TAMPERE UNIVERSITY OF TECHNOLOGY

SAMUEL NAVARRO LOU

AUTOMATIC CONVERSION OF EMOTIONS IN SPEECH WITHIN A SPEAKER INDEPENDENT FRAMEWORK

Master of Science Thesis

Examiners: D.Sc. Jani Nurminen
M.Sc. Hanna Silen
Prof. Moncef Gabbouj
Examiners and topic approved by the Council of the Faculty of Computing and Electrical Engineering on 04.12.2013

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY
SAMUEL NAVARRO LOU: Automatic conversion of emotions in speech within a speaker independent framework
Master of Science Thesis, 72 pages, 6 Appendix pages.
March 2014
Major: Signal Processing.
Examiners: D.Sc. Jani Nurminen, M.Sc. Hanna Silen, Prof. Moncef Gabbouj.
Keywords: voice conversion, emotion conversion, expressive speech, prosody modeling, speech processing, regression.

Emotions in speech are a fundamental part of a natural dialog. In everyday life, vocal interaction with people often implies emotions as an intrinsic part of the conversation to a greater or lesser extent. Thus, the inclusion of emotions in human-machine dialog systems is crucial to achieve an acceptable degree of naturalness in the communication. This thesis focuses on *automatic emotion conversion of speech*, a technique whose aim is to transform an utterance produced in neutral style to a certain emotion state in a speaker independent context.

Conversion of emotions represents a challenge in the sense that emotions affect significantly all the parts of the human vocal production system, and in the conversion process all these factors must be taken into account carefully. The techniques used in the literature are based on *voice conversion* approaches, with minor modifications to create the sensation of emotion. In this thesis, the idea of voice conversion systems is used as well, but the usual regression process is divided in a two-step procedure that provides additional speaker normalization to remove the intrinsic speaker dependency of this kind of systems, using *vocal tract length normalization* as a pre-processing technique. In addition, a new method to convert the duration trend of the utterance and the intonation contour is proposed, taking into account the contextual information.

Preface

This work has been conducted at the Department of Signal Processing of Tampere University of Technology.

I would like to express my gratitude towards my instructors, Jani Nurminen and Hanna Silen, for their invaluable guidance and continuous eagerness to help, which made the development of the thesis a pleasant and easygoing process. I would also like to thank Prof. Moncef Gabbouj for the opportunity of working in his group. I extend my gratefulness as well to Alfonso Ortega, my supervisor in Zaragoza, whose support during the whole process is really appreciated.

Additionally, I would like to show my thankfulness to all the members of the Audio Research Team for their kind and warm attitude. Additional thanks are given to Toni Heittola for his help with the listening test and statistical testing methods, and to Tuomas Virtanen for welcoming me in the team and helping me finding my outstanding instructors.

On a personal level, I want to deeply thank my labmate Gerard Sanchez, who was working with me in the early stages of this work, for his continuous help, support and advice throughout these six months. On the other hand, I would also like to thank all the wonderful people I met during my exchange period in Tampere, because they made my stay here a delightful and enjoyable time. Besides, I would like to express my appreciation to my family and my friends in Spain, for their support and assistance, specially to my parents, who have been there for me in the good and in the bad moments.

Tampere, 10.03.2014

Samuel Navarro

Table of contents

Terms and definitions

AR	Auto-Regressive
ANN	Artificial Neural Network
BAP	Band Aperiodicities
CART	Classification And Regression Tree
DKPLS	Dynamic Kernel Partial Least Squares
DTW	Dynamic Time Warping
F_0	Fundamental Frequency
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
KPLS	Kernel Partial Least Squares
LP	Linear Predicton
LPC	Linear Predictive Coding
LSF	Line Spectral Frequencies
MCC	Mel-Cepstral Coefficients
MCD	Mel-Cepstral Distortion
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLSA	Mel-Log Scale Approximation Filter
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
PCA	Principal Component Analysis
PDF	Probability Density Function
PLS	Partial Least Squares
SPTK	Speech Signal Processing Toolkit

STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum (vocoder)
VAD	Voice Activity Detector
VC	Voice Conversion
VTLN	Vocal Tract Length Normalization

1. Introduction

Automatic generation of expressive speech in multiple emotional styles is growing in popularity as the human-machine dialog systems advance continuously, and their aim is to achieve higher and higher levels of naturalness. The reason for this is the multiple potential applications that are arising, such as in the leisure industry, in which the creation of avatars with properly defined artificial intelligence is becoming more and more important. Additionally, there are several other applications in the industry to create dialog systems that provoke empathy in its clients, which is likely to increase the happiness of the clients with the company, and therefore raise the potential benefit.

Generation of emotional speech still remains as a challenging area in the field of speech processing, although some satisfactory results have already been obtained [Hof04]. A wide range of techniques can be used to achieve this goal, from all the *text to speech* classical methods to *voice conversion*. The first type of methods generate the synthesized speech from scratch, while the latter parts from an existing utterance and uses regression techniques to transform the parameters and create the sensation that the utterance was produced with a determined emotion.

When speaking about text to speech synthesis applications, the classical techniques used to synthesize the output speech require extensive databases that capture every possible phenomenon present in the speech generation process. Let us imagine a situation in which it is necessary to build a text to speech system that is able to change the emotion of the speech depending on the content of the text or the current interaction with the final user. This application would require enormous databases for each of the emotions that the system had to emulate with the additional constraint of being produced by the same speaker, which is absolutely nonviable when the number of emotions is relatively large. At this point, voice conversion applications show a big advantage, because this kind of techniques are meant to model speaking styles using regression methods to mimic the identity of a certain target speaker by learning a transformation function from another source speaker using a small amount of data. In practice, emotions in speech can be regarded as utterances produced by another source speaker, because of the changes that they cause in the voice production system.

1.1 Emotions in conversion systems: Problem definition

In the emotional speech generation field, voice conversion techniques represent a big improvement compared to other methods since it is possible to model emotions with a small amount of data, which is advantageous when there is a relatively big amount of emotions to generate, as for example in the MPEG-4 standard [Ost02], which uses six emotion categories in face animation processes, which would require six additional big databases to generate the different possible emotional speech states if these techniques were not used. However, most voice conversion techniques for emotional speech generation suffer from an additional drawback, which makes them still not perfectly suitable for all the possible applications, which is that voice conversion techniques require a fixed source and target speaker.

The speaker dependency requirement is actually not a problem in the common voice conversion techniques, since their goal is to transform the identity of the source speaker into that of the target speaker. However, in the emotion conversion process the goal is to mimic an emotion, and to have such a strong restriction in the possible input speaker makes the system still not as adaptive as would be desired, since the system would still require to build additional databases (although smaller) for the speaker whose emotion is desired to be transformed.

On the other hand, the modeling of prosody (i.e. the patterns of stress and intonation in speech) is a very important issue in an emotion conversion system, because as it is commonly known, there is a wide variety of emotions that are expressed mainly via prosody, such as sadness or boredom. This fact is commonly not taken into account in the classical voice conversion systems, due to the fact that prosody is not an extremely critical factor in identity perception, but if emotions are present, prosody begins to play an important role. The prosody characteristics that have to be specially taken into account are the speaking rate and the local pitch variations. This way, the least complex model should at least be able to scale the mean duration of the utterance and generate new pitch contours according to the emotion. However, a desirable model should also be able to model the intra sentence variations of the duration and the pitch, which is a problem that can be addressed via regression methods that either take into account explicit contextual information, as in this thesis, or model contextual information implicitly, which can be represented for example using wavelets [San14].

1.2 Objectives and main results

The main goal of this work is to build a functional emotion conversion system, with the additional objective of removing speaker dependency from the voice conversion system. The approach used in this thesis is based on a two step voice conversion. First, the input identity is normalized using a rough parametric transformation function, whose aim is to reduce the acoustic distance between a certain *reference speaker* and the input speaker, not needing the identity transformation to be subjectively perfect. The second step exploits the fact that the transformed speech is close to the reference speaker, and applies a speaker dependent model to transform the emotion, which has been trained for the reference speaker. Finally, the identity transformation is reversed and the remaining speech has the identity of the input speaker with the desired emotion.

This thesis also presents a transformation system to create emotional style intonation patterns in prosody based on direct transformation of the input speech parameters. The proposed method takes into account lexical information to produce the conversion result, resulting in a context sensitive system that evolves with time.

The performance of the system was evaluated using objective and subjective criteria, to test the degradation due to the inclusion of a two step voice conversion in the first case, and to check the perceived quality of the converted emotions in the second case. The results show that the subjective performance of the system is in general significantly close to authentic emotions, although some emotions show better result than others, which is discussed in Chapter 6. The objective results show that the conversion system effectively reduces the original distortion, sometimes performing almost as well as a speaker dependent system.

1.3 Organization of the thesis

This thesis is organized as follows. Chapter 2 briefly introduces speech production system and common parameterizations. The effect of emotions in the speech production system is presented in Chapter 3, as well as the types of emotional speech databases that can be found in the literature. Chapter 4 introduces voice conversion techniques and their usual architecture, along with a description of how the methods have to be adapted in order to model emotions. Chapter 5 presents the methods used in this thesis to build a complete emotion conversion system. The techniques and the data used to evaluate the performance of this system is presented in Chapter 6, together with a brief discussion about the results. Finally, Chapter 7 exposes the conclusions extracted from this thesis and the possible lines of future work and investigation to extend and improve the current system.

2. Speech Representations

An insight of how speech is mathematically represented is essential for speech transformation and synthesis. In this chapter, the basis of human speech production, and its digital mathematical modeling is presented.

2.1 Speech production model

Speech is the main form of human communication, consisting in a sequence of sounds that create a symbolic representation of the information. The set of sounds is unique of each language, but the production mechanism is common to all them, which allows to develop a universal model for the speech production.

2.1.1 The voice production apparatus

Speech is generated by the human vocal production system, which is illustrated in the figure 2.1. The sound is the result of air-pressure waves that are originated in the lungs, and afterward are filtered by the vocal tract, which produces the different *phonemes* that conform the language.

The speech production apparatus can be divided into the lungs, trachea, larynx (organ of voice production), pharyngeal cavity (throat), oral and nasal cavity. The pharyngeal and oral cavities are typically referred to as the vocal tract, and the nasal cavity as the nasal tract [Hua01].

The different sounds that the human vocal system can produce are divided into two essential categories according to the excitation mode: *voiced* and *unvoiced*. Voiced sounds are produced by the pressure wave going through the glottis with the vocal cords in a tense status, which causes the cords to vibrate, acting as a relaxation oscillator, and its oscillation frequency is called *fundamental frequency* (F_0) or *pitch*. This produces quasi-periodic pulses of air flow which are filtered by the vocal tract. Unvoiced sounds are produced by creating a constriction in some point of the vocal tract, and expelling air through it at a speed that causes a turbulence, which produces noise that excites the vocal tract.

Once the excitation is produced, the vocal tract and the nasal tract act as resonance tubes of non-uniform cross-sectional area. The spectrum of the corresponding sound is therefore shaped according to the frequency selectivity of the vocal tract. The resonant frequencies of the vocal tract are called *formant frequencies* or simply *formants*.

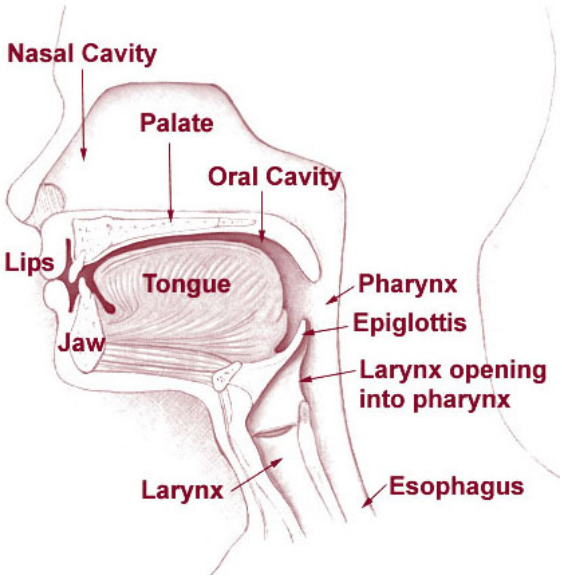


Figure 2.1: Human vocal tract. [Wik14]

The different sounds are generated by varying the shape of the vocal tract, which is translated into movements of the formant frequencies, and speaking style is defined by the movements of F_0. Thus, the speech is completely defined by the excitation signal and the vocal tract filter.

The last part of the human voice production system is the transfer of the generated sound to the free space. This is achieved through the lips, which, due to an acoustic impedance mismatch with the open air, act as a high-pass filter, and attenuates the low frequencies.

2.1.2 The Source-Filter Model

The human vocal tract can be modeled by using theory of acoustic circuits, assuming that the tract can be represented as a concatenation of tubes of non-uniform cross-sectional area, as mentioned in section 2.1.1. Therefore, the complete production model can be represented as an airflow excitation generator, and a series of tubes with time-varying cross-sectional area that shape the frequency content of the excitation signal with a resonant filter. The model assumes that the glottis is a tube of infinite length and the lips are another tube of infinite length as well, as it is shown in the figure 2.2.

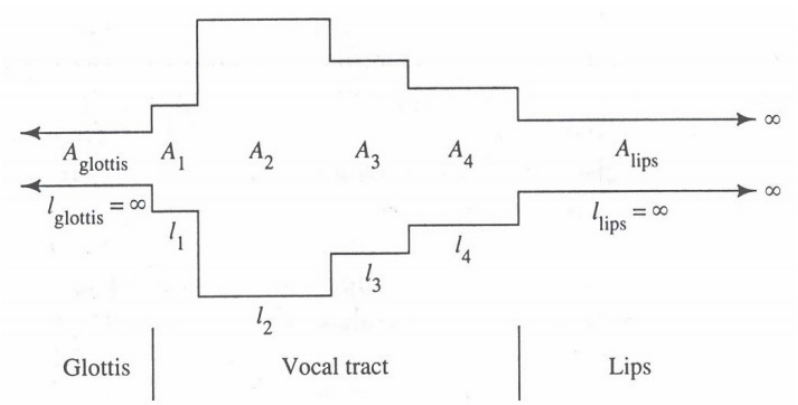


Figure 2.2: Tube model for speech production

The digital model that represents this scheme is called the *source-filter model*, and it consists on the excitation signal ($e(n)$) generator, convolved with a linear filter with impulse response $h(n)$ that represents the vocal tract and a radiation filter with response $r(n)$:

$$y(n) = e(n) * h(n) * r(n) \tag{2.1}$$

In the z-domain this equation can be expressed as:

$$Y(z) = E(z)H(z)R(z) \tag{2.2}$$

The vocal tract filter $H(z)$ can be sufficiently modeled with an all-pole filter, and the radiation filter can be expressed as a simple first order high pass filter:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{P} \alpha_k z^{-k}} \qquad R(z) = 1 - \beta z^{-1} \tag{2.3}$$

where G and $\{\alpha_k\}$ are parameters that depend on the shape of the vocal tract, and β is a radiation parameter that satisfies $\beta < 1$ [Hua01]. The excitation signal $e(n)$ is dependent on the desired excitation type. A periodic train of pulses is required for voiced sounds, and random white noise is needed for the unvoiced sounds. The block diagram of the complete system is shown in the figure 2.3.

2.2 Linear predictive coding

Linear prediction is one of the most powerful tools in speech processing, since it is based in the assumption that a signal $s(n)$ can be expressed as:

$$s(n) = -\sum_{k=1}^{p} \alpha_k s(n-k) + e(n) \tag{2.4}$$

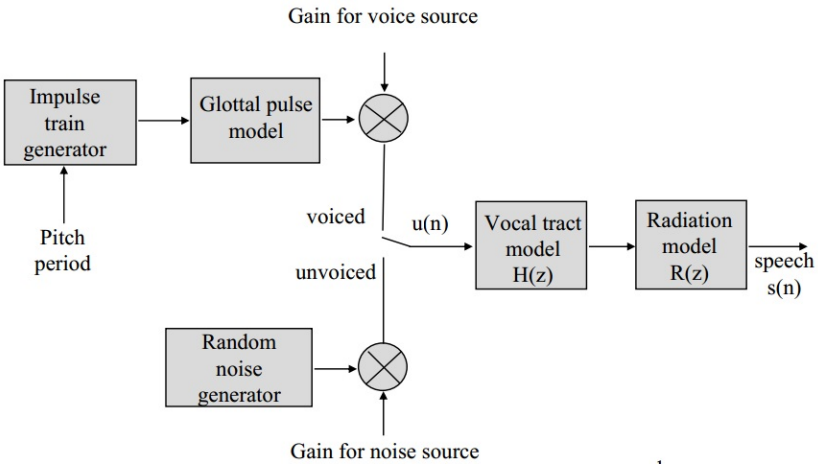


Figure 2.3: Source-filter model block diagram

where the parameters α_k[1] are the result of the linear prediction analysis, and $e(n)$ is the residual error, which is minimized in the *mean squared error* (MSE) sense, and corresponds to the stochastic component of the signal that cannot be predicted by this model.
The equation 2.4 can be written in the z-domain as:

$$S(z)\left(1+\sum_{k=1}^{p}\alpha_k z^{-k}\right) = E(z) \Rightarrow S(z) = \frac{E(z)}{1+\sum_{k=1}^{p}\alpha_k z^{-k}} \tag{2.5}$$

which means that this method is useful to model signals generated by auto-regressive (AR) processes like speech, and therefore the analysis is able to extract significant information about its generation process.

These coefficients $\{\alpha_k\}$ give a compact representation of the spectral envelope of the speech signal, which is suitable for transformation and synthesis, and the coding of speech based on these coefficients is called Linear Predictive Coding (LPC).

2.2.1 Linear prediction analysis

The premise of the LP analysis is, as shown in equation 2.4, that the current sample can be partially predicted as a linear combination of the previous p samples. The block diagram of the system is shown in the figure 2.4.

The prediction filter $P(z)$ can be expressed as $P(z) = -\sum_{k=1}^{p}\alpha_k z^{-k}$, and thus the global filter can be written as $A(z) = 1+\sum_{k=1}^{p}\alpha_k z^{-k}$.

