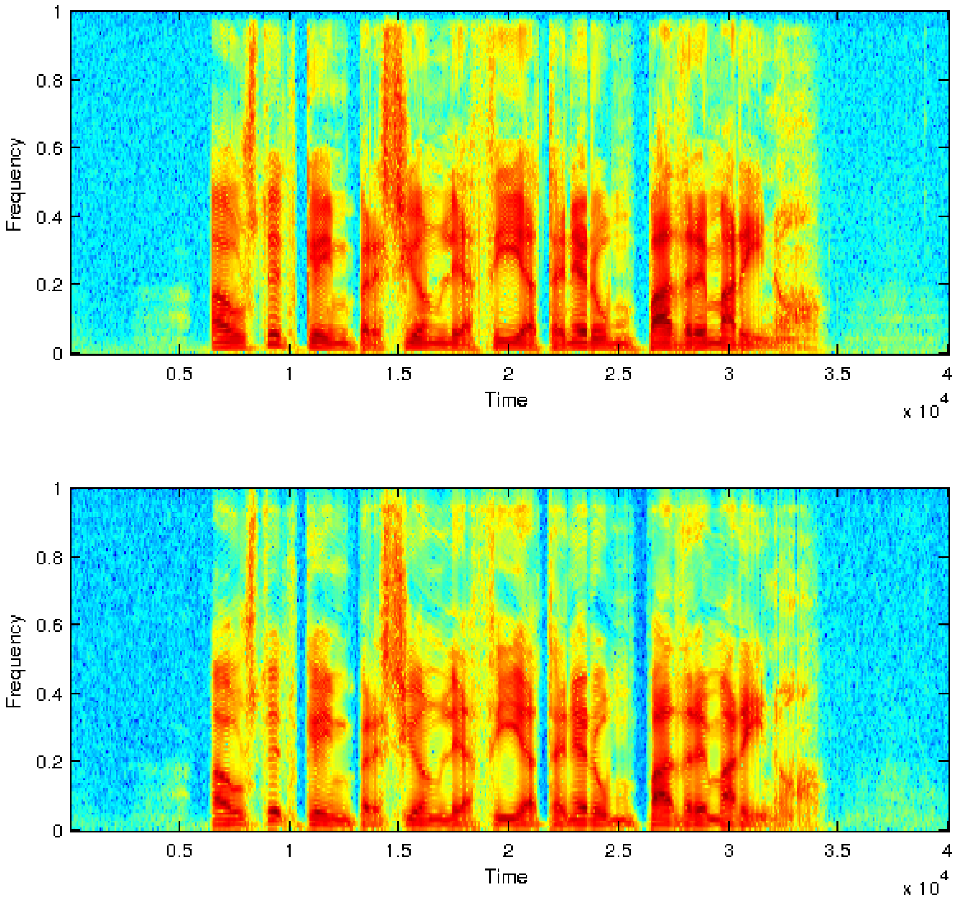

Figure 13: Spectrogram for the angry emotion of the original file (above) and the synthetic file after resynthesis (below)

- F0_MIN: Minimum fundamental frequency
- F0_MAX: Maximum fundamental frequency
- VOICING_THRESHOLD: Voicing threshold with respect to gain in the low-frequency band, so the speech frames under this value will be classified as unvoiced
- ZCR_THRESHOLD: Zero-crossings threshold. Speech segments that have more zero crossings than the threshold value are classified as unvoiced

For the male voice the minimum f0 is between 30 and 50 Hz and the maximum f0 is between 260 and 300 Hz, the voicing threshold is between 70 and 90, and zero-crossing threshold is 110 or 120. For the female voice these values are totally different than for the male, for example the maximum f0 is bigger than in the case of the male voice.

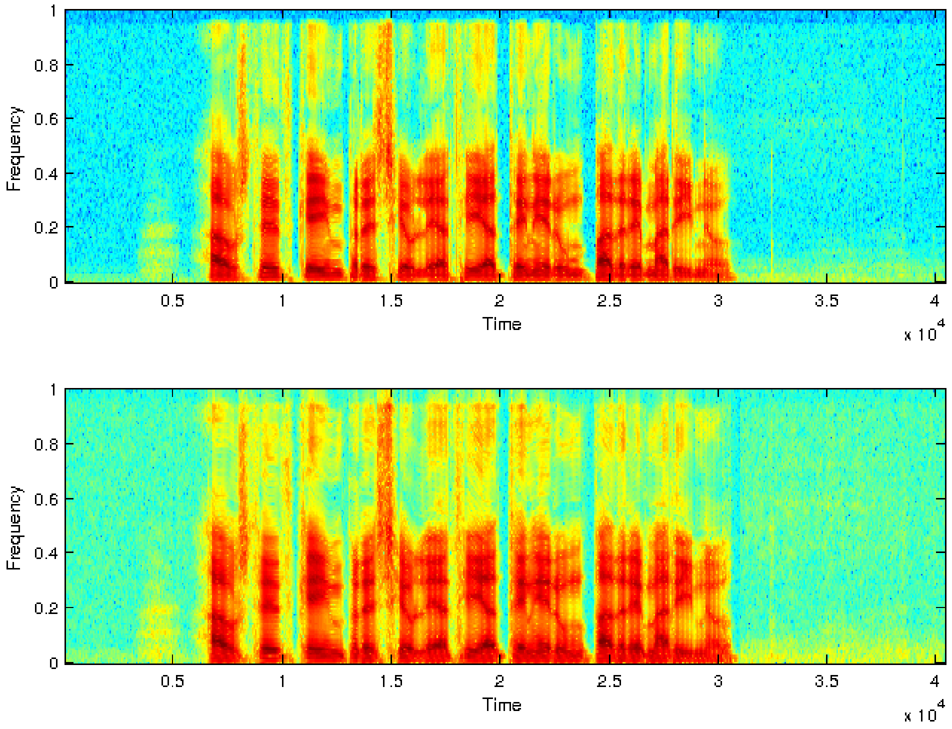

Figure 14: Spectrogram for the sadness emotion of the original file (above) and the synthetic file after resynthesis (below)

If all the f0 values obtained after the feature extraction (Section 8.1.3) of all the files used for training are plot, the different f0 for each emotion can be plot. Some examples for the male voice can be found in Appendix A.

8.1.2 HTS Configuration File

In this configuration file the path where the features are going to be extracted is given, and also the streams of features that are going to be used in the training. In the experiment the next streams have been used:

- f0: fundamental frequency
- lsf1: spectral envelope LSFs
- gain1: gain
- flow: source LSFs
- hnr_i: harmonic to noise ratio with bands

Looking into this file or in the training script it can be seen that the dimension (or size) of these streams is 31 for the lsf (10 for lsf, 10 for the delta coefficients, 10 for the delta-delta coefficients and 1 for the gain), 10 for the flow, 5 for the hnr and 1 for f0.

8.1.3 Feature Extraction

The next step is the feature extraction of the audio files that are going to be used in the training. The features that are going to be extracted can be selected in the GlottHMM configuration file.

The streams that contains the features will be used in the training for building the voice model, so the features have to represent the voice.

In the last step of the feature extraction, the cmp binary files will be created. These files are vectors that contain the information extracted for each file and will be used in the training.

8.1.4 Training

Once the feature extraction is done the training step can be started. For this, a folder with the features (cmp files) and a folder with the time alignment labels is needed.

The time alignment labels can be extracted using a front-end. For this a question file will be needed.

The training is HMM based with five states Gaussian and leaf nodes for the different trees (see Section 5). For each training two models are generated due to a reclustering is applied to obtain better results. Once the training is done and the models are created, one thing that can be done is to realign the training labels using the model that has been created with the training step and train a new model with this realigned labels. This can be done as much times as wanted, and in the case of this project it has be done two times (so we have three rounds) with the male voice in both cases (dependent models and average model) and with the female voice just with the dependent models due to that with the female average model a lot of computation time is needed (several weeks). For realigning the labels the tool *HSMMAlign* was used.

So in the end a lot of models are generated due to the reclustering and the realignment. For the dependent models 6 models are created, 2 for reclustering in each training, and 3 trains are performed. In the case of the male average model the same 6 models are created but the adaptation is done for each emotion with the last one of the reclusterings models so in the end 3 models are generated for each emotion ,so 15 models for the male average model.

As a test is going to be done some new utterances for it need to be synthesized (see Section 9.1) and they have to be the best ones, so before the synthesis of these utterances, one of the generated models has to be chosen as the best one. This has been done as explained in Section 9.1.

8.2 Adaptation

The adaptation consist of transfer the capabilities of one sepeaker to an average voice model (in speaker adaptive training, sat) as is explained in Section 7. So basically the steps that has to be followed are the same that with the previous method (Section 8.1), with the difference that this time an average voice model has been build with all the emotions to have a more robust model.