[1] The minus sign in equation 2.4 is a convention and may differ in the literature

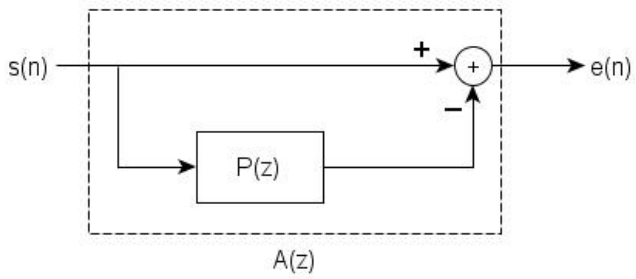


Figure 2.4: Block diagram of the LP analysis

To solve the estimation of the prediction parameters α_k, the residual error is minimized in the MSE sense. The mean squared error is defined as:

$$E\{e(n)^2\} = E\left\{\left(s(n) + \sum_{k=1}^{p} \alpha_k s(n-k)\right)^2\right\} \tag{2.6}$$

The minimization criterion thus requires:

$$\frac{\partial E\{e(n)^2\}}{\partial \alpha_j} = 2E\left\{\left(s(n) + \sum_{k=1}^{p} \alpha_k s(n-k)\right) s(n-j)\right\} = 0 \tag{2.7}$$

Which implies:

$$E\{s(n)s(n-j)\} = -\sum_{k=1}^{p} \alpha_k E\{s(n-k)s(n-j)\} \tag{2.8}$$

$$R_s(j) = -\sum_{k=1}^{p} \alpha_k R_s(j-k) \quad j = 1, \ldots, p \tag{2.9}$$

Where $R_s(k)$ represents the autocorrelation function, which can be estimated from the signal $s(n)$. The equation 2.9 leads to the well-known *normal equations*, which can be solved efficiently using the Levinson-Durbin method [Lev47; Dur60]:

$$\begin{bmatrix} R_s(0) & R_s(1) & \ldots & R_s(p-1) \\ R_s(1) & R_s(0) & \ldots & R_s(p-2) \\ \vdots & \vdots & & \vdots \\ R_s(p-1) & R_s(p-2) & \ldots & R_s(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = - \begin{bmatrix} R_s(0) \\ R_s(1) \\ \vdots \\ R_s(p) \end{bmatrix} \tag{2.10}$$

This equation is valid for stationary signals, but as it has been shown, speech is naturally non-stationary since the vocal tract is changing in time, and thus the filter $H(z)$ is time-varying in the model. This problem can be overcame by assuming that speech is locally stationary within the length of a phoneme, and therefore the LPC analysis can be performed in framed segments of speech, applying a window signal $w(n)$.

2.2.2 Line spectral frequencies

Although the LPC coefficients give a compact representation of the spectral envelope of the speech signal, they are not widely used to code or transform speech, since the operations that may be applied to these coefficients can easily make the resulting filter unstable, and that produces a problem in the synthesis process.

To solve this problem, the line spectral frequencies (LSF) are introduced. Let the LP analysis filter be $A(z) = 1 + \sum_{k=1}^{p} \alpha_k z^{-k}$, then the polynomials

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)} A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)} A(z^{-1}) \end{aligned} \tag{2.11}$$

Are respectively a symmetric and anti-symmetric polynomials which satisfy:

$$A(z) = \frac{1}{2} \left(P(z) + Q(z) \right) \tag{2.12}$$

And have the following properties [Soo84]:

- All zeros of $P(z)$ and $Q(z)$ are in the unit circle (i.e, the modulus equals 1).
- The complex argument of the zeros of $P(z)$ and $Q(z)$ are interlaced with each other.
- Zeros of $P(z)$ and $Q(z)$ do not overlap.

An inspection of these properties show that the LP filter $A(z)$ is totally represented by the zeros of $P(z)$ and $Q(z)$. Furthermore, as the zeros are located on the unit circle, they can be located uniquely by their phase, named Line Spectral Frequency, which allows to store them as a real number rather than a complex number. Additionally, the LSFs are positioned symmetrically respect to the real axis, removing the need of storing the negative LSFs, and therefore the LP filter is completely represented by p real numbers.

The advantage that LSF representation shows compared with the LPC coefficients is that it ensures the stability of the reconstructed filter although the values are altered, which is a desirable property in speech coding and transformation.

2.3 Cepstrum

As with LPC, the cepstral analysis is a major method to extract and represent the spectral envelope of the speech signal. The cepstral analysis is a homomorphic transformation, as it transforms the convolution of discrete-time signals into a sum of discrete-time signals as shown in equation 2.13.

$$s(n) = x(n) * y(n) \xrightarrow{\mathcal{C}} \hat{s}(n) = \hat{x}(n) + \hat{y}(n) \tag{2.13}$$

The *complex cepstrum* of a signal $s(n)$ is defined as the inverse Fourier transform of the complex logarithm of the Fourier transform of the signal [Opp65]:

$$\hat{s}(n) = \mathcal{C}\{s(n)\} = \mathcal{F}^{-1}\left\{\log(\mathcal{F}\{s(n)\})\right\} = \mathcal{F}^{-1}\{\log(|S(\omega)|)\} + \mathcal{F}^{-1}\{j\arg(S(\omega))\} \tag{2.14}$$

From this definition can also be deduced that if the signal $s(n)$ is real, then the complex cepstrum is real as well. The time index n is usually called *quefrency*, in opposition to the word *frequency*.

The homomorphic properties can be easily shown, by inputting a convolved $s(n) = x(n) * y(n)$ signal into the analysis system:

$$\begin{gathered} S(\omega) = \mathcal{F}\{s(n)\} = \mathcal{F}\{x(n) * y(n)\} = X(\omega)Y(\omega) \\ \log(S(\omega)) = \log(X(\omega)) + \log(Y(\omega)) \\ \hat{s}(n) = \mathcal{F}^{-1}\{\log(S(\omega))\} = \mathcal{F}^{-1}\{\log(X(\omega))\} + \mathcal{F}^{-1}\{\log(Y(\omega))\} = \hat{x}(n) + \hat{y}(n) \end{gathered} \tag{2.15}$$

However, the cepstrum is frequently defined as $\mathcal{F}^{-1}\left\{\log|\mathcal{F}\{s(n)\}|\right\}$ instead, and then it is called *real cepstrum* $c_s(n)$, or simply *cepstrum* by some authors [Roa96; Pro07], which can be easily shown to be the even part of the complex cepstrum:

$$c_s(n) = \frac{\hat{s}(n) + \hat{s}(-n)}{2} \tag{2.16}$$

This is easier to compute as there is no need to compute the complex logarithm. Furthermore, the complex cepstrum of a minimum phase signal can be shown to be null for $n < 0$ (annex A.1), and therefore the complex cepstrum is completely defined by the real cepstrum:

$$\hat{s}(n) = \begin{cases} 0 & n < 0 \\ c_s(0) & n = 0 \\ 2c_s(n) & n > 0 \end{cases} \tag{2.17}$$

Using the cepstral analysis it is possible to separate the vocal tract filter from the excitation signal, due to the fact that the vocal tract filter is a minimum phase all-pole filter, which by observation of the equation A.6, it can be deduced that the energy is essentialy concentrated in low quefrencies, whereas the excitation signal has the energy concentrated on harmonics of the *fundamental period*, as shown in equation A.11 in annex A.2. The process of extracting the cepstral representation of the filter is called *liftering*, in contrast to the word *filtering*.

Figure 2.5 illustrates the process of cepstral analysis applied to one vowel. Figure 2.5a shows the spectral envelope obtained by liftering, and figure 2.5b depicts the real cepstrum of the analyzed signal.

According to the source-filter model described in section 2.1.2, the speech signal can be represented by the vocal tract filter and the fundamental fequency F_0. Using cepstral analysis, the vocal tract filter envelope can be represented using a small number of coefficients from the cepstrum, and F_0 can be also determined from the position of the cepstral peak.

2.4 Mel-cepstral representation

Although cepstral modeling can represent the spectral envelope of speech modeling poles and zeros with equal weights, the number of coefficients needed to adequately represent the spectral envelope of the signal is relatively high compared to the LPC representation.

However, the human psycho-acoustic system does not perceive every frequency with equal resolution. Two well known scales that model the human ear frequency response are the *Mel* and *Bark* scales. The Mel scale is widely used for speech feature extraction in many applications, and it is defined as:

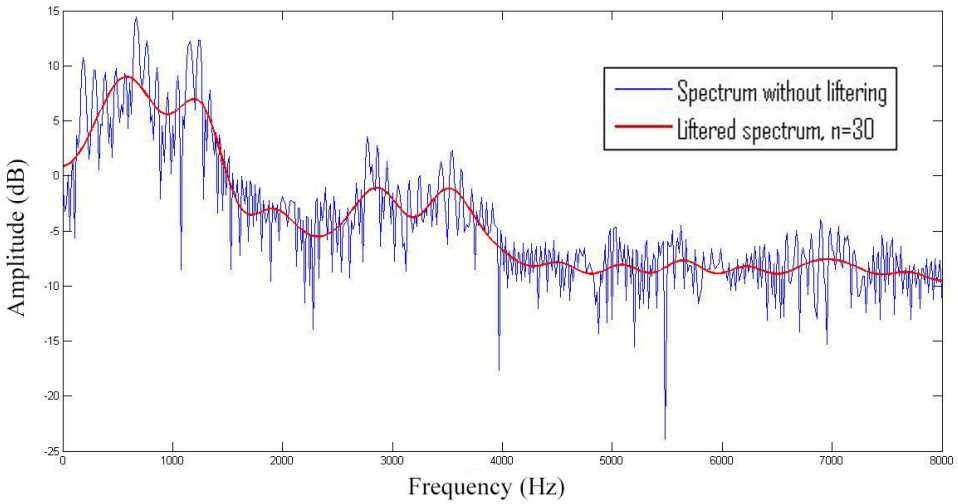
$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.18}$$

Mel-cepstral coefficients (MCC) are frequency warped coefficients [Ima83a] that have been popular in voice transformation tasks [Tod07; Hel12b]. The spectrum is modeled using Dth order MCCs $c_\alpha(k)$, $k = 0, 1, \ldots, D$:

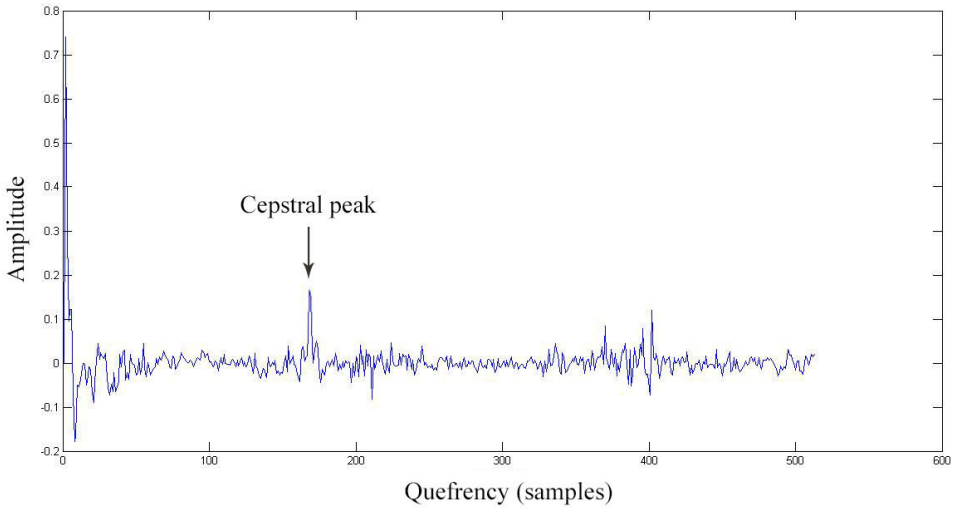
$$H(z) = \exp\left(\sum_{k=0}^{D} c_\alpha(k)\tilde{z}^{-k}\right) \tag{2.19}$$

where $\tilde{z}^{-1}$ is defined via the bilinear function:

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{2.20}$$



(a) Spectrum of the vowel /a/ and its liftered envelope



(b) Real cepstrum of the signal without liftering

Figure 2.5: Cepstral anlysis of the spanish vowel /a/

An appropiate selection of α can approximate the mel scale. In the case of 16 kHz sampling, the usual choice is $\alpha = 0.42$.

The vocal tract filter can be reconstructed by means of the *Mel-Log Scale Approximation Filter* (MLSA), which implements the reconstruction in equation 2.19, as described in [Ima83b].

3. Emotions in speech

Emotions are complex phenomena, and the way they affect the speech production is still an unclear issue. However, several authors have tried to develop the basis of a consistent theory to support the vocal expression of emotions, as in [Sch03; Cor00]. The results presented in this chapter illustrate how emotions are present in the speech signal and that the evaluation methods for emotional speech synthesis are necessarily biased due to the relatively low recognition rate in emotion identification.

To understand how emotions affect speech, the study in [Ban96] is quite representative. It analyzes the acoustic characteristics of recorded emotional speech, including spectral and prosodic features. It was found that emotions affect the energy distribution of the spectrum, and changes occur in the formant positions. Additionally, a linear regression model was fitted to the prosodic features, using as independent variables all those that may affect the value of the parameter: the speaker identity and gender, the linguistic content of the sentence, the environment, and of course the emotion. In the result it is shown that the emotion explains 55% of the variation of the mean intensity, and 50% of the variation of mean F_0. There are several studies conducted by phoneticians and psychologists that confirm the result, a compilation of the most important findings in these studies is presented in [Eri05].

3.1 Emotional speech databases

In order to analyze the effect of emotions in speech, a database of emotional speech is needed. The obtaining of a proper emotional speech data is still a big challenge. The ideal is to gather a closely controlled dataset with spontaneous speech. This is impossible to achieve, as it is not possible to get spontaneous speech under controlled conditions, thus, researchers have developed several techniques to obtain somewhat natural and spontaneous voice.

3.1.1 Acted emotions

The simplest technique consists on acted speech uttered by professional actors pretending a certain emotion. This kind of recordings can be made in a conditioned room, where the recording conditions and the text content are under complete control. On the other hand, it does not provide full naturalness in the selected emotions, generally due to overacting. Because the speaker does not have any other means than his own voice to express the emotion, there is some tendency to overacting so as to get the emotion clearly portrayed [Bat00].

This method is generally used to record databases oriented to developing text-to-speech or voice transformation systems, where the quality of the recorded speech generally plays an important role. Furthermore, because of the overacting, the emotions in these databases are very recognizable, which is adequate to develop a speech synthesizer system of emotional speech, where the important fact is that the listener identifies the emotion.

It is also a widely used technique to analyze the emotional speech signal comparing it with other emotions. Because the method allows a total control over the read text, it is possible to record the same text in every emotion.

3.1.2 Stimulated emotions

This method consists on having emotionally rich carrier sentences read by non professional actors, which intends to evoke genuine emotions. It is expected that the highly emotive text provokes the emotions to arise naturally without the need of overacting. Using this approach the utterances can still be recorded under strictly controlled environmental conditions, but the possibility of having parallel sentences vanishes, which make the databases less suitable for voice transformation, where parallel utterances are a big benefit.

3.1.3 Real emotions

The databases with real emotions are usually obtained from TV programs with high emotional content (such as debates), from interviews, or from Wizard of Oz[1] experiments. The emotions portrayed in these databases are completely natural, but the control over the environment and the content is completely lost. This kind of database is suitable for the testing of automatic emotion recognition systems, where the generalization capability is important.

[1] A Wizard of Oz experiment is a research experiment in which subjects interact with a computer system that subjects believe to be autonomous, but which is actually being operated by a hidden human being

3.2 Emotion perception

Although there is a clear evidence that emotions affect speaking style, in absence of extra clues, the identification of the portrayed emotion is not straightforward even for humans. In [Sch01] it is reported that vocal expression of affect may be motivated in part by universal psychobiological mechanisms, and in part by the segmental and supra-segmental aspects of the particular language. In this work, they gather results from a study conducted in nine countries in Europe, the United States, and Asia on vocal emotion portrayals of anger, sadness, fear, joy, and neutral voice as produced by professional German actors. Data showed an overall accuracy of **66%** across all emotions and countries. Nevertheless, there were differences ranging from **74%** in Germany to **52%** in Indonesia. However, patterns of confusion were very similar across all countries. These data suggest the existence of similar inference rules from vocal expression across cultures, although the highest recognition rate comes from the native speakers of the expressive speech language.

The results of this study shows that humans are well able to infer the emotional status of an unknown speaker, but the recognition rate shows that there is still confusion between emotions when speaking. For instance, joy is the emotion with the lowest recognition rate with only a **48%** of recognition rate among the German listeners, while a neutral speaking style is the best recognized with a **88%** recognition rate. However, there is a clear overestimation of neutral style, being confused up to **34%** of the times when the emotion was joy.

4. Voice Conversion

In this chapter, firstly, a review of the most common voice conversion (VC) techniques is shown and the generic blocks of a VC system are described, emphasizing on the state of the art mapping techniques. Secondly, an insight of how emotions are handled in VC systems is given, describing the necessary components to perform the task successfully.

The term voice conversion refers to the the area of speech processing that deals with the transformation of the speech uttered by one speaker (*source speaker*) to create the impression that it was uttered by another speaker (*target speaker*). The conventional VC systems consist of two steps: Training and conversion. In the training phase, a transformation function is created based on the speech data from both speakers to map the source speaker features to the target speaker features. In conversion, the mapping function is applied to any unknown utterance from the source speaker to make it sound like it was spoken by the target speaker. The goal of a VC system is to achieve the highest quality possible with a relatively small training dataset. The most common scheme of this *stand-alone voice conversion* is depicted in the block diagram of the figure 4.1.

The great majority of existing VC systems are focused in the spectral features modification, leaving the prosodic characteristics in the background or performing simple modifications of these features.

4.1 Parallel dataset alignment: Dynamic Time Warping

The same sentences uttered in different moments rarely have the same speaking rate. The utterances must therefore be aligned in time to estimate the transformation function frame by frame.

The most popular approach for alignment is *Dynamic Time Warping* (DTW), which was introduced in [Sak78] for word recognition. The aim of the DTW algorithm is to find an optimal time correspondence between two feature sequences $\mathcal{A} = \{a_1, \ldots, a_N\}$ and $\mathcal{B} = \{b_1, \ldots, b_M\}$ in terms of a distance function. The algorithm assumes that the first and the last vectors are aligned, and that sequences do not go back in time. A usual choice is to use euclidean distance together with MFCCs for speech recognition or MCCs for speech transformation, which give a reasonably measure of the acoustic distance [Nur12]. For the purpose of this thesis, MCCs were the features to be aligned.

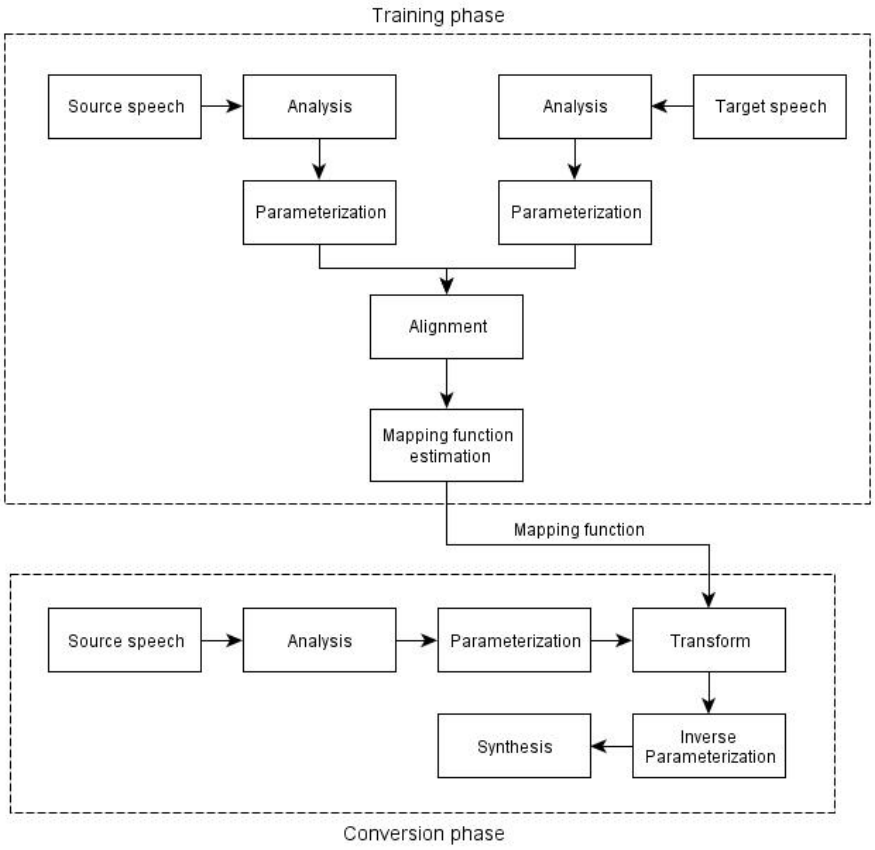


Figure 4.1: Voice conversion block diagram.

First, a *local cost matrix* $D = (d_{ij})$ is built, where d_{ij} contains the distance between the feature vector i from the sequence $\mathcal{A}$ and the vector j from the sequence $\mathcal{B}$:

$$D \in \mathbb{R}_{NxM} : \; d_{ij} = ||a_i - b_j|| \qquad i \in [1:N], \; j \in [1:M] \tag{4.1}$$

Once the local cost matrix is built, the algorithm finds the optimal alignment path, which runs through the minimum cumulative distance path, which can be calculated by defining a cost function for every point in the grid. The most common cost function is defined as follows:

$$\phi(i,j) = d_{ij} + \min[\phi(i-1,j-1), \phi(i-1,j), \phi(i,j-1)] \tag{4.2}$$

where $\phi(i,j)$ is the cumulative distance at point (i,j). The minimum cumulative distance is $\phi(N,M)$, and the optimal path is found by backtracking from the point (N,M) to the point $(1,1)$.

4.2 GMM-based mapping

A *Gaussian Mixture Model* (GMM) is a statistical distribution constituted by a weighted sum of M simple gaussian distributions, so that the density function of this distribution is:

$$f_x(x) = \sum_{m=1}^{M} \omega_m \mathcal{N}(x; \mu_m; \Sigma_m) \tag{4.3}$$

where ω_m represents the prior probability of the mth gaussian, and $\mathcal{N}(x; \mu_m; \Sigma_m)$ denotes the $D-$dimensional multivariate normal distribution with mean μ_i and covariance matrix Σ_i:

$$\mathcal{N}(x; \mu_m; \Sigma_m) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_m|}} \exp\left[-\frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)\right] \tag{4.4}$$

GMM based mapping functions are based on the assumption that the probability density function (PDF) of both source and target features is a GMM distribution. The assumption is reasonable since these distributions with enough number of gaussians can approximate any probability distribution [Fel66].

4.2.1 Source GMM

The GMM approach was first proposed in [Sty98]. In this approach a GMM was built for the source feature vectors, and the mapping function is assumed to be linear for each gaussian in the form:

$$\hat{y}_t = F(x_t) = \sum_{m=1}^{M} P(m|x_t)(A_m x_t + b_m) \tag{4.5}$$

Where $\hat{y}_t$ is the estimated taget feature, the regression matrix A_m and the bias term b_m are to be estimated with *minimum mean squared error* (MMSE) criterion, and the posterior probability of the mth gaussian can be calculated using the Bayes theorem:

$$P(m|x_t) = \frac{\omega_m \mathcal{N}(x_t; \mu_m; \Sigma_m)}{\sum_{m=1}^{M} \omega_m \mathcal{N}(x_t; \mu_m; \Sigma_m)} \tag{4.6}$$

4.2.2 Joint density GMM

The required matrix to obtain the MMSE solution for the source GMM approach is very large and sometimes poorly conditioned, so in [Kai98] it was proposed to build a joint GMM for the source vectors and the target vectors. The source features are augmented with their corresponding target features as $z_t = \left[x_t^T, y_t^T\right]^T$, and $\{z_t\}_{t=1}^N$ modeled with a GMM, such that:

$$f_z(z) = \sum_{m=1}^{M} \omega_m \mathcal{N}(z; \mu_m^{(z)}; \Sigma_m^{(z)}) \tag{4.7}$$

where

$$\mu_m^{(z)} = \begin{bmatrix} \mu_m^{(x)} \\ \mu_m^{(y)} \end{bmatrix} \qquad \Sigma_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix} \tag{4.8}$$

Under this assumption, x_t and y_t are jointly multivariate gaussian for each gaussian in the GMM, and the conditional distribution probabilities are therefore a GMM as well [Mar79]. The conversion function with MMSE criterion is the expected value of the conditional distribution of y_t given x_t, which is:

$$\hat{y}_t = E\{y_t|x_t\} = \sum_{m=1}^{M} P(m|x_t)\overline{\mu}_m(x_t) = \sum_{m=1}^{M} P(m|x_t)\left[\mu_m^{(y)} + \Sigma_m^{(yx)}{\Sigma_m^{(xx)}}^{-1}(x_t - \mu_m^{(x)})\right] \tag{4.9}$$

4.2.3 Problems with the GMM-based mapping

These GMM based techniques suffer from several disadvantages inherited from the machine learning models on which they are based. When determining the complexity of the model, there is a trade-off between the generalization capability that it has and the fidelity to the training data.

The most common problems found in GMM approaches are the following:

- Oversmoothing: Simple models tend to remove the fine details of the spectrum and to broaden the formants.
- Overfitting: Models that are too complex may be overfitted to the training data and the generalization capability is therefore low.
- Time-independency: The temporal correlation of the converted features is ignored in these models.

In order to solve these issues, several solutions have been proposed, such as in [Hua01], where post-filtering is introduced to improve oversmoothed spectra. In [Tod07], a maximum-likelihood (ML) estimation of the parameter trajectory is proposed, which partially solves the time independency of the mapping. Additionally, this work proposes using the global variance of the target features to palliate the effect of oversmoothing, which is simpler than the post-filtering method. In [Hel12a], PLS regression is proposed as an alternative to the MMSE solution to reduce the overfitting of the models.

4.3 Dynamic Kernel Partial Least Squares Regression

Nonlinear mapping can be accomplished using a broad range of techniques, such as artificial neural networks (ANN) [Nar95; Des10], or support vector regression [Son11].

A major drawback of these nonlinear regression systems is the extensive parameter tuning they require. To overcome this problem, in [Hel12b] a new technique called *dynamic kernel partial least squares* (DKPLS) is introduced.

The KPLS method was first introduced in [Ros01], and has the advantage of being able to model nonlinear relations between variables by using a kernel transformation as a preprocessing step. In the DKPLS technique, the dynamics are modeled by augmenting the regressors of every frame with the kernel data from consecutive frames. The regression is performed using PLS criterion, which is able to handle the multicollinearity generated by the use of kernels and the dynamic modeling.

The training scheme for the DKPLS system is depicted in the figure 4.2. After the alignment step, as explained in section 4.1, the *k-means* clustering algorithm is applied on the source data to find a set of C reference vectors, and then the feature vectors are non linearly mapped to a higher dimension space via the kernel transformation, as explained in section 4.3.1, and the transformed vectors are centered, as the regression model requires the data to be zero-mean. The temporal continuity is taken into account by concatenating the kernel transformed vectors of consecutive frames. Finally, the conversion parameters are found using the PLS method [Jon93].

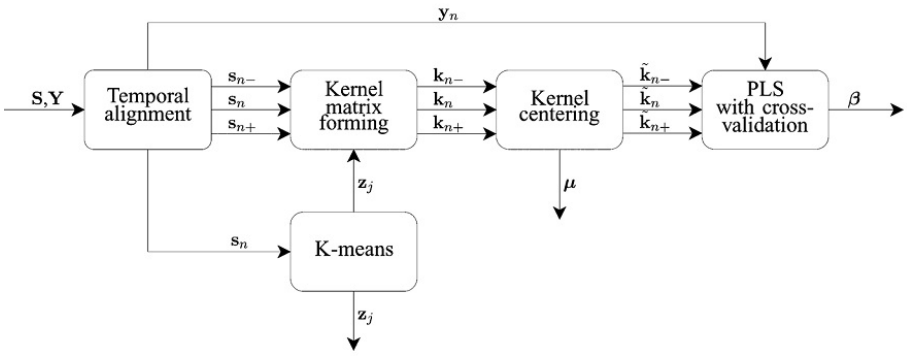


Figure 4.2: DKPLS block diagram for training [Hel12b]

4.3.1 Kernel transformation

The kernel transformation is performed using a Gaussian radial basis function kernel as the transformation function. The transformation calculates the similarity between each source vector $\mathbf{s}_n,\ n = 1, \ldots, N$, and the reference vectors obtained from the k-means clustering $\mathbf{z}_j,\ j = 1, \ldots, C$. The Gaussian kernel is defined as

$$k_{jn} = e^{\frac{-\left\|\mathbf{s}_n - \mathbf{z}_j\right\|^2}{2\sigma^2}} \tag{4.10}$$

where σ is the width parameter of the kernel. The selection of σ is not highly crucial, usually it is enough to find a decent range for it.

The transformed vectors are stored into a matrix $\mathbf{K}$ which has the following form:

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \ldots & k_{1N} \\ k_{21} & k_{22} & \ldots & k_{2N} \\ \vdots & \vdots & & \vdots \\ k_{C1} & k_{C2} & \ldots & k_{CN} \end{bmatrix} \tag{4.11}$$

To force the bias term of the conversion model to zero, kernel centering is required. Centering in the kernel space is not as obvious as in the original feature space, since the mean cannot be computed directly. For the kernel matrix $\mathbf{K}$, the following steps are applied [Ben03]:

1. Calculate the average of each row in the kernel matrix and subtract the values from $\mathbf{K}$. The averages of the rows are stored into vector μ.

2. Calculate the average of each column and subtract them from $\mathbf{K}$ resulting from Step 1. The resulting column and row-wise centered kernel matrix is denoted as $\tilde{\mathbf{K}}$.

In the conversion process, an analogous procedure is followed, with the exception that the vector which is subtracted from the columns of $\mathbf{K}$ is the μ vector obtained in training.

4.3.2 Dynamic Modeling

In order to take into account the time continuity of the speech features, the DKPLS algorithm relies on augmenting the kernel transformed vectors with its respective previous and following vectors, obtaining a vector x_n as:

$$x_n = \begin{bmatrix} \tilde{\mathbf{k}}_{\mathbf{n}-} \\ \tilde{\mathbf{k}}_{\mathbf{n}} \\ \tilde{\mathbf{k}}_{\mathbf{n}+} \end{bmatrix} \tag{4.12}$$

where $\tilde{\mathbf{k}}_{\mathbf{n}}$ denotes the centered kernel vector for feature vector $\mathbf{s_n}$, and $\tilde{\mathbf{k}}_{\mathbf{n}-}$ and $\tilde{\mathbf{k}}_{\mathbf{n}+}$ the centered kernel vectors for the preceding and following frames of $\mathbf{s_n}$, respectively.

The notation $n+$ and $n-$ is introduced due to the fact that in the training process, the frames are aligned using DTW, which alters the natural ordering of the sequences. However, in conversion the order is unchanged and every frame is processed, and thus the preceding and following frames are the frames $n-1$ and $n+1$ respectively.

4.3.3 Kernel Partial Least Squares

After the kernel transformation, a linear regression model $\mathbf{y}_n = \beta \mathbf{x}_n + \mathbf{e}_n$ is applied, where $\mathbf{y}_n$ denote the unprocessed target vector, $\mathbf{x}_n$ represents the augmented kernel transformed vector of the source feature vector (as in equation 4.12), and $\mathbf{e}_n$ represents the regression residual. The entries of $\mathbf{x}_n$ are highly linearly dependent on each other, due to the use of kernels that increases the dimensionality, and to the concatenation of the consecutive transformed vectors.

PLS is a linear regression method that models relationships between a regressor matrix $\mathbf{X}$ and a response matrix $\mathbf{Y}$ which is different from the standard multivariate regression (based on MMSE) in a way that it can cope with collinear data and cases where the number of observations is lower than the number of explanatory variables.

PLS works similarly to principal component analysis (PCA), but in PCA, the principal components are determined by only $\mathbf{X}$ whereas in PLS, both $\mathbf{X}$ and $\mathbf{Y}$ are used for extracting new regressor variables. The aim of PLS is to extract components that capture most of the information in the $\mathbf{X}$ variables that is also useful for predicting $\mathbf{Y}$.

To perform the regression task, PLS constructs new explanatory variables, called *latent variables* or *components*, where each component is a linear combination of $\mathbf{x}_n$, then standard regression methods are used to determine the latent variables in $\mathbf{y}_n$. The number of latent components has to be set beforehand, and the optimal number can be estimated using *cross-validation*.

There exist many algorithms for the PLS regression problem. In this thesis, the SIMPLS algorithm proposed by [Jon93] is used to find the regression matrix β. The SIMPLS algorithm is computationally efficient and avoids calculating matrix inverses.

4.4 Prosody modeling

Prosody refers to the supra-segmental structure of intonation, energy and speaking rate. Prosody structure comes partially defined by the semantic and linguistic content of the sentence, perceptible in accents, intonational patterns, or prosodic pauses to clarify the message (which in written language are translated into commas or full stops). However, prosody is also used frequently to transmit nonverbal information.

Although most of the VC literature is focused on the spectral transformation, it has been shown that prosodic features provide important cues of the speaker identity [Hel07a], and are also very important when emotions are involved [Ban96]. Regarding prosody transformation, literature focuses on F_0 transformation and duration modification.

The most common transformation for prosody is the *mean-variance* scaling, which is based on a simple linear scaling for each sample:

$$F_{0t}(n) = \mu_t + \frac{\sigma_t}{\sigma_s}\left(F_{0s}(n) - \mu_s\right)$$

This modeling is equivalent to assuming that F_0 is a white Gaussian process with mean μ_s and variance ${\sigma_s}^2$, and the transformation is a simple PDF equalization (see annex A.3). This approach is obviously imprecise, but for identity conversion it works reasonably well, since most of the information is in the spectrum, and prosody plays an active role mostly when the speaker is known to the listener [Hel07a].

However, if an accurate description is desired, the above mentioned method does not work appropriately. This may be the case of transforming to a special speaking style, or the aim of mimicking an emotion, which is the purpose of this thesis.

There are several more elaborate F_0 transformation methods, which use more sophisticated regression techniques [Ina03]. Most of these methods ignore the temporal correlation of the F_0 samples, and they perform the regression sample-wise, with more complex functions (i.e. an Nth order polynomial, or a GMM function like in section 4.2).

Nevertheless, there are approaches that try to model F_0 on a higher level, such as the codebook regression, which builds the final F_0 contour by concatenating segments selected from a codebook database [Ina03; Hel07b]. A drawback of this method is that it requires an extensive codebook in order to build an adequate output F_0 contour, which conflicts with the principle of VC of using a small training dataset.

On the other hand, there are approaches which express F_0 contour as a result of a mathematical model [Xu01; Fuj05; Kor03]. These methods are specially popular in Chinese speech manipulation, where F_0 is restricted to the basic 5 tones of this language for each syllable [Kan06].

Recently, a F_0 model based on wavelet decomposition has been proposed in [Sun13] for speech synthesis applications, and its application to VC systems is discussed in [San14]. The study in [Sun13] suggest that the wavelet decomposition of F_0, if calculated appropriately, can give information about the different phonetic units in each wavelet level.

4.5 Emotions in voice conversion systems

When the goal is to mimic an emotion, a conventional VC system is not enough. Emotions exhibit complex prosodic patterns and affect the voice production system. In order to deal with this issues, the emotion applied VC systems usually incorporate additional elements that try to to handle all these contingencies.

It is commonly believed that emotions are essentially a prosodic phenomenon, but according to [Bar07], emotions can be classified in the range of *mainly prosodic* to *mainly segmental*. This means that certain emotions are identified using almost uniquely the speaking style or prosody, such as surprise or sadness, some others are recognized using the information in the spectrum, such as cold anger, and finally there are emotions that show complex identification patterns, such as joy or happiness. Furthermore, emotions affect the voice quality as well, making it creakier or muffled in some cases. Thus, to capture these effects, the conversion system must be able to take into account every element of the voice production chain.