In order to do this, all the extracted features (cmp files) for the previous method will be placed in the same folder, and the same goes for the time alignment labels, and the SAT flag (Section 7.4) has to be set to one in the training script, so there will be only one model (the average).

At this point is where the method differs from the previous one. Now is where the adaptation take place. So an adaptation to this model has been done with every emotion which generates a new model for each emotion. For doing the adaptation the features of the files used in the adaptation have to be extracted.

According to what was told in Section 7 the type of adaptation is supervised and batch-mode, so different adaptation techniques could have been applied here like MLLR, CMLLR, MAP (see Section 7) or a combination of some of them. One of these combinations is the one called CSMAPLR [18], that is a combination of CMLLR, MAP and SMAP.

CSMAPLR make use of the information contained in the connections of the tree structure of the HMMs, which leads into more stable prior information transference into the adapted models. CMLLR do not do this as Figure 15 illustrates.

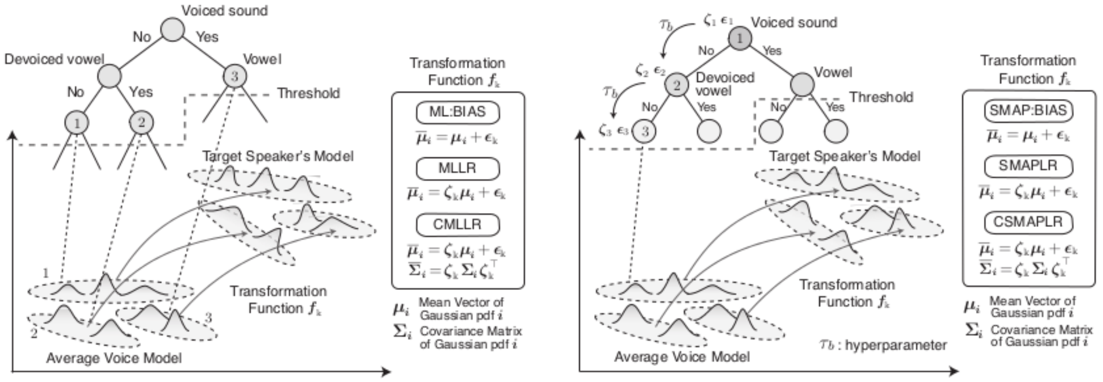
Voiced sound
No
Yes
Devoiced vowel
Vowel
No
Yes
1
2
3
Threshold
Target Speaker's Model
Transformation Function f_k
Average Voice Model
1
2
3
Transformation Function f_k
ML:BIAS
$\overline{\mu}_i = \mu_i + \epsilon_k$
MLLR
$\overline{\mu}_i = \zeta_k \mu_i + \epsilon_k$
CMLLR
$\overline{\mu}_i = \zeta_k \mu_i + \epsilon_k$
$\overline{\Sigma}_i = \zeta_k \Sigma_i \zeta_k^\top$
μ_i Mean Vector of Gaussian pdf i
Σ_i Covariance Matrix of Gaussian pdf i
τ_b
$\zeta_1\ \epsilon_1$
Voiced sound
1
No
Yes
$\zeta_2\ \epsilon_2$
2
Devoiced vowel
Vowel
τ_b
No
Yes
No
Yes
Threshold
$\zeta_3\ \epsilon_3$
3
Target Speaker's Model
Transformation Function f_k
Average Voice Model
τ_b : hyperparameter
Transformation Function f_k
SMAP:BIAS
$\overline{\mu}_i = \mu_i + \epsilon_k$
SMAPLR
$\overline{\mu}_i = \zeta_k \mu_i + \epsilon_k$
CSMAPLR
$\overline{\mu}_i = \zeta_k \mu_i + \epsilon_k$
$\overline{\Sigma}_i = \zeta_k \Sigma_i \zeta_k^\top$
μ_i Mean Vector of Gaussian pdf i
Σ_i Covariance Matrix of Gaussian pdf i

Figure 15: Differences between CMLLR and CSMAPLR [18]

The adaptation technique that has been used in this project is the CSMAPLR adaptation followed by a MAP adaptation. For the CSMAPLR 256 regression tree nodes have been used.

Two rounds of CSMAPLR have been used, followed by a MAP adaptation. Doing this, the log probability per frame improves, which leads in a better adaptation.

The CSMAPLR adaptation can be tuned a little bit with some thresholds which can change the depth of the adaptation, so the emotion level can be tuned with these thresholds. Also changing the regression tree nodes can affect the adaptation. As the adaptation has been done using a good amount of data the regression trees can be big, so a better node will be selected.

For the adaptation all the data of the training for one emotion have been used to replicate the experiment that were done with STRAIGHT, but a big amount of data is not required for a good adaptation using this technique.

8.3 Synthesis

Once the model are created the process for synthesis is the same in both cases with a little exception when synthesizing labels that have not been seen during the training. In the case of the dependent models (Section 8.1) the models has to be changed to know these new labels, in the case of the adaptation these labels are given when adapting, so the new models are created new them. For this change a HTK tool is used and it is called *HHEd* [19].

When this is done, the first step is to extract the features of the label that is going to be synthesized, as it was explained in 6.1, using the models created during the training. This extraction is done with another tool called *HMGenS*. This tool extracts the lsf, flow,logF0 and hnr of the label file.

The next step is to extract information of the extracted features to generate the F0, LSF, LSFsource, HNR and GAIN to use the synthesis tool of GlottHMM to generate the audio file. When synthesizing, the global variance [20] (GV) can be used or not. It compensates the over-smoothing effect.

Also for the synthesis the HTS engine can be used but it requires some transformations to the models.

9 Results

In this section the results for the experiments explained in Section 8 will be showed. Different perceptual test has been carried out for male and female voiced evaluating the two methods used for building the models (see Sections 8.1, 8.2).

9.1 Training and Test Data

For the different methods and genres the amount of data have been different.

9.1.1 Male voice

In both cases the speech data base used was recorded with a single male professional voice actor. It is called SEV (Spanish Emotional Voices) and it consists of the four emotions and the neutral speech.

For the different techniques the amount of data for the male voice is:

- Dependent model (Section 8.1): the same amount of data has been used for each emotion. The total amount of data per emotion is 489 utterances (around one hour of recording speech)

- Adaptation model (Section 8.2): in this case the amount of data used for the training was all the data of the dependent models, which is 2445 utterances, and then all the data for each emotion (489 utterances) to perform the adaptation.

Before synthesizing the test labels one of the created models has to be chosen as the best, for that reason a validation test has been done with 15 utterances of other speaker. Once the model has been chosen 20 utterances from the Albayzin evaluation have been used for the perceptual test.

9.1.2 Female voice

For the female voice, a database with professional speech actors was used for building the dependent models and for the average model the same database plus other two amateur and two professional databases were used. In both cases the data used contains four emotions and neutral speech. For the different techniques the amount of data for the female voice is:

- Dependent model (Section 8.1): with the female voice the amount of data is not the same for each emotion, for the neutral emotion less utterances (504) are used than with the other emotions (around 605 utterances).

- Adaptation model (Section 8.2): for the female voice, as it has less quality and with the average model a more robust model is wanted, a lot of data was used: all the data of the dependent models (2922 utterances) plus the data of other databases (2808 utterances) , which makes a total of 5730 utterances.

For the female voice no so many models have been created due to that in the adaptation the amount of data is too big and it takes more than a week to perform one training. In the dependent models the same models than in the male voice were obtained but the realignment did not have good results in this case. The validation and test utterances are the same as for the male voice. All this information is compacted in Table 1.

#utt **Voice**	**anger**	**happiness**	**neutral**	**sadness**	**surprise**	**average**	**validation**	**test**
Male	489	489	489	489	489	2445	15	20
Female	605	603	504	605	605	5730	15	20

Table 1: Number of utterances used in training, validation and test

9.2 Test

A perceptual test following the Latin Square testing strategy [21] has been done through a web interface hosted in the Speech Technology Group (GTH) and distributed among family and friends without knowledge of the system details. It has been recommended the use of headphones when doing the test.

As the purpose of the test is to compare GlottHMM with STRAIGHT the test has been divided into two parts. In each part 20 sample utterances are presented. In the first one the quality, the naturalness and the emotional strength of the vocoder are tested, so two audio files are showed (A and B) and the next questions are asked in the test:

- Choose the file that represent better the emotion (A or B)
- Choose the file that is more natural (A or B)
- Choose for both files the level of emotion (from very poor to very high)
- Choose for both files the speech quality (from very poor to very high)

In the second one, the test is focused on the speaker voice, so it is asked to choose the file (A or B) with the voice more similar to the original speaker. In this second part 4 reference audios with the original voice of the speaker are given to compare with the synthetic voices.