The problem of expressive speech generation from neutral speech has been addressed by several authors with similar results. Most works usually have a spectral model for timbre transformation and a prosodic and duration models, which mimic the prosody of the emotion.

4.5.1 Spectral transformation

It has been shown that emotions affect the vocal tract, due to the tension or relaxation associated with certain emotions, and thus an adequate spectral transformation model is needed. Emotions that involve tension in the vocal tract, such as anger, show a tendency to increase energy distribution in high frequencies, and the formant positions are moved, which creates an effect of "narrow voice" [Sch03]. On the other hand, emotions that get the voice production apparatus relaxed result in a raise of low frequency energy and broadening of formant bandwidths. In [Sal10], an analysis of the variation of energy distributions due to emotions is performed, and the results show that the mean filter that redistributes the energy from neutral style to a certain emotion is very similar for each phoneme.

Spectral conversion is a well-known technique in the field of VC. The spectral model can be a standard GMM regression as described in section 4.2, which is used in several works where the focus area is the prosodic model [Ina07; Cen10]. However, emotions are phenomena where the dynamic information plays an important role, and some authors focus their work in developing a more elaborate regression method to capture these dynamics, such as in [Wu06], where the target and source features are modeled by left-to-right HMMs, and the transformation function is a dependent on the state of the so called *Bi-HMMs*. Additionally, this work modifies the state duration of the HMMs so that it follows a gamma distribution.

4.5.2 Prosody transformation

It is clear that the classic mean-variance scaling for F_0 is inadequate for the purpose of emotion transformation. Thus, a proper model should take into account contextual information that affect the speaking style. The prosodic model usually deals with F_0 transformation and duration modification, and relies on the spectral model for energy transformation.

A common approach is to generate a whole new F_0 using similar techniques to those that speech synthesizers use to generate spectral samples, using for instance context-dependent HMMs [Ina07]. Nevertheless, this method requires a reasonably big training dataset, which counterposes the main goal of VC.

Parametric descriptions of F_0 are also used to transform prosody [Kan06], as this way the transformation function only has to deal with a small set of parameters, and can be mapped using any kind of known regression methods, such as GMM regression.

The duration of phonemes or syllables is an important element to take into account, as gives important clues of the portrayed emotion [Ban96]. Duration can be modified by simply scaling the mean duration of the syllables, which performs reasonably good [Cen10], or it can be modified either phoneme-wise or syllable-wise [Ina07], which gives a more accurate results in very prosodic emotions, such as boredom. In this sense, a regression method to predict the durations is needed, such as CART regression, which additionally allows the use of categorical predictors, useful to represent linguistic information.

5. Methodology

This chapter discusses the implementation details of a VC system applied to emotion conversion. First, an overview of the system is presented. Secondly, the building blocks of the conversion scheme are described in detail.

The proposed system is based on the classical VC scheme, using additional emotion-specific subsystems whose purpose is to improve the perceived sensation of the portrayed emotion, as shown in section 4.5, and with the additional goal of removing the speaker dependency in the conversion.

5.1 System architecture

The conversion method has to take into account every element that emotions affect in speech, and thus a good description of speech is needed in order to perform the conversion. The features used must be descriptive and easily convertible, and they must allow a complete resynthesis of speech after the conversion.

For this purpose, the STRAIGHT [Kaw97] high quality vocoder has been chosen to extract the speech features and synthesize back the converted speech. The vocoder provides the spectral envelope of the signal, which is transformed to $24th$ order MCC representation using the SPTK implementation [SPT]. Additionally, the vocoder provides an estimation of the fundamental frequency F_0, that is used to convert the intonation, and also information about the excitation signal in the form of *aperiodicities*, that explain the degree of voicing in speech as a function of frequency. The extracted features are then converted using the system depicted in figure 5.1.

Prosody conversion is achieved through a CART regression. The CART provides scales for the mean F_0 and the duration in each syllable. The transformed F_0 is obtained by means of sample-wise scaling using a soft contour created through interpolation of the scaled syllable-wise means, and subsequently the syllable segments of F_0 are resampled to achieve the desired duration.

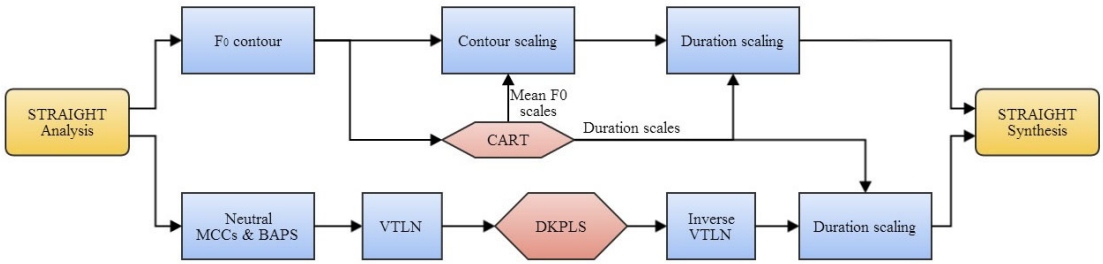


Figure 5.1: Block diagram of the emotion conversion system

Spectral conversion is performed in a two step transformation. First, a normalization technique called *vocal tract length normalization* (VTLN) is applied, which acts as a parametric "pre-voice conversion" step, which is intended to reduce the acoustic distance between the spectral parameters of the input speaker and the parameters of a certain reference speaker chosen from the training database, whose features were used to train the emotion transformation function of the second step. Secondly, a speaker dependent emotion spectral conversion model is applied, which is based on DKPLS regression, and captures the conversion function in detail as it has been trained with both neutral and emotional data from the reference. After the emotion model is applied, the VTLN effect is reversed, and the converted spectrum is resampled to match the duration of the F_0 converted segments.

Speaker adaption can nevertheless be achieved via other methods, such as "average voices" training, which is based on using several speakers in the training process. The reason why the system is designed using VTLN based adaption is because it is able to normalize any input features without prior knowledge of them, as the process is on-line. This is advantageous due to the reduced amount of training data it requires in contrast to other techniques, since there is usually a large number of emotions to model, which increases the number of required training utterances dramatically. For example, in the case of six emotions plus neutral style and 30 utterances per emotion for training, the minimum amount of utterances would be just 210 for VTLN based adaption, and $210N$ for average speakers training, where N is the number of training speakers, usually large.

5.2 Feature extraction

The selection of the features is a crucial issue in a VC system, and specially in a system where the aim is to transform emotions, since the description of the speech signal should also describe the emotional status. Thus, the analysis of the speech signal must return a compact representation of the vocal tract filter and the excitation signal. For this purpose, the STRAIGHT[Kaw97] high quality analysis-synthesis framework is used to analyze the speech signal. The parameters returned by STRAIGHT are then converted to a compact representation, as shown in chapter 2.

5.2.1 STRAIGHT Framework

STRAIGHT is the most established of the commonly used vocoding methods. It was originally proposed by Kawahara in [Kaw97], and it has been under continuous research and development.

STRAIGHT is a *multi-band mixed excitation vocoder*, which means that it uses additional parameters along with the F_0 value to generate the excitation signal, instead of a regular periodic train of impulses, in order to reduce the "robotic" effect that they create. STRAIGHT was originally designed as a tool for speech transformation and accurate spectral envelope representation. Original STRAIGHT parameters are represented as Fourier transform magnitudes and aperiodicity measurements corresponding to them.

First, STRAIGHT uses a F_0 extraction algorithm called TEMPO (Time-domain Excitation extractor using Minimum Perturbation Operator). The TEMPO algorithm is based on the following almost harmonic representation of speech:

$$x(t) = \sum_{k=1}^{N} a_k(t) cos\left(\int_0^t (k\omega_0(\tau) + \omega_k(\tau))\, d\tau + \phi_k\right)$$

where $a_k(t)$ represents a slowly time-varying instantaneous amplitude, $\omega_0(\tau)$ is the instantaneous frequency (to estimate), and $\omega_k(\tau)$ is a slowly varying FM component of the kth harmonic.

The fundamental frequency is then extracted by means of a continuous wavelet transform with a Gabor mother wavelet, and calculating a measure called *fundamentalness* [Kaw99], defined as:

$$M(t,\tau_c) = -\log\left[\int_{\Omega_c}\left(\frac{d|D(t,u)|}{du}\right)^2 du\right] + \log\left[\int_{\Omega_c}|D(t,u)|^2 du\right] - \log\left[\int_{\Omega_c}\left(\frac{d^2\arg(D(t,u))}{du^2}\right)^2 du\right] + 2\log(\tau_c)$$

where $D(t,\tau_0)$ is the wavelet transform of the speech signal, and Ω_c is an integration interval proportional to the size of the analyzing wavelet at scale τ_c. Extracting F_0 is performed by simply finding the maximum in terms of the scale τ_0. The instantaneous frequency is then calculated for each channel τ_c as:

$$\omega_0(t,\tau_c) = \frac{d(\arg(D(t,\tau_c))}{dt}$$

And selecting the fundamental frequency is reduced to finding the maximum of $M(t,\tau_c)$ in terms of τ_c, and then either choose the instantaneous frequency corresponding to that scale, or interpolate through the scales proportionally to the value of M.

Then, the spectral envelope of the signal is calculated by smoothing the spectrum using F_0 adaptive windows with equivalent temporal and spectral resolution. The signal is windowed using two complementary windows:

$$w(t) = e^{-\pi\left(\frac{t}{t_0}\right)^2} h\left(\frac{t}{t_0}\right)$$
$$w_c(t) = w(t)\sin\left(\pi\frac{t}{t_0}\right)$$

where t_0 is the fundamental period, and $h(t)$ is the 2nd order cardinal B-spline function, defined as:

$$h(t) = \begin{cases} 1-|t| & |t|<1 \\ 0 & \text{otherwise} \end{cases}$$

Smoothing in the frequency domain is achieved through the 2^{nd} order cardinal B-spline function, which robustly interpolates the sampled spectrum (due to the periodicity of the signal). The complimentary window function $\omega_c(t)$ is sinusoidally modulated so that the spectrogram produces maxima there where the original spectrogram has minima [Kaw99].

The final spectrogram is obtained through the combination of the smoothed spectrograms $P_0(\omega, t)$ and $P_c(\omega, t)$ calculated using the window functions:

$$P(\omega, t) = \sqrt{P_0(\omega, t)^2 + \xi P_c(\omega, t)^2}$$

Where ξ is a blending factor that minimizes the temporal variation of the spectrogram.

Additionally, STRAIGHT provides measures of aperiodicity for each frequency. The measures are taken by comparing the ratio of the upper and lower spectral envelopes for each frequency, and taking a value from a look-up-table (LUT). Then, the aperiodicity values are smoothed by averaging throughout the speech spectrum:

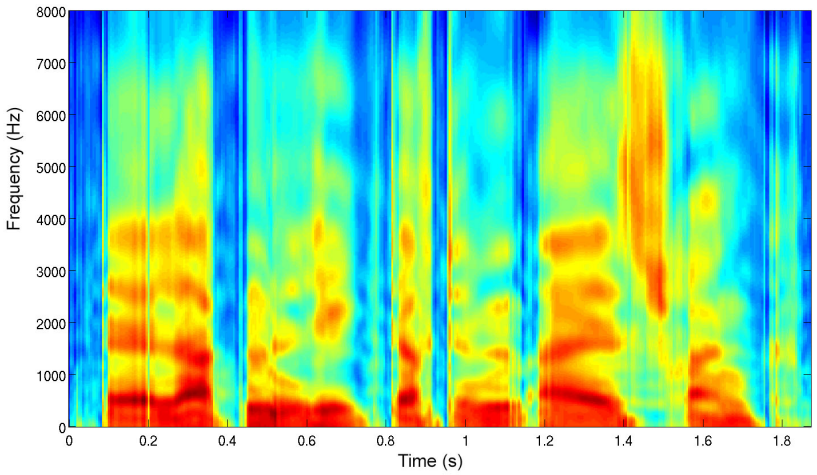
$$P_{AP}(\omega, t) = \frac{\int_{-\infty}^{\infty} w(\lambda, \omega) S(\lambda)^2 \mathrm{LUT}(\lambda) d\lambda}{\int_{-\infty}^{\infty} w(\lambda, \omega) S(\lambda)^2 d\lambda}$$

Where w is a simplified auditory filter centered at frequency ω, and $S(\lambda)^2$ is the speech power spectrum.

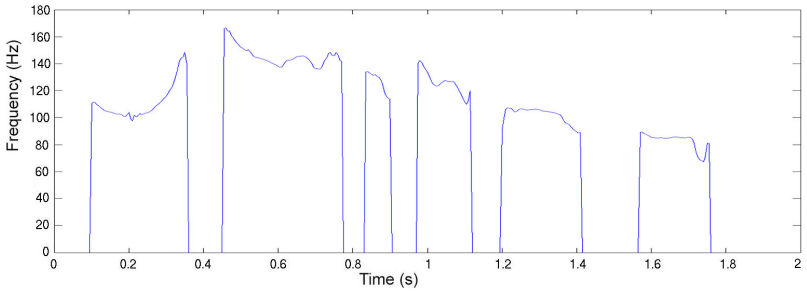
5.2.2 Representation in the system

The STRAIGHT framework provides high quality features, which makes them suitable for speech manipulation. However, the representation that the vocoder gives is very high-dimensional, since it returns an uncompressed estimate of the spectral envelope and aperiodicities for every frequency. The dimensionality of the features makes the conversion process very computationally demanding.

The parameters returned by the vocoder are a power spectral envelope of 513 coefficients, which are the first coefficients of a 1024-dimensional fast Fourier transform, to avoid redundancy; a measure of aperiodicities for each frequency, which therefore has 513 coefficients as well; and an F_0 measure. The parameters are updated every 5 ms and F_0 is computed in the range from 60Hz to 310Hz. An example of the parameters returned by STRAIGHT is shown in figure 5.2.



(a) Spectrogram with the power spectral envelope as a function of time



(b) F_0 contour

Figure 5.2: STRAIGHT analysis of the German sentence *"Der lappen liegt auf dem eisschrank"*

In order to achieve a representation which requires a reasonable computational power, the spectral features are converted to 24th order MCCs, and the aperiodicity measures are compressed to *band aperiodicities* (BAPs).

The BAPs are obtained by averaging the aperiodicity measures in five bands: 0-1kHz, 1-2kHz, 2-4kHz, 4-6kHz and 6-8kHz. This way, only five-dimensional features are used to represent the excitation.

To represent the spectral envelope of the speech signal, MCCs are used, with a frequency warping factor of 0.42 (see section 2.4), which approximates the mel-scale. This selection is motivated by the fact that MCCs are commonly used in VC tasks, and they are a robust method to characterize the amplitude spectrum, which is very close to the way the human auditory system processes audio.

In order to obtain the spectral representation, the SPTK toolkit [SPT] is used to obtain the MCCs. SPTK takes the 513-dimensional power spectrum obtained from STRAIGHT and obtains the warped log-spectrum using a cost function based on the unbiased log-spectrum estimation [Tok94], which is minimized using the Newton-Raphson method.

5.3 Spectral conversion

A major drawback of the previous approaches to emotional speech generation using VC techniques is that they are strongly speaker dependent. The problem is essentially originated in the spectral regression method, in which the data is taken from parallel sentences uttered by the same speaker, which creates a speaker dependency in the model. In this thesis, this problem is addressed by performing a two-step spectral regression. First, an identity conversion is performed with a parametric invertible transformation called *vocal tract length normalization* (VTLN), whose function is to act as a pre-voice conversion step to provide a normalized identity to the following element in the conversion chain. It is important to note that the identity transformation does not create a subjectively perfect sensation of the reference speaker's identity, but reduces the acoustic distance between them. Secondly, the normalized speech is introduced into a speaker dependent DKPLS model (see section 4.3) trained for the reference speaker, which is expected to handle all the possible complex effects caused by dynamics and nonlinearities originated by the presence of emotions. Finally, the converted spectrum is vocal-tract denormalized to recover the original speaker identity, and it is resampled in time to fit the durations dictated by the prosody model (see section 5.4.2).

5.3.1 Vocal Tract Length Normalization

VTLN is a well known technique in the field of speech processing. It has been widely used for speaker recognition [Kam95], as it proved to increase significantly the recognition rate, since it reduces the spurious variability that the recognizer has to deal with.

The basic theory of VTLN is based on the underlying idea that resonances in an acoustic tube (such as the vocal tract) are inversely proportional to the length of the tube. Therefore, the formant positions depend on the length of the vocal tract, and the speaker normalization can thus be achieved by means of frequency warping of the spectrum, as shown in figure 5.3.

In sum, VTLN was originally developed as a tool to compensate for the speaker differences due to the differences in vocal tract length by warping the frequency axis of the amplitude spectrum. This purpose is the exact same as that of VC, hence VTLN is a technique that can be used to perform identity transformation. VTLN has already proved to be a useful technique for VC purposes, as shown in [Sun03]. In this thesis this fact is exploited to normalize the identity before the emotion model is applied.

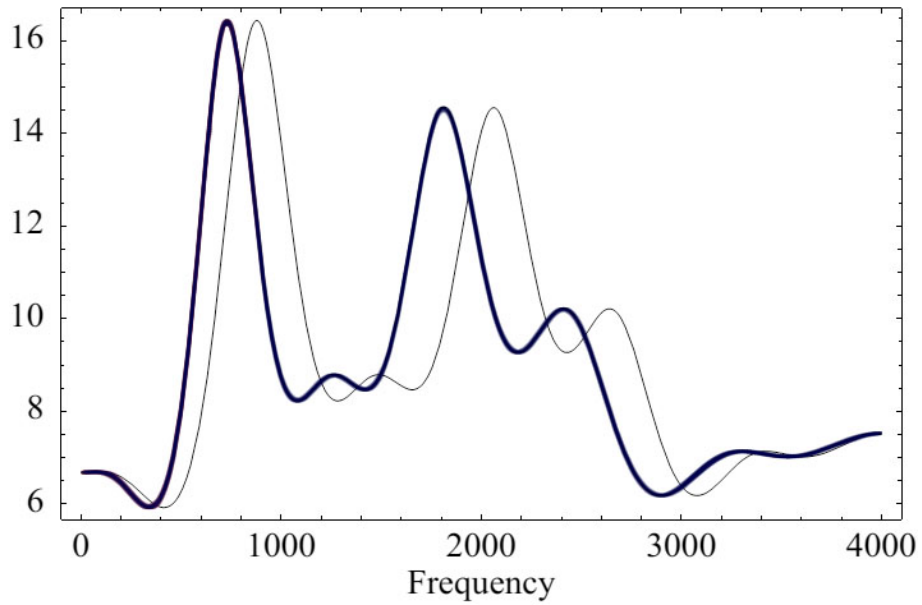


Figure 5.3: Frequency warping of speech spectrum. Warped spectrum is shown with a thick blue line, and original spectrum is shown with a thin black line. [Mcd98]

VTLN is characterized by being an invertible parametric technique, which means that it performs the frequency warping by means of a fixed function that depends on one or more parameters, and whose effect can be subsequently reversed. Most warping functions depend only on a single parameter α, out of which the most popular are following:

- Symmetric piece wise linear warping [Weg96; Wel99]:

$$\tilde{\omega}_\alpha = \begin{cases} \alpha\omega & \omega \leq \omega_0 \\ \alpha\omega_0 + \dfrac{\pi - \alpha\omega_0}{\pi - \alpha\omega_0}(\omega - \omega_0) & \omega \geq \omega_0 \end{cases} \tag{5.1}$$

$$\omega_0 = \begin{cases} \frac{7\pi}{8} & \alpha < 1 \\ \frac{7\pi}{8\alpha} & \alpha \geq 1 \end{cases}$$

- Bilinear function warping [Mcd98]:

$$F(z) = \frac{z - \alpha}{1 - \alpha z} \quad \text{where} \quad \tilde{\omega}_\alpha = F(e^{j\omega}) \tag{5.2}$$

In the context of this thesis, *bilinear warping* VTLN is used, because it can be easily embedded within the process of feature extraction, since the extraction of the MCCs requires the same type of warping. Furthermore, as shown in [Ace93], a cascade of two bilinear transforms with parameters α_M and α_V is equivalent to a single transform with parameter:

$$\alpha_T = \frac{\alpha_M + \alpha_V}{1 + \alpha_M \alpha_V} \tag{5.3}$$

Which besides means that the concatenation of two bilinear transforms is conmutative. Using this result, the extraction of the spectral features followed by the VTLN can be expressed as an only system with one warping parameter, as shown in figure 5.4.

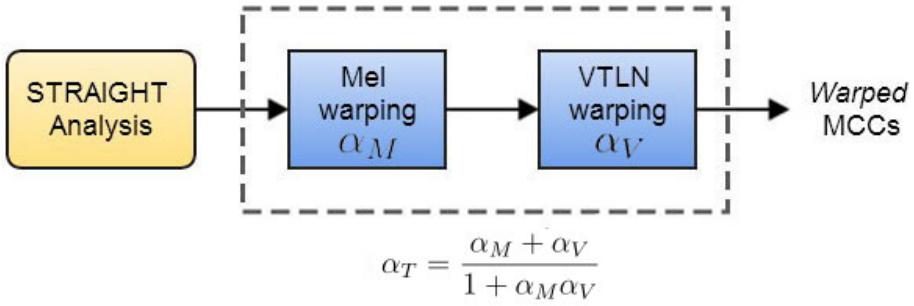


Figure 5.4: Block diagram of the MCC and VTLN systems concatenation equivalence

The parameter selection is therefore developed taking into account the mel warping parameter of the MCC extraction. In order to estimate the optimal overall warping parameter in the conversion process, first a statistical model is built for the distribution of the reference speaker's MCCs, and then the warping parameter can be estimated in the ML sense, such that:

$$\tilde{\alpha} = \arg\max_{\alpha}\{\mathcal{L}(\mathbf{x}_\alpha|\Theta)\} = \arg\max_{\alpha}\left\{\sum_{k=0}^{N}\mathcal{L}(x_{\alpha k}|\Theta)\right\}$$

Where $\mathcal{L}(\mathbf{x}_\alpha|\Theta)$ represents the log-likelihood of the warped sequence resulting from transforming the input sequence $\mathbf{x}$, containing N vectors, using the warping parameter α, given the statistical model Θ.

The statistical distribution of the reference speaker is modeled using a GMM with 256 Gaussian components, as it has been shown to be effective for speaker modeling and identification [Jou13]. The search is bounded for the VTLN warping parameter, which is forced to satisfy $|\alpha_V| < 0.2$ in order to prevent the system from deviating too much from the identity mapping (it is assumed that the variations due to vocal tract differences are subtle). The ML search is finally performed using *Brent's method* [Bre73], a minimization method for one dimensional functions without computing derivatives, which has proved to give good results in the parameter estimation [Mcd98], and preserves the bound restrictions in the search.

Brent's method

The ML criterion requires finding the maximum of the likelihood function, which depends on the the warping parameter α. Instead of searching a local maximum directly, the negative log-likelihood function is defined as a one dimensional function $\tilde{\mathcal{L}} = -\mathcal{L}(\mathbf{x}_\alpha|\Theta)$, and Brent's method is used to find the minimum, using the implementation in [Bur].

Brent's method is a minimization algorithm that combines the bisection method with inverse quadratic interpolation within a bounded interval to find a local minimum of a function f [Bre73]. The method starts with a fixed triplet (a, b, c), such that $f(b) > f(a)$ and $f(b) < f(c)$, and iteratively collapses the bounds of the interval until the final solution is reached. In every iteration, an estimate of the minimum x_0 is obtained with inverse quadratic interpolation or using bisection, and the bounds are collapsed using the following rule:

$$(a_{k+1}, b_{k+1}, c_{k+1}) = \begin{cases} (x_0, b_k, c_k) & \text{if } x_0 < b_k \wedge f(x_0) > f(b_k) \\ (a_k, x_0, c_k) & \text{if } f(x_0) < f(b_k) \\ (a_k, b_k, x_0) & \text{if } x_0 > b_k \wedge f(x_0) > f(b_k) \end{cases}$$

The algorithm tries to use the inverse quadratic interpolation when possible, and if the bounds of the interval are not collapsing rapidly enough, a bisection step is applied. This guarantees that the convergence is at worst linear, but generally superlinear.

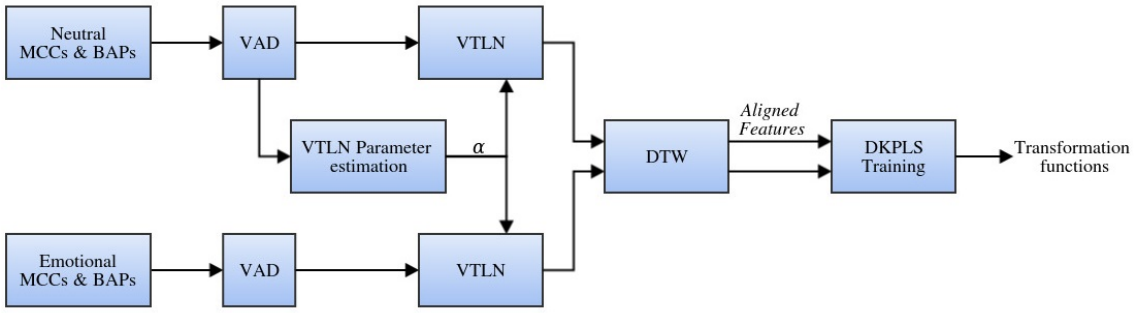


Figure 5.5: Block diagram of the spectral subsystem training

5.3.2 System training

The spectral model subsystem requires examples of both the emotion which is going to be portrayed, and the same sentences using neutral style. Then, the regression function is learned using the DKPLS method due to the nature of emotions, which are expected to introduce complex temporal dynamic effects (as the effect of emotions is highly dependent on the linguistic content, which is time varying), and possible nonlinear effects that can be well captured by the system.

The training system is designed to be able to handle several speakers in training by using VTLN to remove the variability due to the identity, although it is not strictly necessary if enough examples of the target emotion are available with only one speaker. Furthermore, it is encouraged to train the system with only one speaker whenever possible, since VTLN does not remove the identity variations perfectly, and each speaker expresses emotions in a slightly different way in the spectral sense. However, a gender dependent model can be built using VTLN with satisfactory results (see chapter 6), which allows to have more examples in the training function estimation.

The complete training system is depicted in figure 5.5. As shown in the figure, the system requires parallel sentences parameterized with MCCs and BAPs. The parameters first go through a *voice activity detector* (VAD), which removes the silent frames based on a power threshold, since they barely contain spectral information and most likely would worsen the resulting function. In practice, this can be achieved by thresholding the first MCC, which is directly related to the power in the frame, although in this context the actual power is used, and was computed beforehand.

Once the silent frames are removed, VTLN is applied to the MCC values, but the parameter is estimated using only the neutral MCCs for the likelihood calculation. This is because it is assumed that VTLN removes the effect of the identity, but the identity of the speaker is the same for both sentences. In addition, the estimation of two different parameters would partially remove the effect of the emotion, since the VTLN transformation would try to make it as similar as possible to the reference model, which is built with neutral sentences.

Finally, the warped features are aligned using DTW, and the aligned parameters are then introduced into the DKPLS training scheme. The first MCC is not included in the training of the transformation function and it is copied from the source in the conversion process. This is because the first MCC describes the energy of the frame and its value is usually rather different in scale and range from the rest of the coefficients, and if it were introduced in the system it would be likely to dominate the kernel formation process due to the difference in scale, which would worsen the resulting function.

DKPLS is able to handle collinear input data, and therefore partially linear dependent regressors may be used. For this reason, both MCCs and BAPs are used as input regressors simultaneously to predict either the target MCCs or the target BAPs, since they are highly correlated and information can be extracted from both of the features [Sil11].

5.4 Prosody conversion

In the context of this thesis, the prosody conversion subsystem creates a F_0 contour that tries to mimic the emotional speech style, and modifies the duration trend so that the speaking rate is closer to the emotional rhythm.

The main unit for prosody conversion is the syllable, and it is assumed that the effects that emotions produce in prosody are essentially observable at this level. Prosody modeling at syllable level is relatively common in VC systems, and it has been used in elaborate prosody transformation systems [Hel07b].

In order to do the transformation, CART regression is used to obtain the syllable related parameters (see annex A.4 for an in-depth description of the CART regression method). The predicted outcomes are scales for the mean of F_0 in each syllable, the duration of the voiced part of the syllable, and the duration of the unvoiced part of the syllable; each of them returned by a different CART.

5.4.1 Training of CARTs

A key issue of a regression problem is how well the predictor variables describe the dependent variable. In this case, a precise description of F_0 and the duration of the syllables is desired to imitate the portrayed emotion adequately, and for this purpose, contextual information must be included in the predictors.

The regression method used to achieve an adequate description of all the emotion related phenomena is the CART, due its ability to handle both categorical and non-categorical predictors simultaneously to perform the transformation, because linguistic information can be easily represented in categorical variables. Furthermore, not all emotions are best represented with the same set of predictors, as they affect the vocal production system differently, which implies that a parameter selection must be performed before the actual model is trained. An overview of the training scheme is shown in figure 5.6.

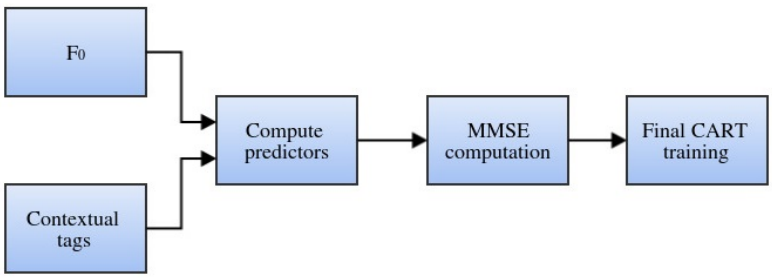


Figure 5.6: Block diagram of the prosody model training

The parameter selection is performed in the MMSE sense. To accomplish this, the MSE is evaluated using *2-fold cross validation*. First, optimally pruned CARTs are trained with every possible combination of predictors (see table 5.1) using a random subset of the available training data (training set). Secondly, the prediction error is computed as the average of the MSE for each sentence in the remaining subset (test set). The final CART is trained using the predictor subset that resulted in the lowest MSE.

In addition, an outlier removal is performed by removing 1% of the data from the tails of the distribution of the training parameters, to prevent miscalculations that may be affect the final regression.

Computation of the predictors

The total set of predictors before any selection is performed is a collection of 11 different features related to the linguistic content of the sentence and to the prosody of the sentence, as depicted in table 5.1.

Predictor name	Description	Categorical
isMonosyllabic	The word that the syllable belongs to is monosyllabic (yes/no)	YES
partOfSpeech	Grammatical category of the word that the syllable belongs to (9 categories)	YES
isTonical	Lexical stress of the syllable (stressed/non-stressed)	YES
posInWord	Position of the syllable in the word (0-1)	NO
posInSentence	Position of the word in the sentence (initial, initial-mid, middle, mid-end, end)	YES
durSyllRatio	Ratio of the duration of the syllable to the mean duration	NO
vuvRatio	Ratio of the voiced part of the syllable to the total duration	NO
prevSyllDurRatio	Ratio of the duration of the syllable to the duration of the previous syllable	NO
meanF0Ratio	Ratio of the mean F_0 in the syllable to the global mean of F_0	NO
isLast	Current syllable is the last in the sentence (yes/no)	YES
isPrevLast	Current syllable is the penultimate (yes/no)	YES

Table 5.1: Description of the different predictors used for the CART training

The computation of this set of features requires the availability of several contextual tags that have to be obtained beforehand. The required tags are the temporal boundaries of the syllables and the words, the stress of the syllables, and the part of speech of each word. In the context of this thesis, annotated databases are used, in which each sentence has an associated tag file with the phoneme boundaries, and the sentence content. The parts of speech are obtained automatically using the TreeTagger software [Sch95] on the sentence content from the annotation file, and then the categories are compressed to the 9 basic categories: Pronouns, adjectives, verbs, prepositions, adverbs, nouns, interjections, conjunctions and determinants.

Duration CARTs

Two regression trees are built to obtain the duration scales, since the duration scales for the unvoiced and voiced parts of the syllables are predicted.

In the training phase, the duration scales are normalized by the mean speaking rate of each sentence (as shown in equation 5.4), to prevent the CART from having to predict this approximately constant scale. The mean speaking rate quotient between emotions is stored additionally to perform the regression afterward:

$$dSc_{ij} = \frac{\frac{emotionalDuration_{ij}}{emotionalRate_j}}{\frac{neutralDuration_{ij}}{neutralRate_j}} = \frac{emotionalDuration_{ij}}{neutralDuration_{ij}} \frac{neutralRate_j}{emotionalRate_j} \tag{5.4}$$

$$rateQuotient = \frac{1}{M} \sum_{j=1}^{M} \frac{emotionalRate_j}{neutralRate_j} \tag{5.5}$$

Where j represents the jth training sentence, i represents the ith syllable in the sentence j, M is the number of syllables in the sentence j, and dSc_{ij} represents the training scale of the syllable i and sentence j.

F_0 CART

An analogous process to that of the duration CARTs is performed to train the F_0 CART.

The regression tree is trained using the ratio of the mean F_0 of the emotional sentence to the mean F_0 of the neutral sentence calculated syllable-wise, and normalized by the global F_0 averages of each sentence, as shown in the following equation:

$$f0Sc_{ij} = \frac{emotionalF_{0ij}}{neutralF_{0ij}} \frac{neutralMean_j}{emotionalMean_j} \tag{5.6}$$

where $neutralMean_j$ and $emotionalMean_j$ represent the global mean of the neutral F_0 and the global mean of the emotional F_0 respectively, averaging only over the voiced segments.

Analogously to the duration scale procedure, the mean ratio of F_0 averages is stored for the regression process. Additionally, since F_0 is updated every 5 ms, any syllable in which the voiced part lasts less than 50 ms is discarded from the training set to prevent possible small local variations from affecting the final result.

5.4.2 Feature conversion

The CARTs provide a set of scales significant at syllable level, but in order to be able to synthesize speech, a frame-wise description is needed. To obtain an F_0 representation valid for synthesis, a two step process is applied. First, the syllable level scales for the F_0 syllable averages are transformed into a soft contour using spline interpolation, and then F_0 is scaled using the interpolated contour. The scaled F_0 is divided into syllable segments, and each segment is resampled in time so that it fits the obtained duration.

Global contour scaling

The transformation of the F_0 sequence is performed via simple scaling. However, the scale must be a function of time, since the intonation patterns vary with time as a function of the emotion and the linguistic content. To accomplish this, CART regression is used to obtain syllable dependent scales. The scales are calculated over the mean F_0 in each syllable, and therefore need to be transformed into a sample-wise scale vector.

The most straightforward approach would be to simply scale every sample in the syllable with the same scale, but this would generate undesired steps in the F_0 contour. To solve this inconvenience, F_0 is described using a soft contour extracted from the mean F_0 values in the syllables. The conversion is performed in a five step process:

1. Obtain a vector $\mathbf{S}$ containing the mean values of F_0 syllable-wise
2. Produce the scale vector with the normalized scales obtained from the CART multiplied by the mean ratio of F_0 averages between emotions that was stored within the model.
3. Scale the vector $\mathbf{S}$ with the scale vector, generating the scaled vector $\mathbf{S}_{sc}$
4. Generate soft interpolated contours $S^i(n)$ and $S^i_{sc}(n)$ using cubic spline interpolation over $\mathbf{S}$ and $\mathbf{S}_{sc}$ respectively, assuming that the elements of the vectors $\mathbf{S}$ and $\mathbf{S}_{sc}$ are located in time in the middle of the corresponding syllables.
5. Obtain the scaled fundamental frequency sequence $\hat{F}_0$ by multiplying the original F_0 by the quotient of the obtained soft contours:

$$\hat{F}_0(n) = \frac{S^i_{sc}(n)}{S^i(n)} F_0(n)$$

The process is depicted in figure 5.7. The pattern illustrated in the figure corresponds to boredom, in which it is characteristic to start with a high F_0 and fall down to a low level. As it can be seen, the scaling process is able to capture the emotional trend while mostly preserving microprosodic phenomena.