In the test the listeners are not going to listen to all the audio files, thanks to the Latin Square strategy, so nobody has control over the test.

In order to obtain enough results, three test have been performed: one for the male voice using the dependent models, another one for the female voice using the dependent models and the last one for both male and female voices using the average models with adaptation. The reason for doing the adaptation test together was the difficulty to find listeners for the test.

9.3 Perceptual Test Results

Here the results of the perceptual test for the two genres and the different techniques used are exposed. The tables with the values can be found in Appendix C.

9.3.1 Preference, Naturalness and Similarity

In this section three results will be studied:

- Preference: which of the files (A or B) represent better the emotion.
- Naturalness: which of the files (A or B) is more natural.
- Similarity: which of the files (A or B) is more similar to the original speaker.

These results are illustrated in Figures 16, 17, 18 and 19 where PREF denotes the preference of the file, NAT the naturalness of the file and SIM the similarity of the file (ST for STRAIGHT and Gl for GlottHMM) chosen by the listener.

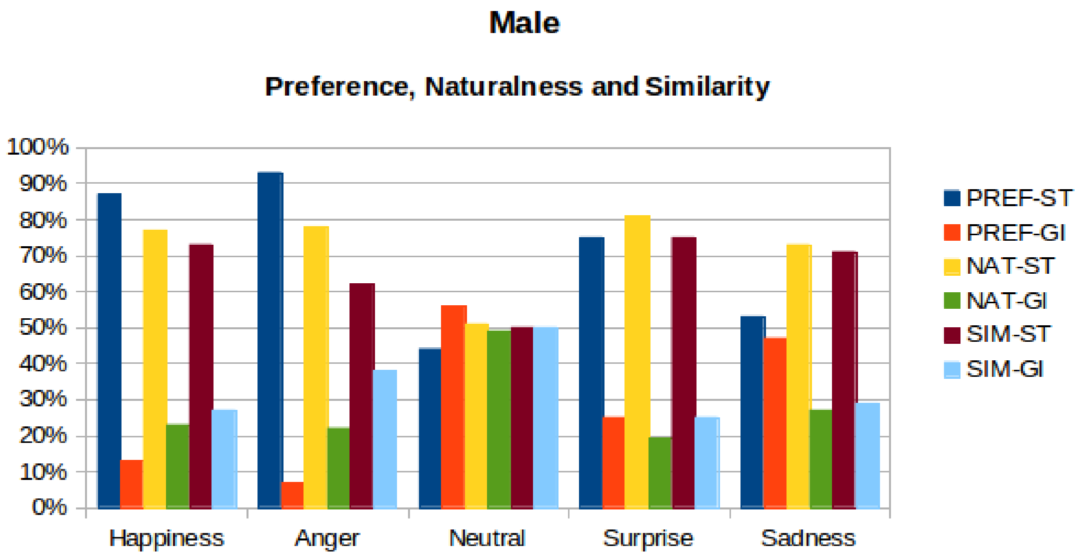

Figure 16: Preference, Naturalness and Similarity for the male voice

As it can be seen in Figure 16, for all the emotions STRAIGHT is preferred over GlottHMM, with the exception of the sadness emotion, where GlottHMM is close to STRAIGHT regarding the preference. For the neutral speech GlottHMM is preferred regarding the preference and tied with STRAIGHT regarding the naturalness and the Similarity.

For the average male voice the situation is worst in general, excluding that in the Similarity with happiness and sadness emotions GlottHMM is closer to STRAIGHT that with the dependent model but below it.

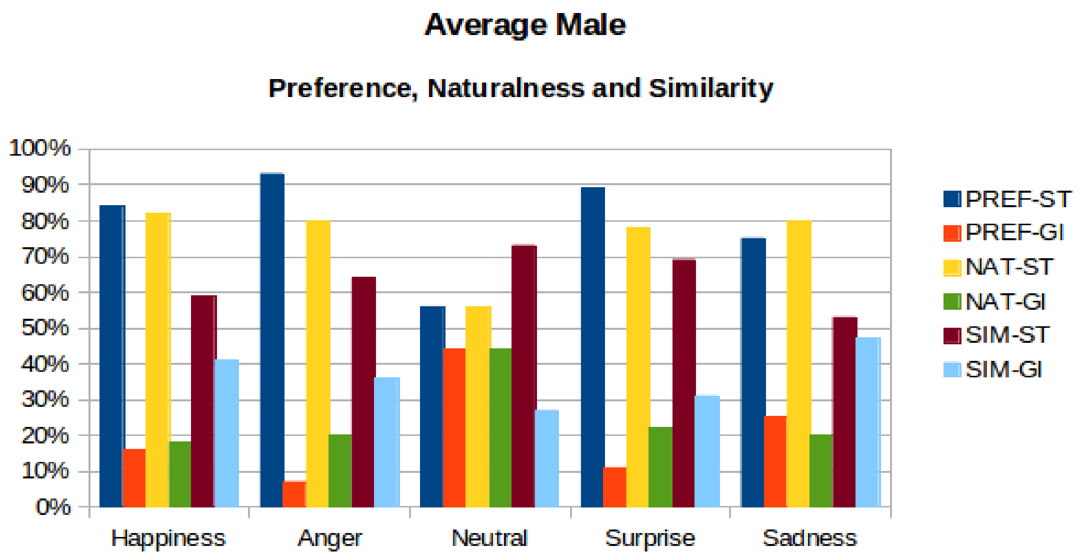

Figure 17: Preference, Naturalness and Similarity for the average male voice

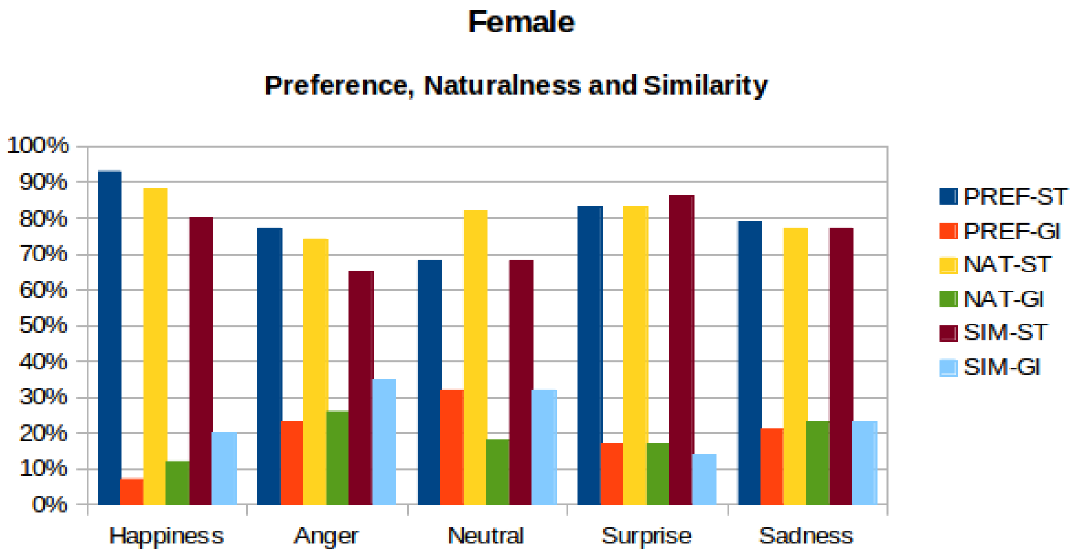

Figure 18: Preference, Naturalness and Similarity for the female voice

Figure 18 shows that the situation for the female voice is worst than with the male voice, GlottHMM in not even close to STRAIGHT. But in this case it seem that the use of an average voice model helps, this can be seen in Figure 19, where there is not preference between STRAIGHT and GlottHMM regarding the neutral speech, although in the naturalness and the Similarity STRAIGHT is chosen, and also in the sadness emotion there is not so much different in the Similarity as in the