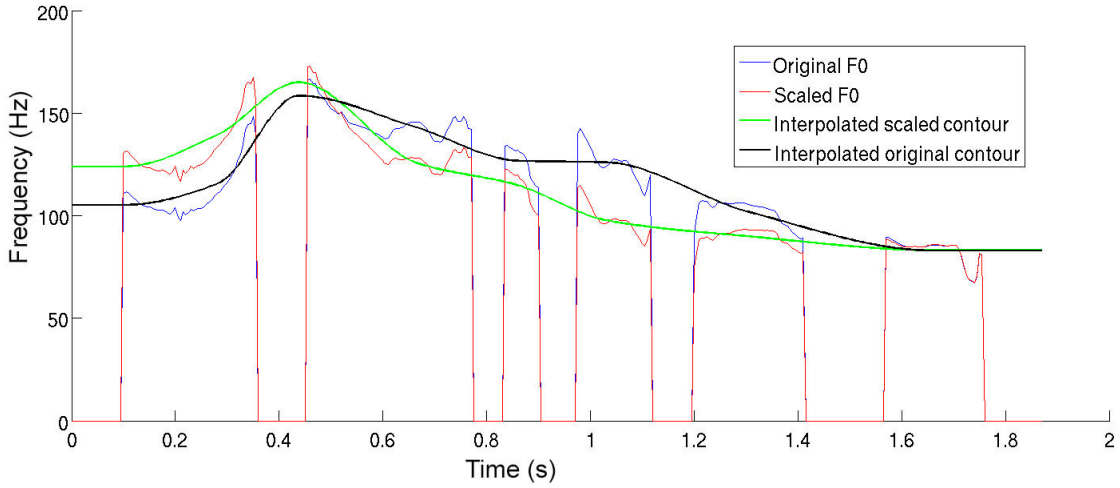


Figure 5.7: Scaling process for the fundamental frequency, converting from neutral style to boredom

Duration fitting

In order to achieve the corresponding speaking rate, the duration trend is modified. The duration is represented as function of the syllable. However, a distinction is made between the voiced part of the syllable and the unvoiced part of the syllable. This way, it is assumed that the voiced sounds and unvoiced sounds inside the syllable unit can vary their duration differently depending on the emotion.

To fit the parameters to the duration scales, *resampling* is applied. The process requires the splitting of the temporal sequences into voiced parts and unvoiced parts, and each of this parts is resampled according to the corresponding scale. The procedure can be explained in five steps:

1. Produce the scales vectors with the normalized scales obtained from the duration CARTs multiplied by the mean ratios of duration averages between emotions (voiced and unvoiced) that were stored within the model.

2. For each syllable segment, use the voicing decisions provided by STRAIGHT in F_0 to separate the voiced and the unvoiced parts in the F_0 sequence, and in the MCCs and BAPs sequences.

3. For every subsegment resulting from the splitting, subtract the mean of the first and the last element in the subsegment to improve the accuracy in the resampling process.

4. Resample each subsegment using linear interpolation to fit the corresponding scaled duration.

5. Concatenate all the subsegments in the proper order to create the new duration scaled sequence.

This process preserves the synchrony between F_0 and the spectral representation, which is essential in the process of re-synthesis.

6. Evaluation and discussion

This chapter describes the details of the procedures used to evaluate the effectiveness of the proposed methods. First, the databases used for evaluation are described, along with the recording conditions and special requirements that each of them fulfill. Subsequently, the results obtained under these conditions are presented. Finally, the performance of the system is discussed based on the obtained results.

6.1 Databases of emotional speech

In order to evaluate the performance of this work, two different databases in different languages were used. The languages used are Spanish and German, and both were recorded under strict conditions.

6.1.1 German database

The German database is called Berlin Database of Emotional Speech [Bur05], which contains 535 recordings of ten speakers, five men and five women, simulating six different emotional states plus a neutral style. The database is therefore an *acted database* (see section 3.1). The portrayed emotional styles are anger, boredom, disgust, fear, joy and sadness; and the corpus is comprised of ten sentences with neutral semantic content. All the recordings are available at 16 kHz sampling and 16 bits per sample.

The authors of this database made a great effort to achieve high naturalness in the collected emotions. Around 40 speakers went through a selection process in which they had to record one sentence in each of the emotional states in an office environment. Afterward, three experts evaluated the naturalness and recognizability of the performance, and finally selecting the ten actors that participated in the experiment.

As the content of the sentences is nonsense, an actual emotional recording representing the intended emotional state was played for the actors prior to each recording to prevent overacting and reinforce naturalness (e.g. happiness after winning a large amount of money in the lottery or sadness caused by losing a very good friend or relative). In addition, two phoneticians gave advice and instructions to the actor to improve naturalness, such as not to shout when portraying anger. If in any moment the actor considered that the emotion was not adequate, the recording was repeated, storing every repetition.

To achieve a high audio quality, the recordings were performed in the anechoic chamber of the Technical University Berlin, Technical Acoustics Department. Recordings were taken with a sampling rate of 48 kHz, and they were downsampled afterward to 16 kHz.

After the recording sessions were finished, around 800 utterances were available. These sentences were subjected to an evaluation test, in which 20 listeners had to identify the emotion and rate the naturalness. In order to remain in the final database, the sentences had to have at least 80% recognition rate and 60% rate of naturalness. After the selection process, 535 sentences remained, which means that emotions are not balanced in the database, and some are more represented than others. Finally, the phoneme boundaries of each recording were annotated by expert phoneticians, with the aid of spectrograms and laryngograph records.

This fact has some impact on the system, since the models are gender dependent and therefore the average amount of data per emotion is halved, and in the least represented emotions, such as disgust, the available amount of data was somewhat small (e.g. 11 sentences in the case of disgust uttered by male speakers), which may result in an inaccurate transformation function. Due to the size of the training set, it is unfeasible to build speaker dependent models, and therefore the models for this database were built exclusively in a gender dependent way, exploiting VTLN in the training process.

6.1.2 Spanish database

The Spanish database used in this thesis is part of a bigger multilingual database in English, Slovenian, French and Spanish developed as a collaboration between the University of Maribor, Slovenia; Lernout & Hauspie, Belgium; and the Polytechnic University of Catalonia, Spain [Amb00].

The databases contain utterances produced by a male speaker and a female speaker simulating six emotional states, compliant with the MPEG-4 standard [Ost02], which means that this is an *acted database* as well. The portrayed emotions are anger, disgust, surprise, joy, sadness and fear. All the recordings are available at 16 kHz sampling and 16 bits per sample.

The recordings were performed in a silent room, divided by a thick glass window. The actors had to read the sentence in the indicated emotional style from a computer screen. The computer and the screen were located at the other side of the window to prevent additional noises caused by the computer, such as the computer fan. Two operators were checking the recording process, one of them checked that the content of the sentence corresponded exactly with the sentence prompted to the actor, and the other checked the recording system. No deviation or mispronunciation was allowed. The recordings were initially made at 32 kHz sampling and afterward downsampled to the required 16 kHz.

There corpus is comprised of 186 parallel utterances in each emotional category, produced by both speakers, which makes the database highly suitable for VC purposes or even speech synthesis, due to its extensive size. The utterances comprise isolated numbers and words, sentences in affirmative, exclamatory or interrogative forms and paragraphs, as described in the table 6.1. The corpus was selected to have examples of all the Spanish phonemes in different parts of the sentences.

Utterance number	**Description**
1-100	Affirmative sentences including short and long ones.
101-134	Interrogative and stressed sentences.
135-150	Paragraphs.
151-160	Digits.
161-184	Isolated words.

Table 6.1: Content of the different utterances in the Spanish database

The corpus was recorded twice in the six MPEG-4 emotions plus neutral. The recording of the styles took place in two different sessions delayed more than 15 days. In the context of this thesis, only the affirmative sentences were used, with a top of 40 sentences for the training process to be compliant with the requirement of low data of VC systems. The training was performed with sentences from the first recording session, and the tests were done with sentences from the second session.

The database contains the phonetic transcription of every sentence in the corpus, which was used afterward to obtain the phoneme temporal boundaries using forced alignment with HMMs [Bru93] in the Polytechnical University of Catalonia.

6.2 Objective results: Mel-Cepstral distortion

This section is intended to give a measure of the performance of the whole spectral conversion system. To evaluate the system, the *mel-cepstral distortion* (MCD) is calculated. The MCD is defined as [Tod07]:

$$MCD(dB) = \frac{10}{\ln(10)} \sqrt{2 \sum_{d=1}^{D} (c(d) - \hat{c}(d))^2}$$

Where $c(d)$ and $\hat{c}(d)$ are the dth MCCs from the target vector and the converted vector respectively.

However, in the proposed system the target features are unclear, since the system does not try to mimic the features of a specific speaker, but a certain identity-warped version of the reference speaker that matches the identity of the input speaker. To partially overcome this problem and get measurements, the scheme presented in figure 6.1 is proposed.

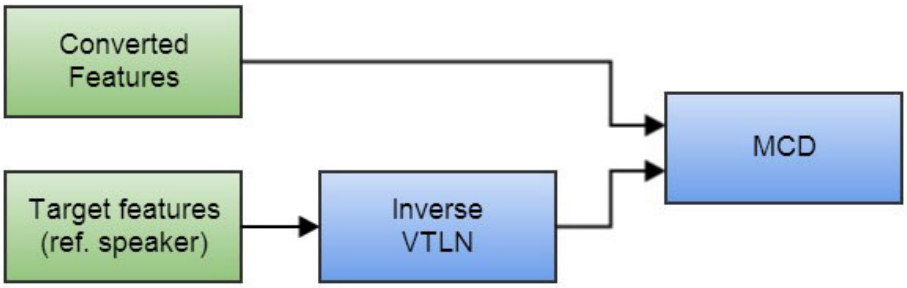


Figure 6.1: Calculation of mel-cepstral distortion

This scheme uses VTLN to transform the identity of the reference speaker to that of the input speaker. To follow the convention used in figure 5.1, the transform is indicated here as inverse VTLN.

The result are measured using the Spanish database, taking advance of its extension. The reference speaker is the male speaker, and the method was applied to both the male speaker and the female speaker. The results are gathered in table 6.2. Additionally, a speaker dependent DKPLS model was built without the presence of VTLN, whose result can be directly compared to the target utterance from the database.

The results here show that the presence of VTLN scarcely increases the distortion in the case of the reference speaker (male in this case), and effectively reduces the distortion in the case of a different speaker (female in this case).

		No conversion	Converted	Speaker dependent converted
Anger	Male	6.837 dB	6.162 dB	6.049 dB
	Female	7.2 dB	6.744 dB	6.053 dB
Disgust	Male	6.653 dB	6.17 dB	6.166 dB
	Female	7.5 dB	6.943 dB	6.035 dB
Fear	Male	7.013 dB	6.573 dB	6.429 dB
	Female	7.507 dB	7.241 dB	5.649 dB
Joy	Male	6.648 dB	6.923 dB	7.018 dB
	Female	7.33 dB	7.44 dB	7.503 dB
Surprise	Male	7.368 dB	6.671 dB	6.535 dB
	Female	7.709 dB	7.148 dB	6.296 dB
Sadness	Male	7.368 dB	6.671 dB	6.535 dB
	Female	7.903 dB	7.431 dB	5.464 dB

Table 6.2: Mel-cepstral distortion in the Spanish database

On the other hand, results show that the reduction in distortion may not seem as high as it could be expected for some emotions. Nevertheless, the fact that VTLN does not perform a perfect identity conversion must be taken into account when interpreting this results, and therefore reference utterance which the converted utterance is being tested against does not perfectly represent the target, but an approximation of it. In addition, the absence of another male speaker to test the effectiveness of the system in a gender-dependent way makes difficult to interpret the result, since convertig from male to female forces VTLN significantly more than intra gender conversion. However, the results shown in section 6.3 show that the emotions are converted successfully independently of the speaker.

6.3 Subjective results: Listening test

In order to evaluate the performance of the system, a listening test was performed. Two different listening test were elaborated, one for the German language and another for the Spanish language using the available databases. The test was performed online, but the instructions encouraged to do the test in a silent environment and with good quality headphones.

The test consisted on ten questions per emotion category, evaluating four random emotions from the total set of six, to prevent fatigue in the listener. In every question the listener was shown an utterance, and he was asked to rate how well the audio represents the intended emotion, which was indicated in the question test. The possible grades ranged from one to five, being 1 equivalent to "No emotion present or the emotion does not correspond" and 5 to "The emotion is perfectly portrayed". Additionally, the listener was shown a face icon that represents the emotion indicated in the question, to help matching the utterance with the emotion. An example question from the online test is shown in the annex A.5. There was an additional introductory question to show the emotional level of the utterances, which contained an original acted emotion from the database.

In every question, the utterance was selected randomly between neutral style, original acted emotional utterance from the database, and converted by the system; but this was not indicated to the listener. Since it is expected to have low scores on the neutral style and high scores on the original acted utterances, these two groups are slightly less represented, appearing with probability 30% each, whereas the converted sentences appeared with probability 40%.

The results of the tests are presented here as box plots. The box plot is a representation in which the distribution of the data is plotted as a box, where the lower edge of the box represents the 25th percentile of the distribution and the upper edge represents the 75th percentile. The median of the distribution is represented as a crossing red line and the mean is represented with an asterisk inside a circle. Box plots may also have lines extending vertically from the boxes (*whiskers*) indicating variability outside the upper and lower percentiles. If any data is classified as an outlier, it is plotted as an individual point.

On the other hand, two *Welch's t-tests* were performed on the converted data to support the graphical visualization. The null hypotheses were that the average score of the converted data is equal to the average score of the neutral data in the first test, and that it is equal to the average score of the original data in the second test. The hypotheses are rejected at a 5% significance level, which means that the probability that the null hypotheses are valid is lower than 5%. This probability is called *p-value*, and they are presented along with the box plots for each emotion using this notation:

- p_{C-N}: P-value of the test with null hypothesis "The average score of the converted data is equal to the average score of the neutral data"

- p_{C-O}: P-value of the test with null hypothesis "The average score of the converted data is equal to the average score of the original data"

The results show that the average score of the converted data is never equal to the average score of the neutral data, which implies $p_{C-N} < 0.05$ in every case, and the average score of the converted data is always higher than the average score of the neutral data. However, in the case of testing the scores of the converted data against the scores of the original data, the results show that they are generally different, but in some emotions the test does not show enough evidence that the mean scores are different, achieving very high p-values (up to 0.93), which could possibly mean that the converted utterances perform essentially as well as the natural acted sentences in those cases.

6.3.1 Results with the German database

The results presented in this section correspond to the listening test conducted in German language. The total number of listeners was 12, out of which 6 were German speakers. The average number of examples is 80 per emotion category, with an average of 32 examples of converted utterances and 24 examples of neutral and original utterances respectively.

Emotion: Boredom

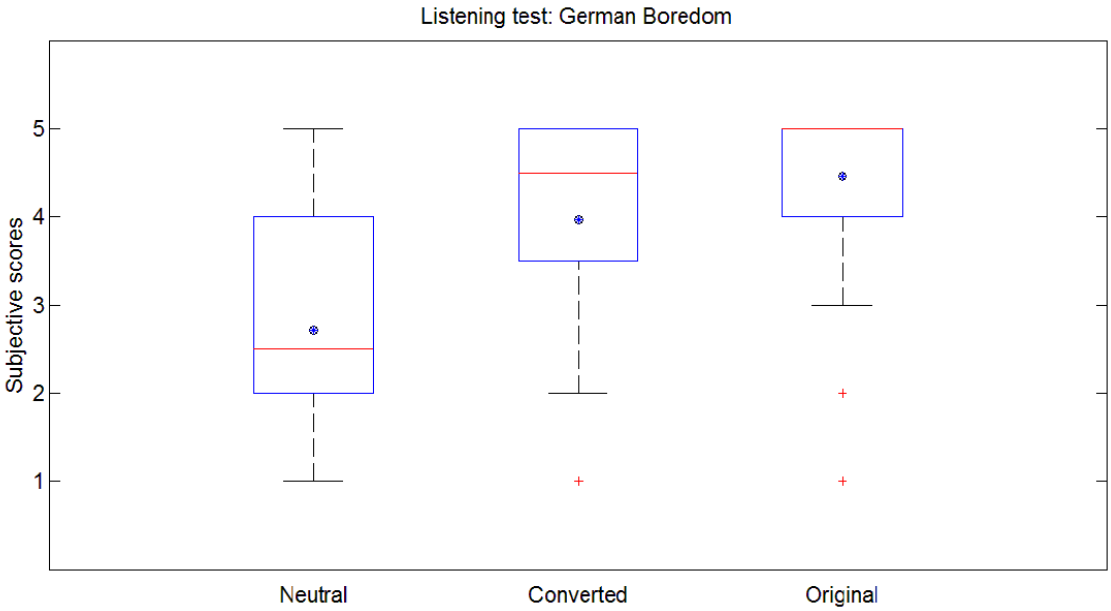


Figure 6.2: Box plot of the distribution of the scores for emotion "Boredom" in German language