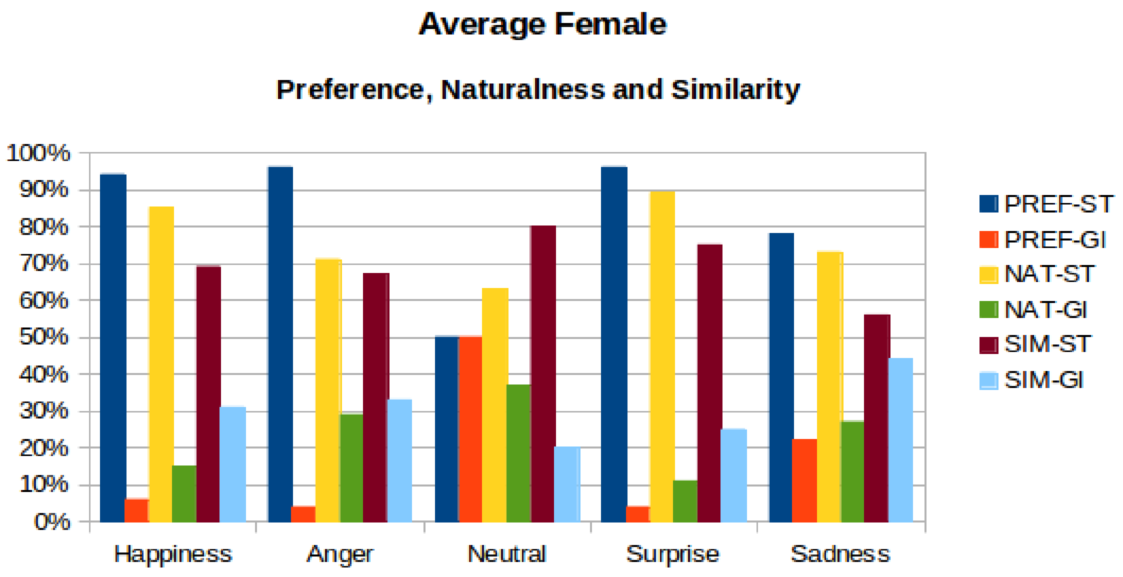

Figure 19: Preference, Naturalness and Similarity for the average female voice

dependent model.

9.3.2 Emotional Strength

Here the emotional strength level will be compared for the two vocoders.

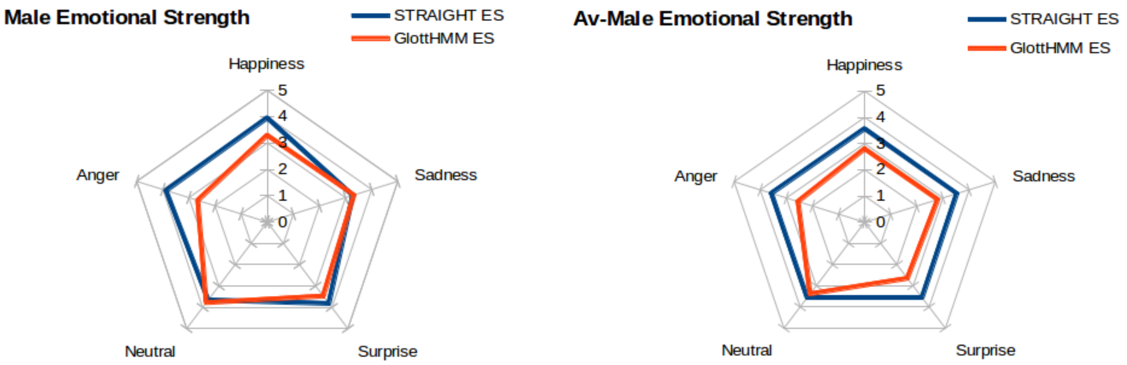

Figure 20: Emotional Strength for the male voice

Looking at Figure 20 it can be seen that the level of emotional strength is bigger for the dependent models although the difference is greater for GlottHMM. For the dependent model, the difference between the vocoders is practically non-existent for both sadness and surprise emotions and for the neutral speech but in the average voice model the difference is bigger and the two vocoders are similar only in the neutral speech.

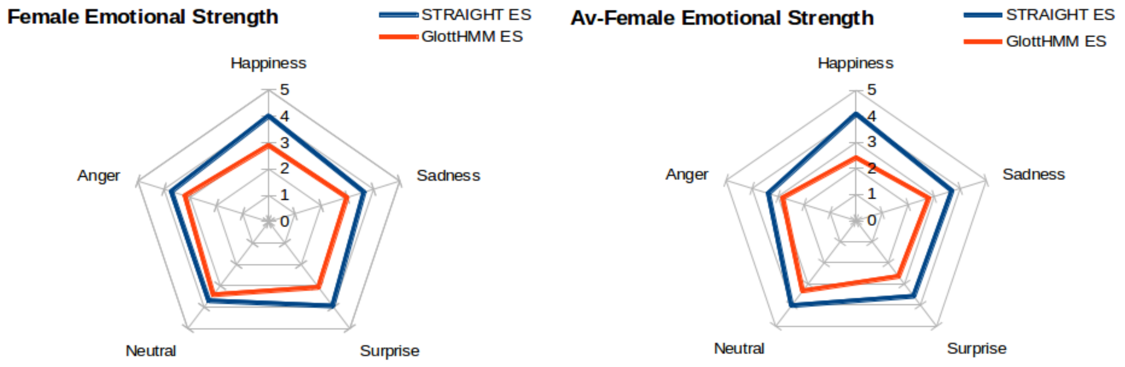

Figure 21: Emotional Strength for the female voice

For the female voice it can be seen in Figure 21 that the situation for STRAIGHT is more or less the same for the dependent model and better in some emotions comparing with the male voice. It can be also appreciated that the anger emotion is stronger in the female voice. Of the two methods for the female voice just the neutral speech of the dependent model is similar in both vocoders and in the emotions STRAIGHT has higher level of strength.

More information can be obtained from the boxplots (appendix D), where it can be seen the median for all the emotions and neutral speech, and also where is the 95% of the data concentrated. Also the maximum and minimum values as the outliers (represented as circles) can be seen.

9.3.3 Speech Quality

In this section the Mean Opinion Score (MOS) is presented in Figures 22 and 23 and also in the boxplots that can be found in Appendix D.

In a first look at Figures 22 and 23 it can be seen that STRAIGHT has a better speech quality than GlottHMM, and that it is in general good. In this case GlottHMM only evens STRAIGHT in the neutral speech of the male dependent model. It can also be appreciated that the speech quality of the female voice is worst in GlottHMM. In STRAIGHT this difference between genres is not big, and as it can be seen is just for the sadness emotion, that improves in the average voice model.

The speech quality of GlottHMM is acceptable according to the results of the test for the male voice, and for the female voices is not always acceptable, but it is known that the GlottHMM works better with male voices.

The bloxpots for the speech quality can be found in Appendix D.

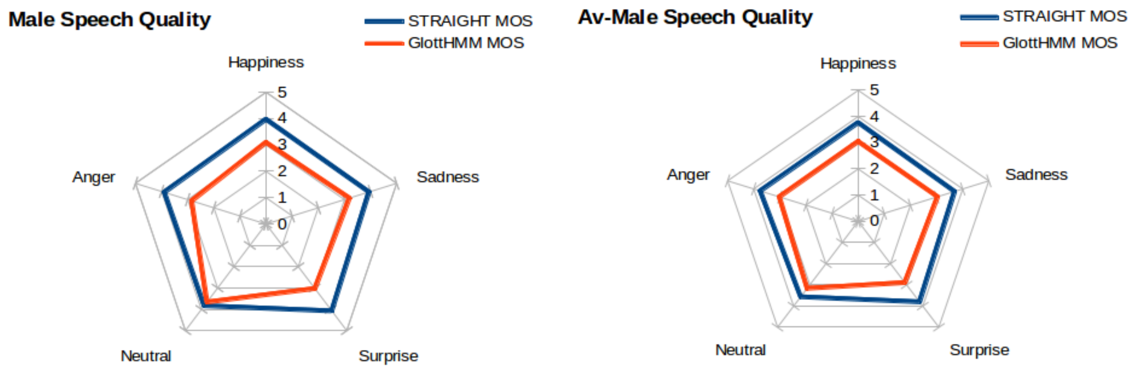

Figure 22: Speech Quality for the male voice

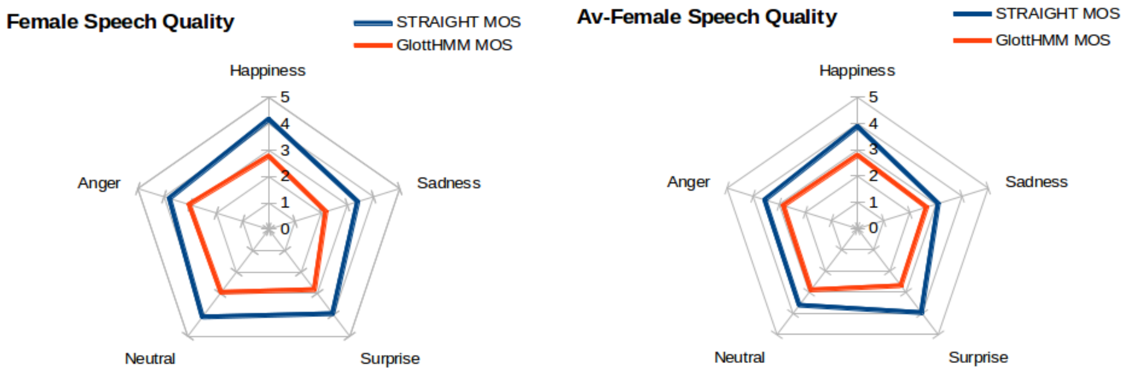

Figure 23: Speech Quality for the female voice

10 Discussion and Conclusion

10.1 Discussion

Although the test said that STRAIGHT is better regarding the emotional strength, one thing that can be appreciated listening the synthetic voices for both vocoders is that the STRAIGHT voices have more gain (they are louder) that GlottHMM ones, and this can make the test listeners think that in some cases the emotional strength is higher, or it can help to select STRAIGHT in the similarity test when the decision is not clear enough. This difference is bigger in the female voices.

As it has been told in this project a lot of data have been used to match the experiments that were done with STRAIGHT, but the main point of adaptation is that it does not need as much data to perform a good adaptation. Similar quality can be obtained with less adaptation data.

10.2 Conclusion

The main conclusion that can be extracted of this project given the results of the perceptual test is that regarding the emotional speech synthesis GlottHMM is not yet at the level of STRAIGHT that has been proved very robust in all the situations.

The neutral speech is very similar or even better in GlottHMM that STRAIGHT for the dependent models and the male voice. For this dependent models in male voices GlottHMM emotional strength is similar to the STRAIGHT (not for the anger and happiness emotions), but in the speech quality is better STRAIGHT regarding the emotions.

As it was already known, the results for the male voice with GlottHMM are better than the ones with the female voice. This is due to the higher the frequency the less important the glottal pulse becomes, and female voices have a higher frequency.

References

[1] J. Lorenzo-Trueba, R. Barra-Chicote, J. Yamagishi, O. Watts, and J. M. Montero, "Towards speaking style transplantation in speech synthesis," in *Proc. 8th ISCA Speech Synthesis Workshop*, 2013.

[2] G. O. Hofer, "Emotional speech synthesis," Master's thesis, University of Edinburgh School of Informatics, 2004.

[3] J. Lorenzo-Trueba, R. Barra-Chicote, T. Raitio, N. Obin, P. Alku, J. Yamagishi, and J. M. Montero, "Towards glottal source controllability in expressive speech synthesis," in *Proc. Interspeech 2012*, (Portland (Oregon), USA), 2012.

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

[5] J. L. Flanagan, *Speech Analysis, Synthesis and Perception.* second ed., 1972.

[6] M. University, "A brief historical introduction to speech synthesis: A macquarie perspective." last accesed 11-03-2014.

[7] C. Darwin, *The expression of the emotions in man and animals.* New York D. Appleton and Co., 1897.

[8] D. Goleman, *Emotional Intelligence.* A Bantam book, Bantam Books, 2006.

[9] A. C. Aarón Blanco, "Inteligencia emocional."

[10] J. Yamagishi, *An Introduction to HMM-Based Speech Synthesis.* 2006.

[11] K. Tokuda, H. Zen, and A. W. Black, "An hmm-based speech synthesis system applied to english," 2002.

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

[13] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for hmm-based speech synthesis.," in *ICSLP*, vol. 98, pp. 29–31, 1998.

[14] 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.

[15] D. Povey and K. Yao, "A basis method for robust estimation of constrained mllr.," in *ICASSP*, pp. 4460–4463, IEEE, 2011.

[16] M. Gales and S. Young, “The application of hidden markov models in speech recognition,” *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

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

[18] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, “Analysis of speaker adaptation algorihms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm,” *IEEE Transactions on Audio, Speech and Language Processing*, 2008. In print.

[19] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[20] T. Toda and K. Tokuda, “A speech parameter generation algorithm considering global variance for hmm-based speech synthesis,” *IEICE - Trans. Inf. Syst*, pp. 816–824, 2007.

[21] L. Gao, “Latin squares in experimental design,” Michigan State University, 2005.