$$p_{C-N} = 6.273 \cdot 10^{-3} \qquad p_{C-O} = 0.175$$

Emotion: Fear

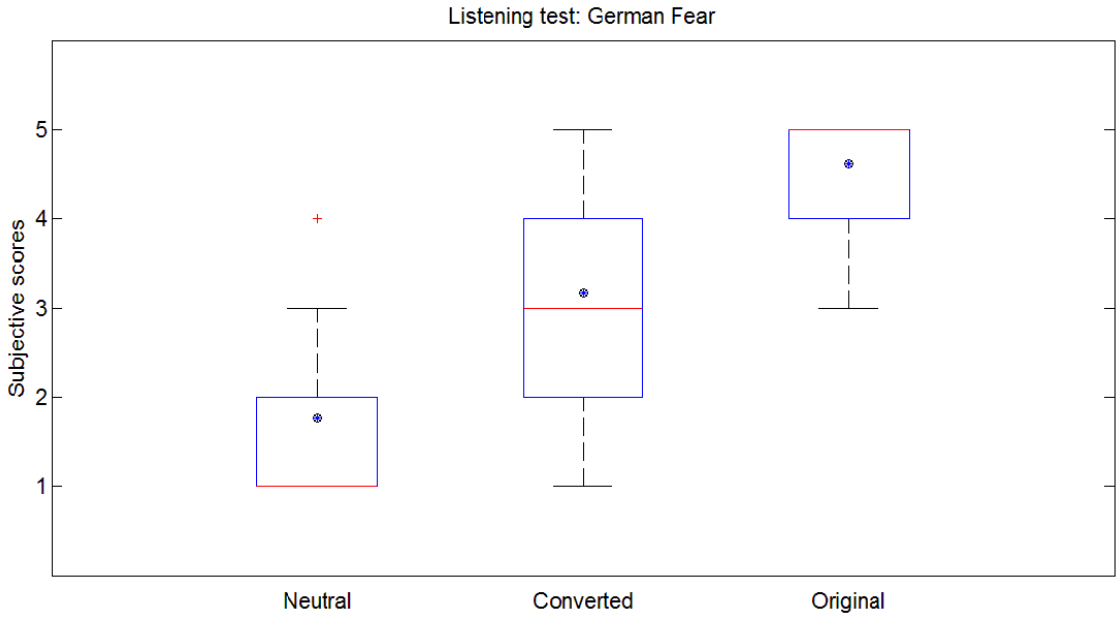


Figure 6.3: Box plot of the distribution of the scores for emotion "Fear" in German language

$$p_{C-N} = 5.716 \cdot 10^{-5} \qquad p_{C-O} = 6.842 \cdot 10^{-6}$$

Emotion: Disgust

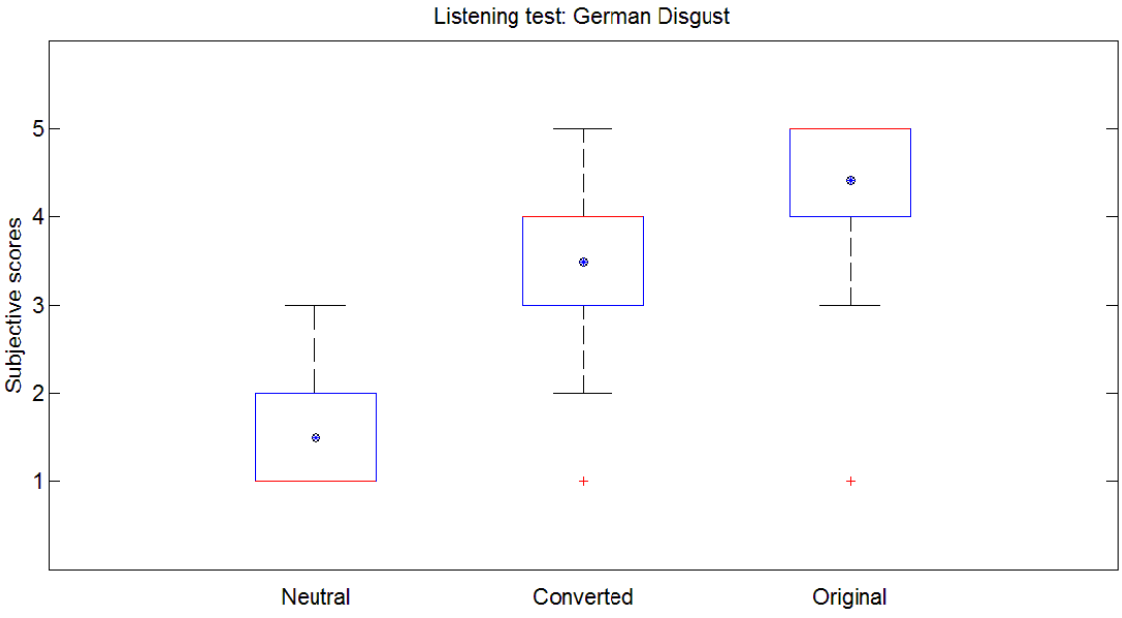


Figure 6.4: Box plot of the distribution of the scores for emotion "Disgust" in German language

$$p_{C-N} = 6.625 \cdot 10^{-7} \qquad p_{C-O} = 0.011$$

Emotion: Joy

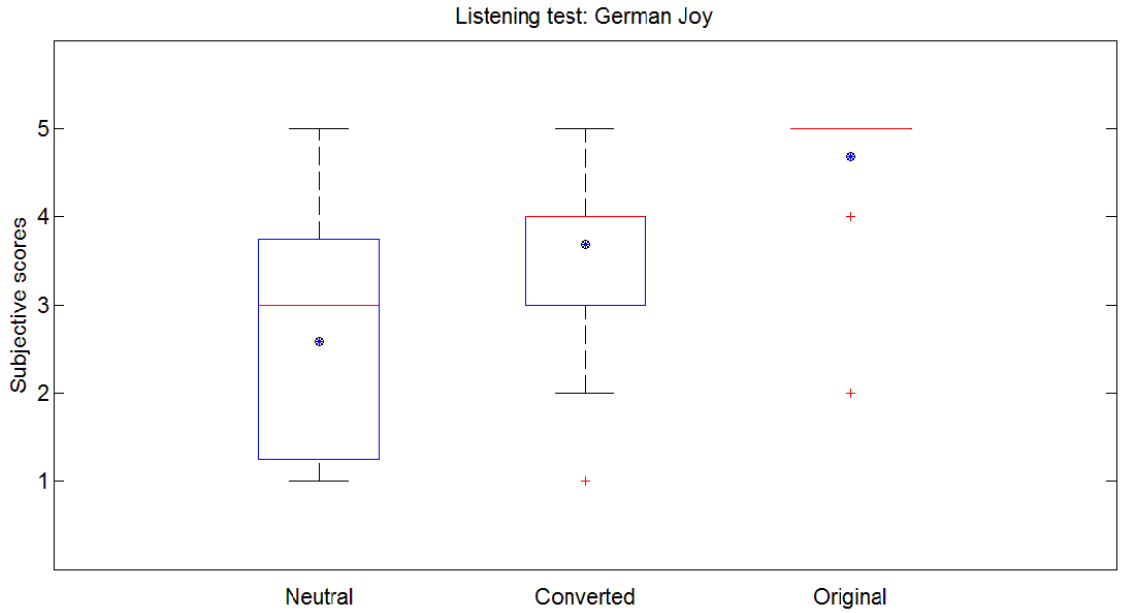


Figure 6.5: Box plot of the distribution of the scores for emotion "Joy" in German language

$$p_{C-N} = 2.729 \cdot 10^{-4} \qquad p_{C-O} = 4.701 \cdot 10^{-5}$$

Emotion: Sadness

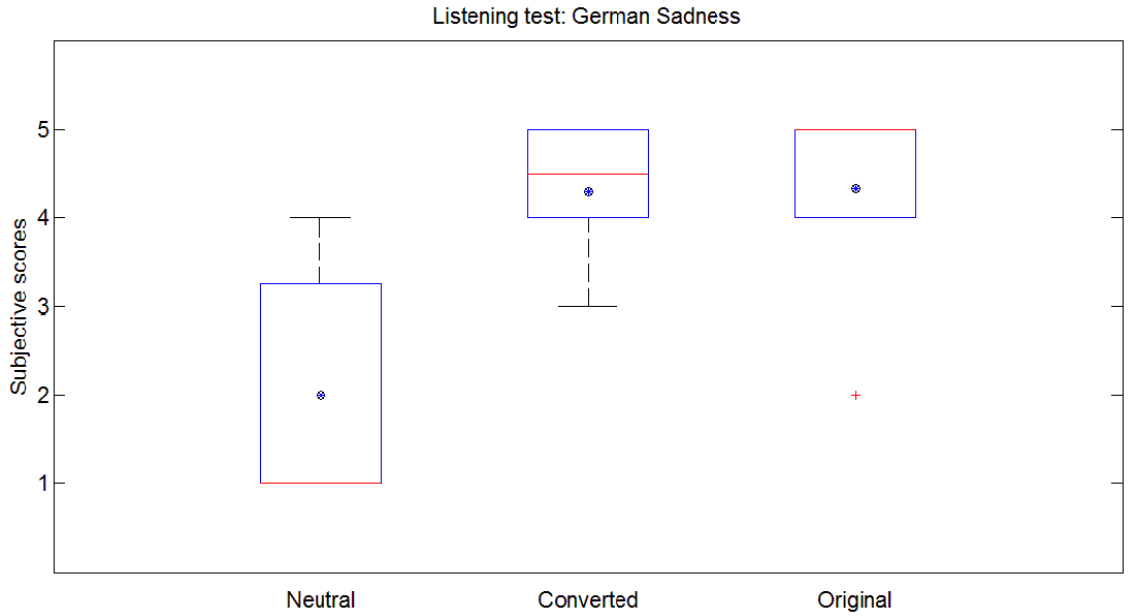


Figure 6.6: Box plot of the distribution of the scores for emotion "Sadness" in German language

$$p_{C-N} = 2.531 \cdot 10^{-4} \qquad p_{C-O} = 0.937$$

Emotion: Anger

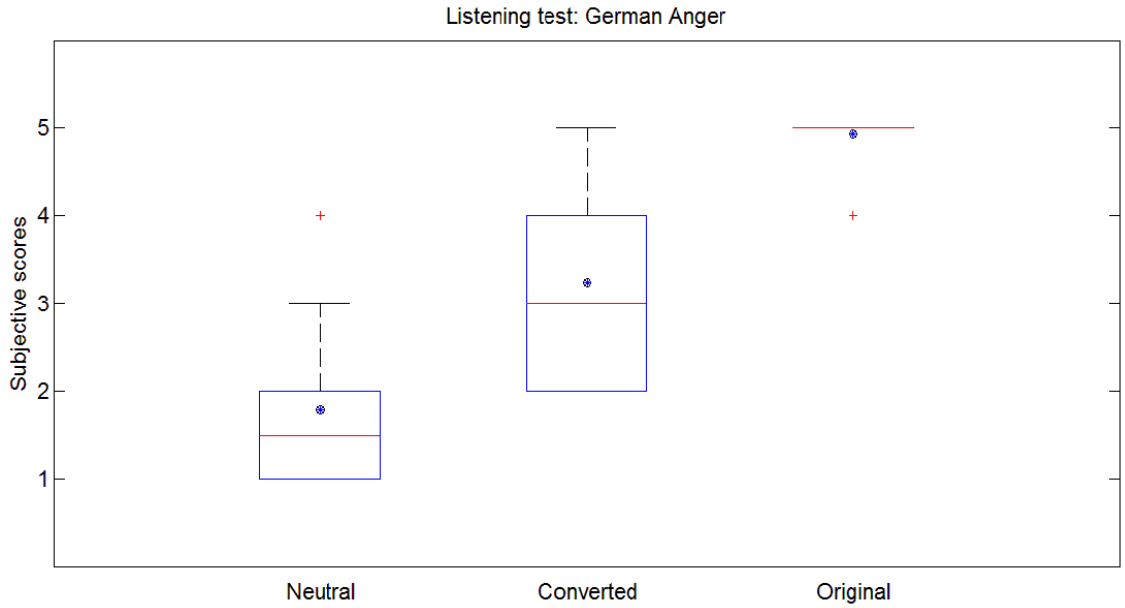


Figure 6.7: Box plot of the distribution of the scores for emotion "Anger" in German language

$$p_{C-N} = 1.1068 \cdot 10^{-3} \qquad p_{C-O} = 1.632 \cdot 10^{-5}$$

6.3.2 Results with the Spanish database

This section shows the results of the listening test conducted in Spanish language. The total number of listeners was 18, out of which 13 were Spanish speakers. The average number of examples is 120 per emotion category, with an average of 48 examples of converted utterances and 36 examples of neutral and original utterances respectively.

Emotion: Surprise

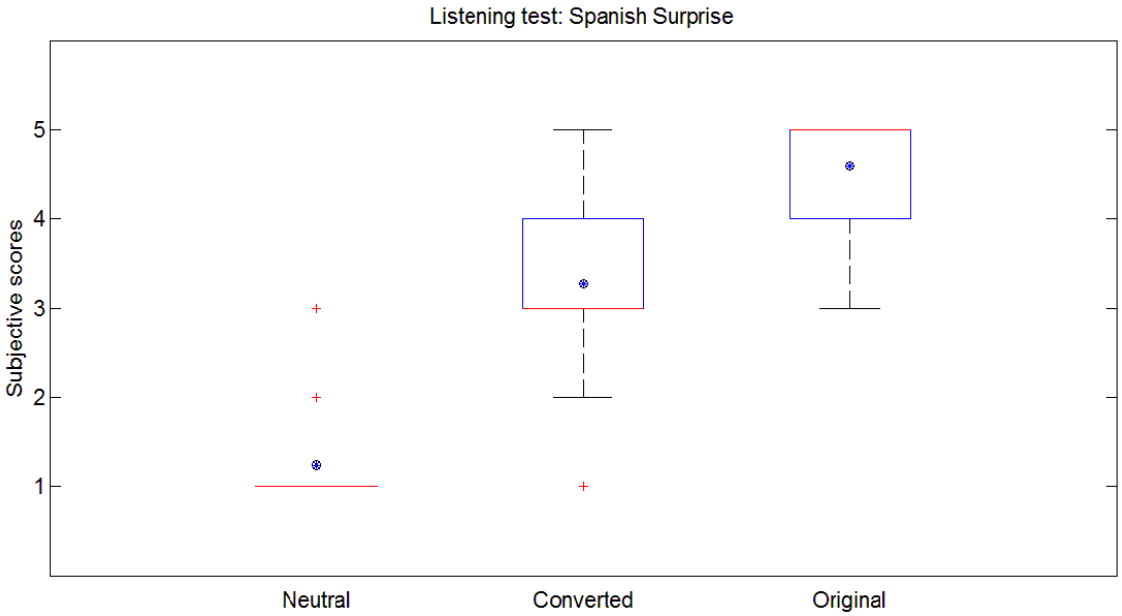


Figure 6.8: Box plot of the distribution of the scores for emotion "Surprise" in Spanish language

$$p_{C-N} = 2.74 \cdot 10^{-11} \qquad p_{C-O} = 2.842 \cdot 10^{-8}$$

Emotion: Fear

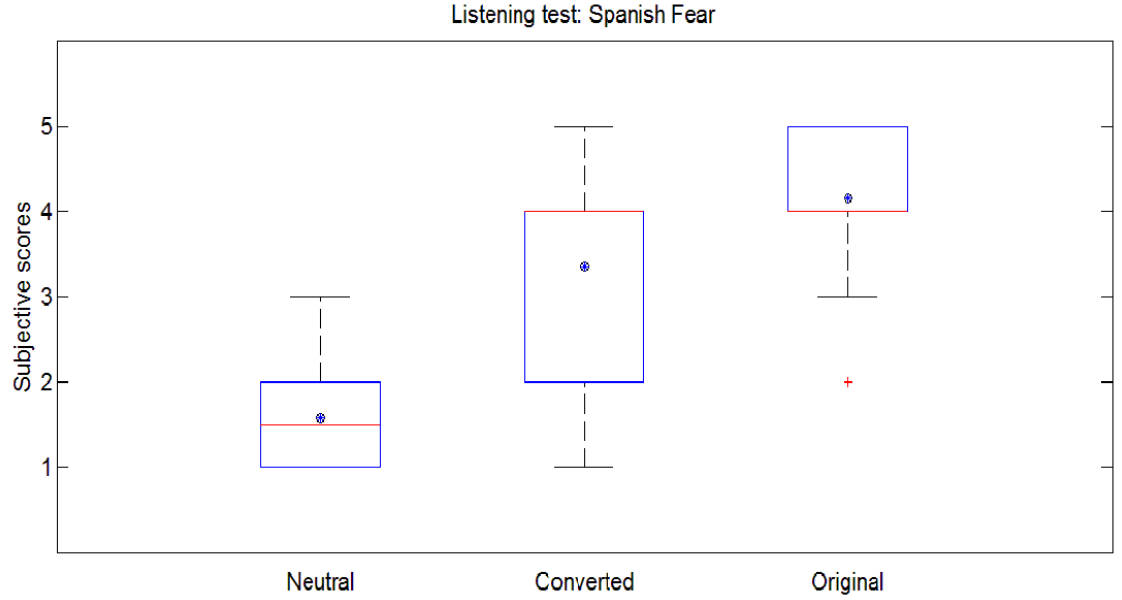


Figure 6.9: Box plot of the distribution of the scores for emotion "Fear" in Spanish language

$$p_{C-N} = 8.571 \cdot 10^{-9} \qquad p_{C-O} = 8.427 \cdot 10^{-3}$$

Emotion: Disgust

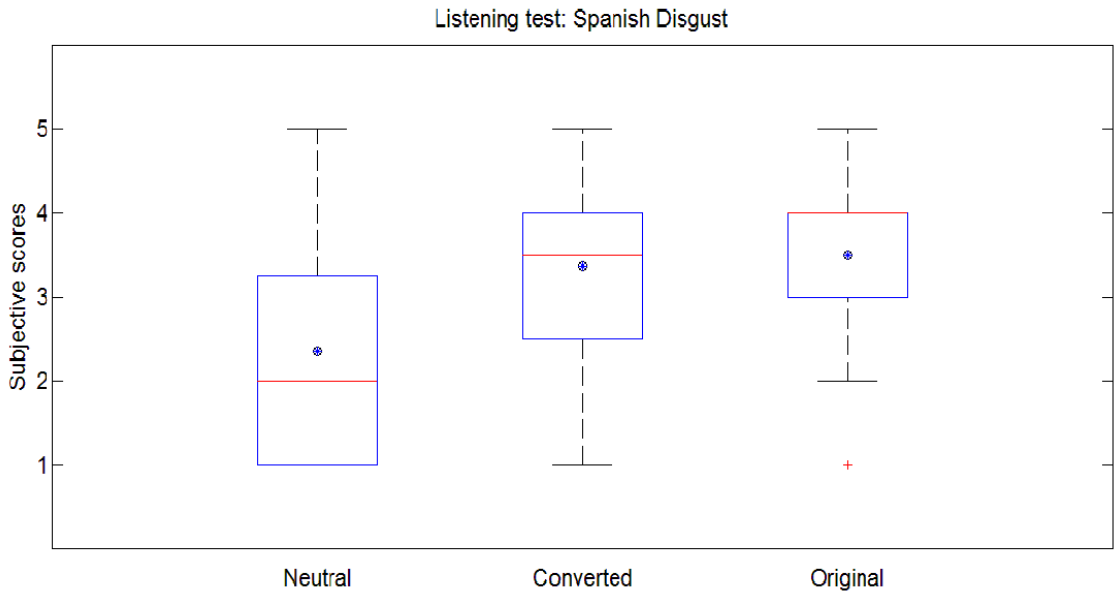


Figure 6.10: Box plot of the distribution of the scores for emotion "Disgust" in Spanish language