A F0 examples

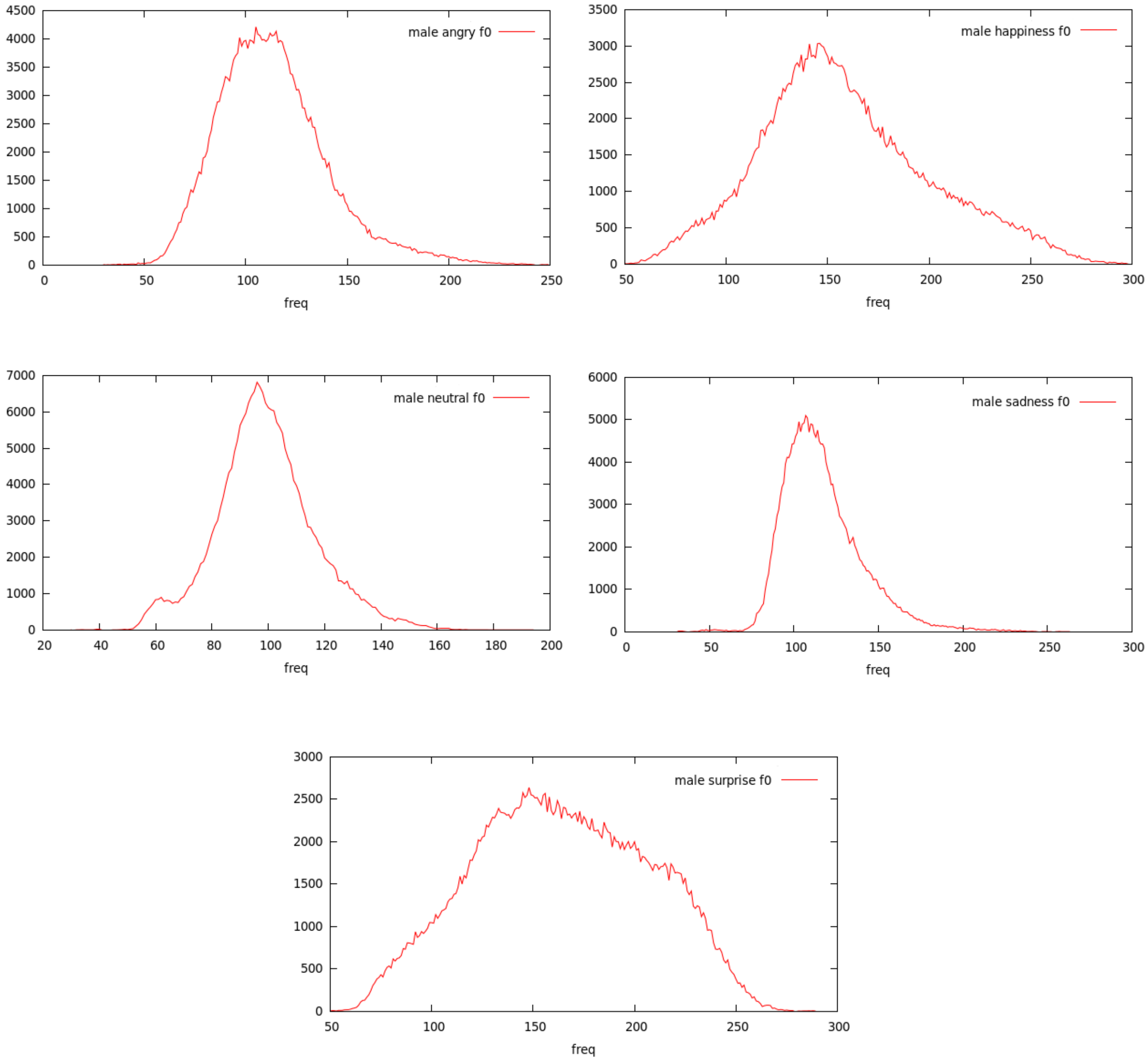

Figure A1: Male F0 representation

B GlottHMM Configuration File Example

The GlottHMM configuration file showed in this section corresponds to the sadness emotion.

```
##############################################################
#    Sadness 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 =              2.0;
        NOISE_REDUCTION_DB =                    30.0;

# Analysis:
        # Analysis: General parameters:
                PITCH_SYNCHRONOUS_ANALYSIS =            true;
                INVERT_SIGNAL =                         true;   # Remember to set true e.g. for MV voice (it's inverted)
                HP_FILTERING =                          true;
                HPFILTER_FILENAME =                     "/home/romeroa2/GlottHMM/glott_anasyn/hp_16khz";

        # Analysis: Parameters for F0 estimation:
                F0_MIN =                                30.0;
```

```
            F0_MAX =                        300.0;
            VOICING_THRESHOLD =             90.0;
            ZCR_THRESHOLD =                 120.0;
            USE_F0_POSTPROCESSING =         true;
            RELATIVE_F0_THRESHOLD =         0.005;
            F0_CHECK_RANGE =                10;
            USE_EXTERNAL_F0 =               false;
            EXTERNAL_F0_FILENAME =          "filename.F0
              ";

    # Analysis: Parameters for extracting pulse
      libraries:
            MAX_NUMBER_OF_PULSES =          10000;
            PULSEMAXLEN =                   45.0;
            RESAMPLED_PULSELEN =            10.0;
            WAVEFORM_SAMPLES =              10;
            MAX_PULSE_LEN_DIFF =            0.05;
            EXTRACT_ONLY_UNIQUE_PULSES =    true;
            EXTRACT_ONE_PULSE_PER_FRAME =   true;

    # Analysis: Parameters for spectral modeling:
            USE_IAIF =                      true;
            LPC_ORDER_GL_IAIF =             8;
                        # Order of the LPC analysis for
              voice source in IAIF
            USE_MOD_IAIF =                  true;
                        # Modified version of IAIF
            LP_METHOD =                     "LPC";
                        # Select between "LPC" / "WLP" /
              "XLP"
            LP_STABILIZED =                 false;
            LP_WEIGHTING =                  "GCI";
                        # Select between "STE" / "GCI"
            FORMANT_PRE_ENH_METHOD =        "NONE";
                        # Select between "NONE" / "LSF" /
               "LPC"
            FORMANT_PRE_ENH_COEFF =         0.8;
            FORMANT_PRE_ENH_LPC_DELTA =     20.0;
                        # Only for LPC-based method

    # Analysis: Select parameters to be extracted:
            EXTRACT_F0 =                    true;
            EXTRACT_GAIN =                  true;
            EXTRACT_LSF =                   true;
            EXTRACT_LSFSOURCE =             true;
```

```
        EXTRACT_HNR =                          true;
        EXTRACT_HARMONICS =                    true;
        EXTRACT_H1H2 =                         true;
        EXTRACT_NAQ =                          true;
        EXTRACT_WAVEFORM =                     false;
        EXTRACT_INFOFILE =                     true;
        EXTRACT_PULSELIB =                     false;
        EXTRACT_SOURCE =                       false;

# Synthesis:
    # Synthesis: General parameters:
        SYNTHESIZE_MULTIPLE_FILES =            false;
        SYNTHESIS_LIST =                       "synthesis_list_filename";
        USE_HMM =                              false;

    # Synthesis: Choose excitation technique and related parameters:
        USE_PULSE_LIBRARY =                    false;
        GLOTTAL_PULSE_NAME =                   "/home/romeroa2/GlottHMM/glott_anasyn/gpulse";
        PULSE_LIBRARY_NAME =                   "/home/traitio/Desktop/pulselibrary/pulse_libraries/lpc5/lpc5";
        NORMALIZE_PULSELIB =                   false;
        USE_PULSE_CLUSTERING =                 false;
        USE_PULSE_INTERPOLATION =              true;
        AVERAGE_N_ADJACENT_PULSES =            1;
        ADD_NOISE_PULSELIB =                   false;
        MAX_PULSES_IN_CLUSTER =                2000;
        NUMBER_OF_PULSE_CANDIDATES =           200;
        PULSE_ERROR_BIAS =                     0.3;
        MELSPECTRUM_CHANNELS =                 22;
        CONCATENATION_COST =                   0.1; # 2.0
        TARGET_COST =                          1.0;
        PARAMETER_WEIGHTS =                    [0.2, 0.2, 0.0, 0.05, 0.15,0.4, 0.0, 0.0, 0.0, 0.0];
        #PARAMETER_WEIGHTS =                   [0.0, 1.0, 1.0, 2.0, 3.0, 5.0, 1.0, 1.0, 1.0, 0.0];
        # Parameter names                      [LSF SRC HARM HNR GAIN F0 WAV H1H2 NAQ PCA/ICA]

    # Synthesis: Select used parameters:
        # F0, Gain, and LSFs are always used
        USE_LSFSOURCE =                 true;
        USE_HNR =                       true;
        USE_HARMONICS =                 false;
        USE_H1H2 =                      false;
        USE_NAQ =                       false;
        USE_WAVEFORM =                  false;
        USE_MELSPECTRUM =               false;
        USE_PULSE_PCA =                 false;

# Synthesis: Set level and band of voiced noise:
        NOISE_GAIN_VOICED =             0.01;
        NOISE_LOW_FREQ_LIMIT =          2400.0; # Hz

# Synthesis: Smoothing of parameters for analysis-synthesis:
        LSF_SMOOTH_LEN =                5;
        LSFSOURCE_SMOOTH_LEN =          3;
        GAIN_SMOOTH_LEN =               5;
        HNR_SMOOTH_LEN =                15;
        HARMONICS_SMOOTH_LEN =          5;

# Synthesis: Gain related parameters:
        GAIN_UNVOICED =                 1.0;
        NORM_GAIN_SMOOTH_V_LEN =        0;
        NORM_GAIN_SMOOTH_UV_LEN =       0;
        GAIN_VOICED_FRAME_LENGTH =      25.0;
        GAIN_UNVOICED_FRAME_LENGTH =    20.0;

# Synthesis: Postfiltering:
        POSTFILTER_METHOD =             "LSF"; # Select between "NONE" / "LSF" / "LPC"
        POSTFILTER_COEFFICIENT =        0.6;

# Synthesis: Utils:
        USE_HARMONIC_MODIFICATION =     false;
        HP_FILTER_F0 =                  false;
        FILTER_UPDATE_INTERVAL_VT =     0.3;
        FILTER_UPDATE_INTERVAL_GL =     0.05;
        WRITE_FFT_SPECTRA =             true;
        WRITE_EXCITATION_TO_WAV =       false;

# Synthesis: Voice adaptation:
        PITCH =                         1.0;
        SPEED =                         1.0;
```

```
                JITTER =                        0.0;
                ADAPT_TO_PULSELIB =             false;
                ADAPT_COEFF =                   1.0;
                USE_PULSELIB_LSF =              false;
                NOISE_ROBUST_SPEECH =           false;

        # Synthesis: Pulse library PCA/ICA:
                USE_PULSELIB_PCA =              false;
                PCA_ORDER =                     12;
                PCA_ORDER_SYNTHESIS =           12;
                PCA_SPECTRAL_MATCHING =         true;
                PCA_PULSE_LENGTH =              400;
```

C Test Details

In this appendix the tables used to build the graphics of Section 9 are shown.

C.1 Male Dependent Model Tables

	STRAIGHT MOS	GlottHMM MOS	STRAIGHT ES	GlottHMM ES
Happiness	3.98	3.1	3.95	3.29
Anger	3.89	2.85	3.87	2.66
Neutral	3.83	3.66	3.66	3.78
Surprise	4.07	3.02	3.82	3.47
Sadness	3.94	3.19	3.28	3.33

Table C1: Male MOS and ES

	PREF-ST	PREF-Gl	NAT-ST	NAT-Gl	SIM-ST	SIM-Gl
Happiness	87%	13%	77%	23%	73%	27%
Anger	93%	7%	78%	22%	62%	38%
Neutral	44%	56%	51%	49%	50%	50%
Surprise	75%	25%	81%	19%	75%	25%
Sadness	53%	47%	73%	27%	71%	29%

Table C2: Male Preference, Naturalness and Similarity

C.2 Male Average Model Tables

	STRAIGHT MOS	GlottHMM MOS	STRAIGHT ES	GlottHMM ES
Happiness	3.77	3.05	3.57	2.8
Anger	3.75	3.02	3.57	2.55
Neutral	3.56	3.16	3.56	3.38
Surprise	3.8	2.89	3.56	2.65
Sadness	3.69	3.04	3.53	2.8

Table C3: Average Male MOS and ES

	PREF-ST	PREF-Gl	NAT-ST	NAT-Gl	SIM-ST	SIM-Gl
Happiness	84%	16%	82%	18%	59%	41%
Anger	93%	7%	80%	20%	64%	36%
Neutral	56%	44%	56%	44%	73%	27%
Surprise	89%	11%	78%	22%	69%	31%
Sadness	75%	25%	80%	20%	53%	47%

Table C4: Average Male Preference, Naturalness and Similarity

C.3 Female Dependent Model Tables

	STRAIGHT MOS	GlottHMM MOS	STRAIGHT ES	GlottHMM ES
Happiness	4.17	2.78	4.01	2.89
Anger	3.79	3.04	3.71	3.19
Neutral	4.10	2.95	3.71	3.42
Surprise	3.95	2.82	3.95	3.08
Sadness	3.40	2.18	3.64	2.99

Table C5: Female MOS and ES

	PREF-ST	PREF-Gl	NAT-ST	NAT-Gl	SIM-ST	SIM-Gl
Happiness	93%	7%	88%	12%	80%	20%
Anger	77%	23%	74%	26%	65%	35%
Neutral	68%	32%	82%	18%	68%	32%
Surprise	83%	17%	83%	17%	86%	14%
Sadness	79%	21%	77%	23%	77%	23%

Table C6: Female Preference, Naturalness and Similarity

C.4 Female Average Model Tables

	STRAIGHT MOS	GlottHMM MOS	STRAIGHT ES	GlottHMM ES
Happiness	3.89	2.8	4.09	2.41
Anger	3.55	2.84	3.37	2.82
Neutral	3.61	2.89	4.02	3.33
Surprise	3.96	2.69	3.58	2.64
Sadness	3.09	2.64	3.69	2.8

Table C7: Average Female MOS and ES

	PREF-ST	PREF-Gl	NAT-ST	NAT-Gl	SIM-ST	SIM-Gl
Happiness	94%	6%	85%	15%	69%	31%
Anger	96%	4%	71%	29%	67%	33%
Neutral	50%	50%	63%	37%	80%	20%
Surprise	96%	4%	89%	11%	75%	25%
Sadness	78%	22%	73%	27%	56%	44%

Table C8: Average Female Preference, Naturalness and Similarity

D Boxplots

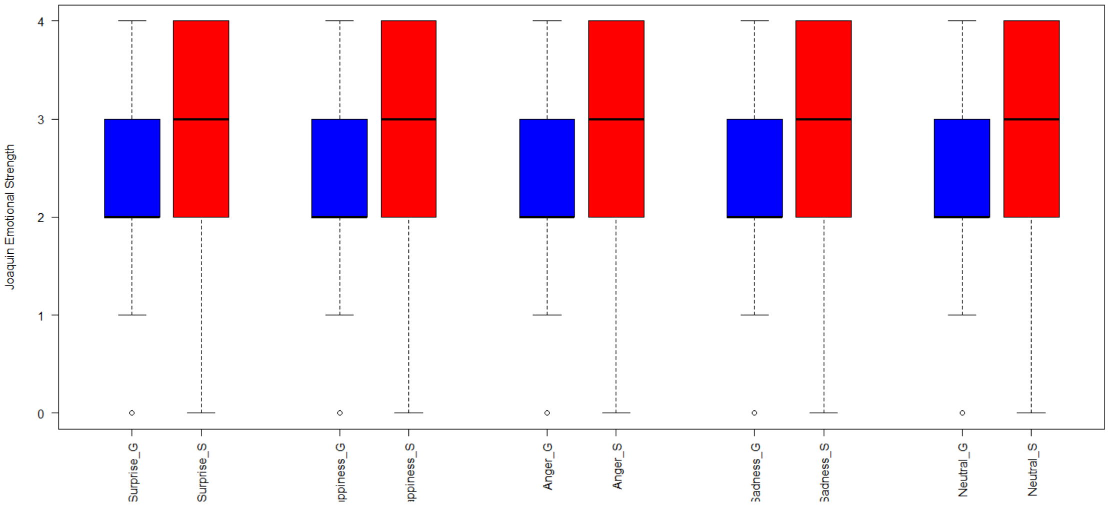

Figure D1: Boxplot Representation for male ES

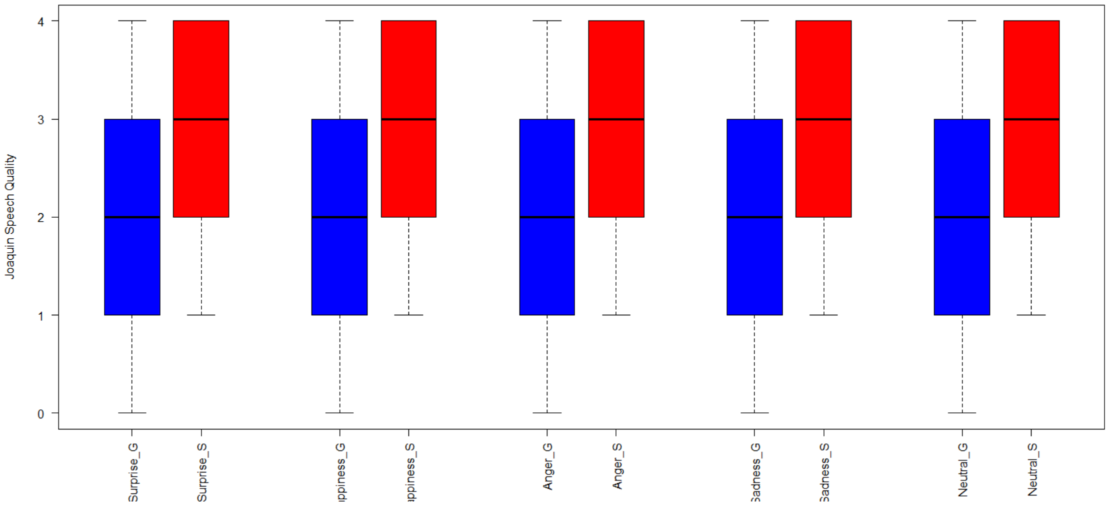

Figure D2: Boxplot Representation for male MOS

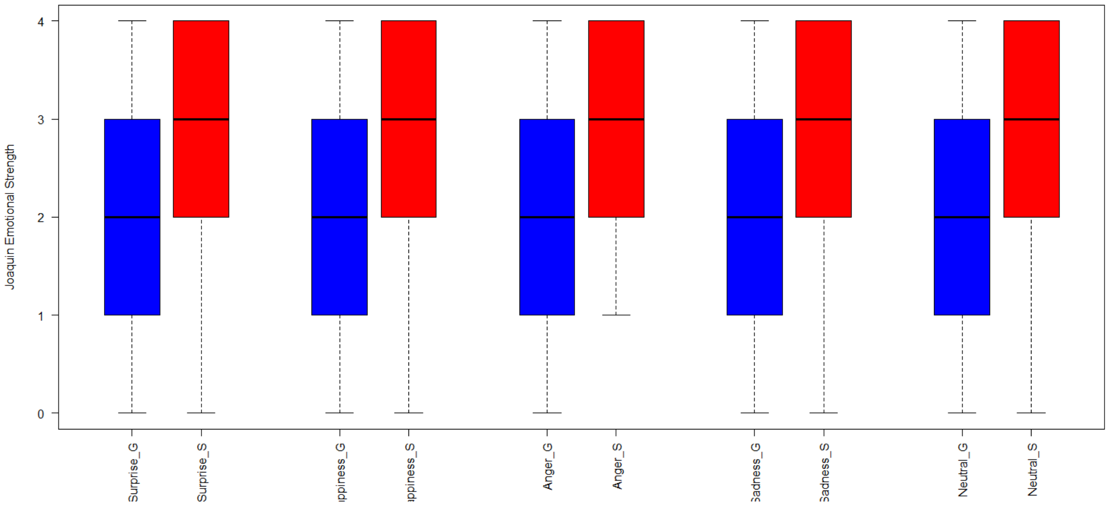

Figure D3: Boxplot Representation for average male ES

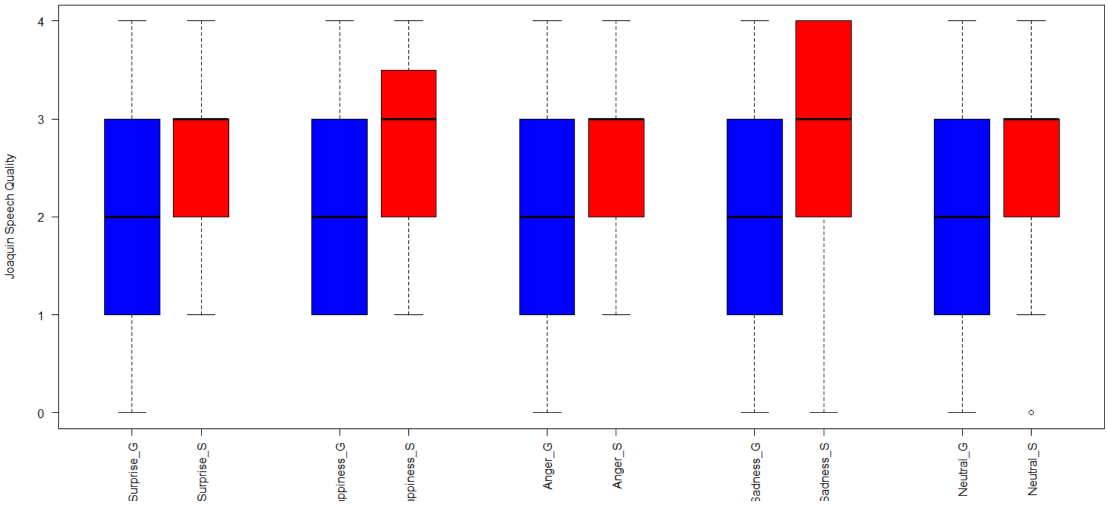

Figure D4: Boxplot Representation for average male MOS

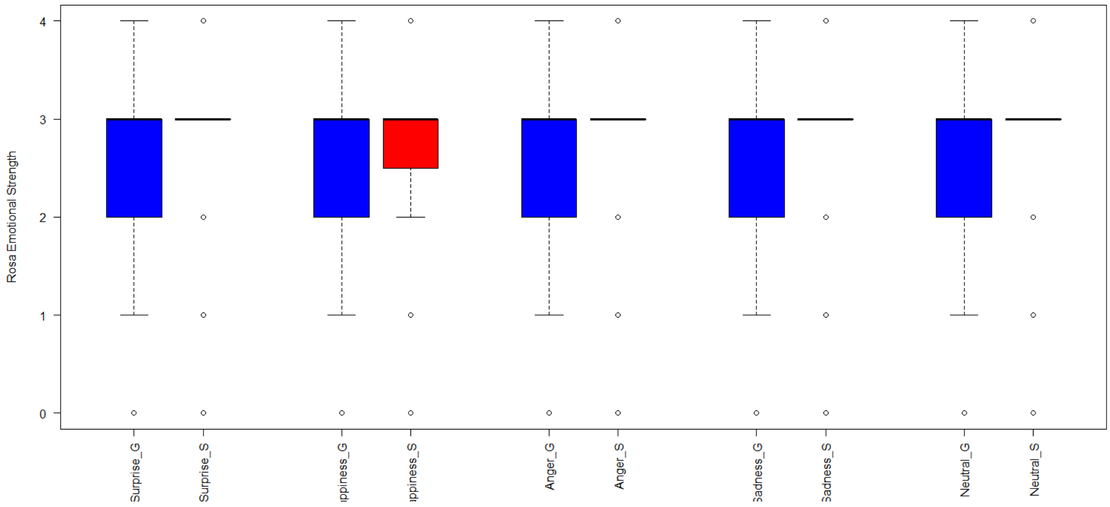

Figure D5: Boxplot Representation for female ES

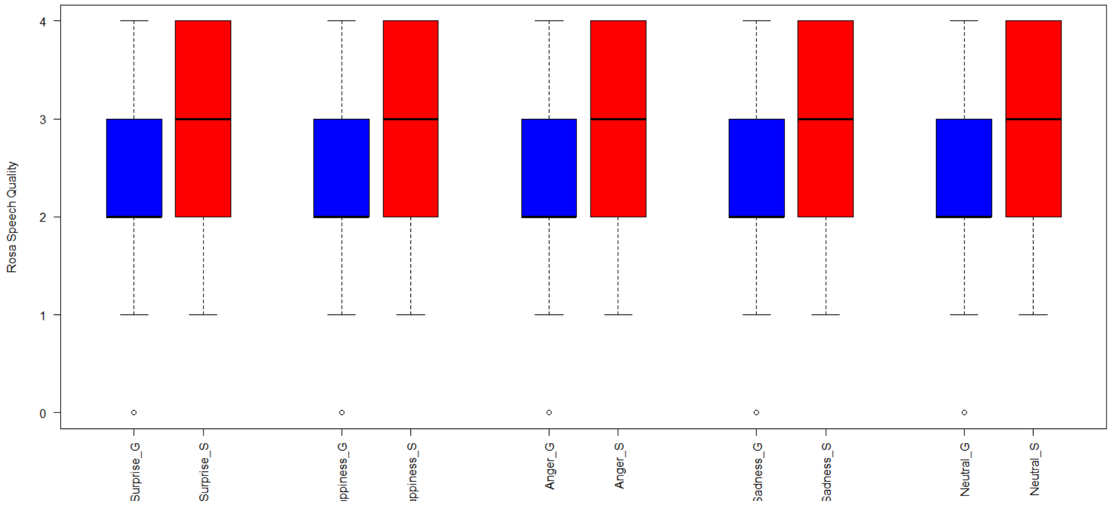

Figure D6: Boxplot Representation for female MOS

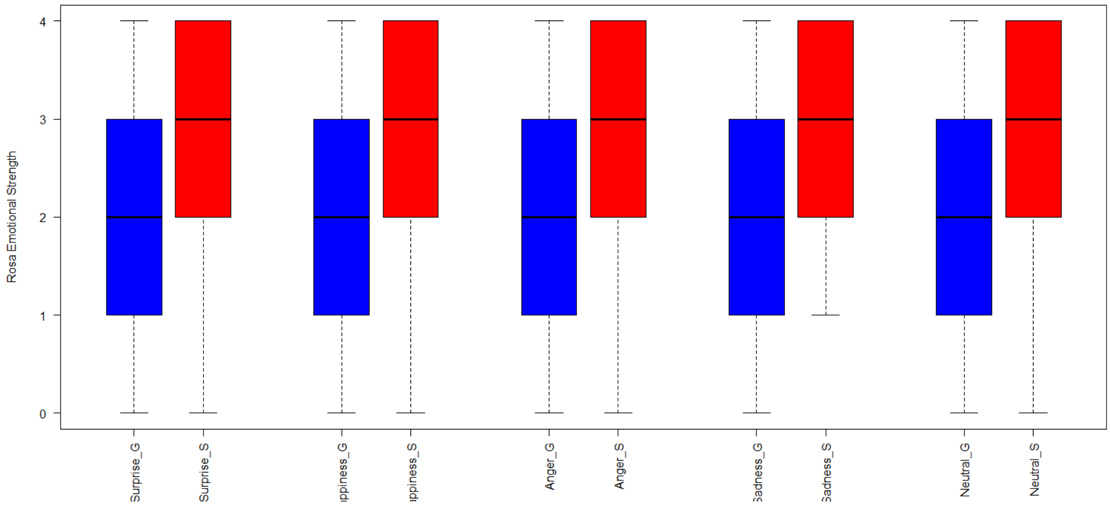

Figure D7: Boxplot Representation for average female ES

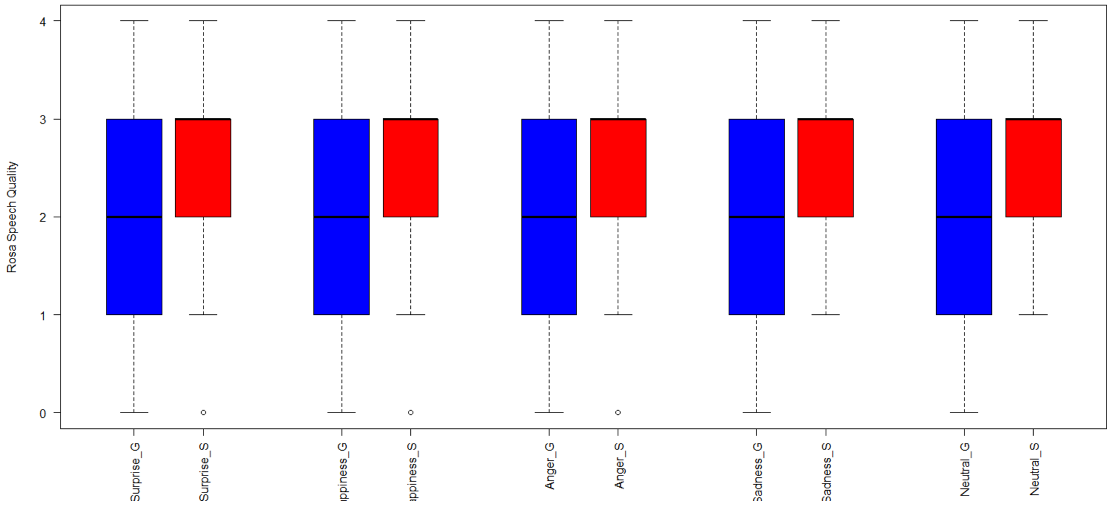

Figure D8: Boxplot Representation for average female MOS