$$p_{C-N} = 0.012 \qquad p_{C-O} = 0.684$$

Emotion: Joy

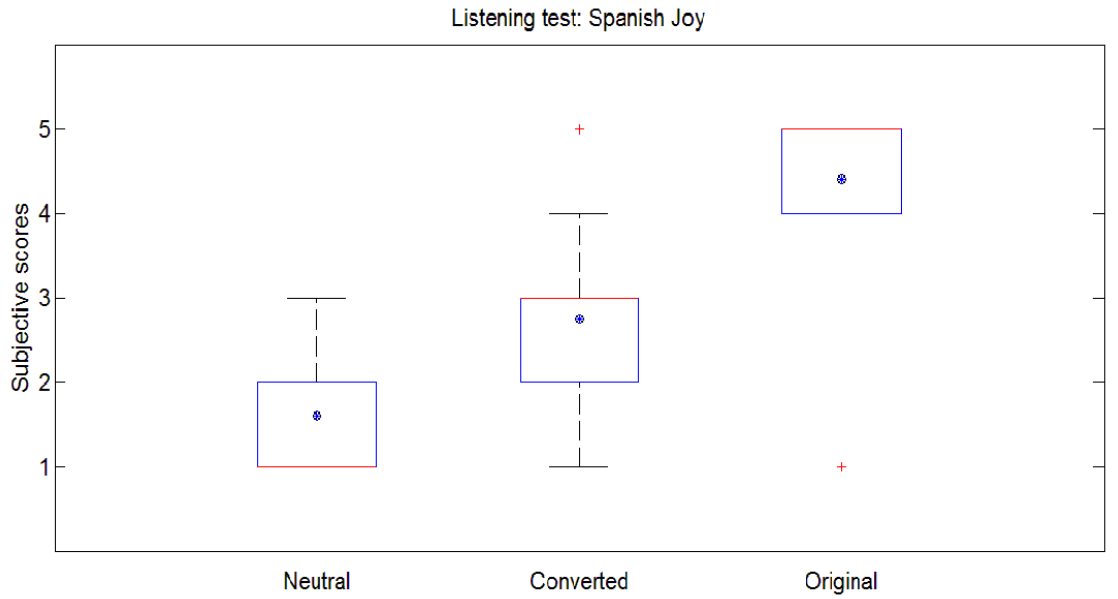


Figure 6.11: Box plot of the distribution of the scores for emotion "Joy" in Spanish language

$$p_{C-N} = 6.602 \cdot 10^{-4} \qquad p_{C-O} = 1.381 \cdot 10^{-9}$$

Emotion: Sadness

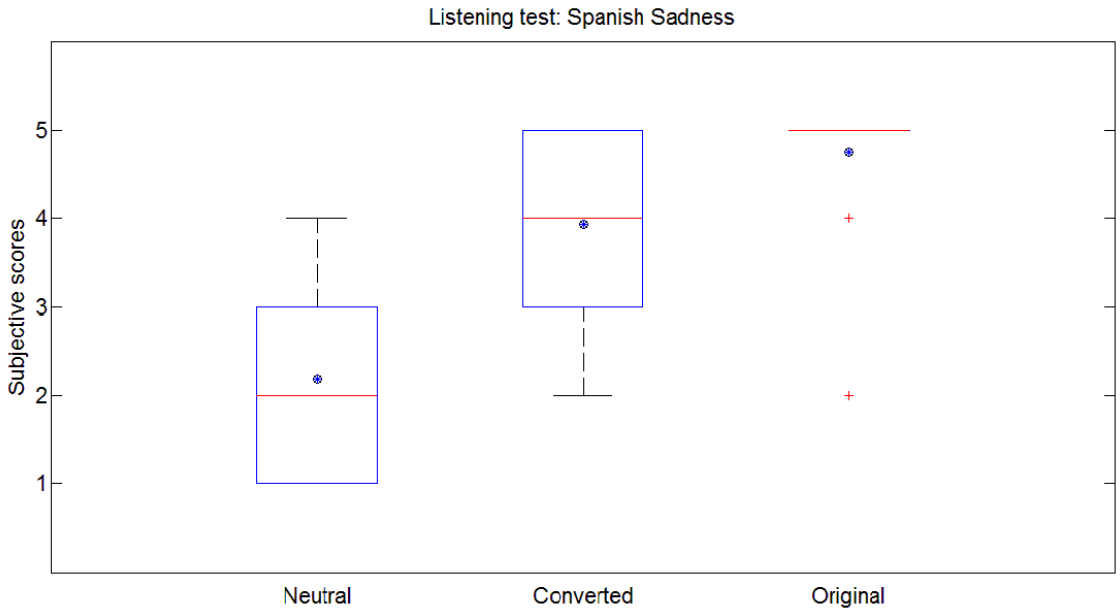


Figure 6.12: Box plot of the distribution of the scores for emotion "Sadness" in Spanish language

$$p_{C-N} = 1.396 \cdot 10^{-8} \qquad p_{C-O} = 9.921 \cdot 10^{-5}$$

Emotion: Anger

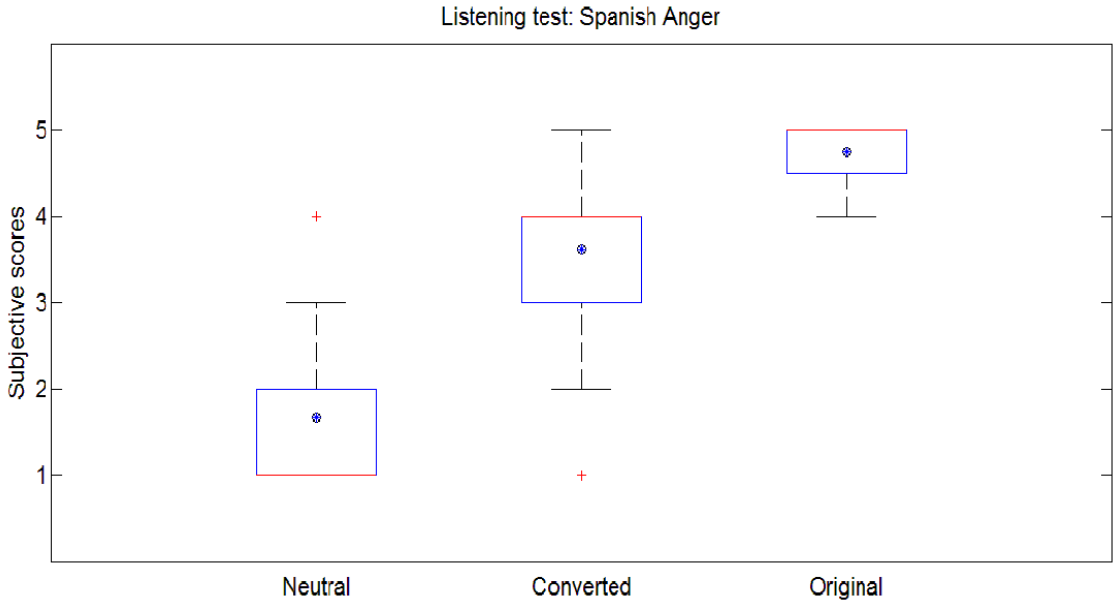


Figure 6.13: Box plot of the distribution of the scores for emotion "Anger" in Spanish language

$$p_{C-N} = 6.547 \cdot 10^{-9} \qquad p_{C-O} = 2.665 \cdot 10^{-5}$$

6.4 Discussion

The method described in this thesis has proved to be able to transform neutral speech into emotional speech with reasonably good results. The listening test results show that the emotions that are better represented with the system are those that are expressed mainly via prosody (e.g. sadness or boredom) in both languages, although these emotions are also the ones that create the most confusion when the listener is prompted a neutral style utterance. On the other hand, more complex emotions that show strong interaction between the spectral part and the prosodic part tend to have slightly lower scores in conversion (e.g. anger or fear), which is likely to happen because of the need of synchrony between certain spectral patterns with their corresponding prosodic patterns, which are not modeled in detail by this system. A phoneme-wise description would probably improve the result, although this would require more precision and complexity when estimating the temporal boundaries.

It is somewhat noteworthy the case of joy in both languages, which shows results closer to the neutral scores than the rest of emotions. The improvement that the system provides for this emotion is rather small compared to the neutral style, although it is enough to prove that the mean scores of the converted utterances are still higher. Joy is a special emotion because it is not expressed uniformly throughout the sentence, and it is more noticeable at the ending of the utterances than at the beginning. Thus, although the prosody model can model this effect due to the inclusion of temporal information in the predictors, the spectral model cannot capture this, and the spectral evolution with time introduces confusion in the regression function estimation, therefore producing a voice which is difficult to identify. This fact can be seen in the MCD, illustrated in table 6.2, which shows that for this concrete emotion, the distortion increases even with a speaker dependent model.

7. Conclusions and future work

In this work, voice conversion systems and their application in emotional speech generation tasks has been studied, and a conversion system has been proposed. In addition to being able to model emotions in speech, the proposed system had the additional task of converting the emotion independently of the input speaker to the system. This goal was achieved by performing a two-step voice conversion, exploiting VTLN to perform the initial identity conversion, and then using a complex transformation model to capture the effects caused by emotions.

The prosody subsystem has been developed to be context dependent, in order to model efficaciously the intonation contour and the speaking rate to fit that of the target emotion. To achieve this, regression is performed syllable-wise using parameter driven conversion, and these parameters are then used to generate the converted F_0 contour from the neutral one.

The proposed system has been evaluated in two languages by means of a listening test, proving to be successfully able to generate emotional speech out of neutral style speech. Additionally, objective measures have been provided that prove that the speaker independence does not increase distortion in the converted speech significantly. However, objective results suggest that it is advisable to build gender dependent models, since there are important differences in average vocal tract length between genders that the system has to model.

As a suggestion for further investigation, a more sophisticated VTLN algorithm could be incorporated in the system, such as the one proposed in [Sun03], which reduces spectral distortion more effectively than the bilinear transform used in this work, which is likely to produce a better identity normalization and hence a more reliable result. Nevertheless, the advantage of the bilinear transform method is that it can easily be embedded in the parameter extraction process.

Additionally, the system can be subjected to more extensive tests to measure the effectiveness, building speaker dependent models and gender dependent models, and develop listening tests that measure the relative degradation between the different systems.

On the other hand, the current prosody model is based on a syllable-wise description, and a phoneme-wise description is likely to increase the performance at expenses of more difficulty in the parameter obtaining. However, this issue should not represent a big problem in text to speech systems, where the content is perfectly controlled, and the system can perfectly be used within this context.

Another area of investigation is the possible substitution of the context dependent prosody model for a simpler model that is able to model somehow contextual information implicitly, such as the system suggested in [San14]. Following this lead, prosodic information could also be included as regressors for the spectral transformation, which possibly improves the results found in emotions such as joy.

A. Annex

A.1 Derivation of the complex cepstrum for AR-MA processes

Let $H(z) = \mathcal{Z}\{h(n)\}$ be the filter of an AR-MA process, then:

$$H(z) = \frac{Az^D \prod_{k=1}^{M_i}(1 - a_k z^{-1}) \prod_{k=1}^{M_o}(1 - u_k z)}{\prod_{k=1}^{N_i}(1 - b_k z^{-1}) \prod_{k=1}^{N_o}(1 - v_k z)} \tag{A.1}$$

Where M_i and M_o denote the number of zeros inside and outside the unit circle respectively, and N_i and N_o denote the number of poles inside and outside the unit circle.
To obtain the complex cepstrum, take logarithms:

$$\begin{aligned} \log(H(z)) = \log(A) + \log(z^D) + \sum_{k=1}^{M_i} \log(1 - a_k z^{-1}) + \\ + \sum_{k=1}^{M_o} \log(1 - u_k z) - \sum_{k=1}^{N_i} \log(1 - b_k z^{-1}) - \sum_{k=1}^{N_o} \log(1 - v_k z) \end{aligned} \tag{A.2}$$

Then use the Taylor series to express:

$$\log(1 - x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} \quad \text{for } |x| < 1 \tag{A.3}$$

Thus, for any of the terms of equation A.2:

$$\sum_{k=1}^{M} \log(1 - b_k z^{-1}) = \sum_{n=1}^{\infty} \left(\sum_{k=1}^{M} \frac{{b_k}^n}{n} \right) z^{-n} = \mathcal{Z}\left\{ \sum_{k=1}^{M} \frac{{b_k}^n}{n} \right\} \qquad n > 0 \tag{A.4}$$

$$\sum_{k=1}^{N} \log(1 - v_k z) = \sum_{n=1}^{\infty} \left(\sum_{k=1}^{N} \frac{{v_k}^n}{n} \right) z^{n} = \mathcal{Z}\left\{ \sum_{k=1}^{N} \frac{{v_k}^n}{n} \right\} \qquad n < 0 \tag{A.5}$$

In the region of convergency of the Z transform, which is $|a_k z^{-1}| < 1$ and $|v_k z^{-1}| < 1$. Hence, the complex cepstrum of $h(n)$, defined as $\hat{h}(n) = \mathcal{Z}^{-1}\{log(H(z))\}$ is:

$$\hat{h}(n) = \begin{cases} \log(A) & n = 0 \\ \sum_{k=1}^{N_i} \frac{{b_k}^n}{n} - \sum_{k=1}^{M_i} \frac{{a_k}^n}{n} & n > 0 \\ \sum_{k=1}^{N_o} \frac{{v_k}^n}{n} - \sum_{k=1}^{M_o} \frac{{u_k}^n}{n} & n < 0 \end{cases} \tag{A.6}$$

obviating the term z^D, which is just a signal delay.

Then, if the signal is minimum phase, every pole and zero is inside the unit circle, thus $v_k = 0$ and $u_k = 0$, and therefore $\hat{h}(n) = 0$ for $n < 0$.

A.2 Cepstrum of a windowed periodic signal

Let $s(n)$ be a windowed periodic signal with period N, then it can be expressed as $s(n) = s_{base}(n) * e(n)$, where $e(n)$ is a train of unit impulses:

$$e(n) = \sum_{k=0}^{M-1} \alpha_k \delta(n - kN) \tag{A.7}$$

Applying the properties of the cepstrum, it is possible to write $\hat{s}(n) = \hat{s}_{base}(n) + \hat{e}(n)$. To calculate the cepstrum of the impulse train, first calculate the Z transform and take logarithm:

$$\mathcal{Z}\{e(n)\} = \sum_{k=0}^{M-1} \alpha_k z^{-kN} = \prod_{k=1}^{M} (1 - c_k z^{-N}) \tag{A.8}$$

$$\log(E(z)) = \sum_{k=1}^{M} \log(1 - c_k z^{-N}) \tag{A.9}$$

Then, using equation A.3, it is obtained:

$$\sum_{k=1}^{M} \log(1 - c_k z^{-N}) = \sum_{n=1}^{\infty} \left(\sum_{k=1}^{M} \frac{{c_k}^n}{n} \right) z^{-nN} \tag{A.10}$$

And therefore, the cepstrum of the signal $e(n)$ is:

$$\hat{e}(n) = \mathcal{Z}^{-1} \left\{ \sum_{r=1}^{\infty} \left(\sum_{k=1}^{M} \frac{{c_k}^r}{r} \right) z^{-rN} \right\} = \sum_{m=1}^{\infty} \sum_{k=1}^{M} \frac{{c_k}^m}{m} \delta(n - mN) \quad m > 0 \tag{A.11}$$

A.3 Mean-scale transformation as a PDF equalization

Let $F_0(n)$ be a discrete white Gaussian process with mean μ_x and variance σ_x^2. This assumption implies that each sample x_n is uncorrelated with each other and they follow this Gaussian distribution:

$$f_{x_n}(x_n) = \mathcal{N}(\mu_x, \sigma_x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left[-\frac{(x_n - \mu_x)^2}{2\sigma_x^2}\right]$$

The function T that transforms a sample to fit a different PDF is:

$$y_n = T(x_n) = F_y^{-1}(F_x(x_n))$$

where $F_y(\cdot)$ represents the cumulative density function of the target PDF and $F_x(\cdot)$ represents the cumulative density function of the Gaussian PDF. If the target PDF is a Gaussian distribution with mean μ_y and variance σ_y^2, then:

$$F_y(y_n) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \int_{-\infty}^{y_n} e^{\frac{(t-\mu_y)^2}{2\sigma_y^2}} dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{y_n-\mu_y}{\sigma_y}} e^{-\frac{t^2}{2}} dt = \Phi\left(\frac{y_n - \mu_y}{\sigma_y}\right)$$

$$F_x(x_n) = \Phi\left(\frac{x_n - \mu_x}{\sigma_x}\right)$$

And therefore:

$$F_y^{-1}(\alpha) = \sigma_y \Phi^{-1}(\alpha) + \mu_y \Longrightarrow F_y^{-1}\left[\Phi\left(\frac{x - \mu_x}{\sigma_x}\right)\right] = \sigma_y \Phi^{-1}\left[\Phi\left(\frac{x - \mu_x}{\sigma_x}\right)\right] + \mu_y$$

So the final transformation function for the sample x_n is:

$$y_n = F_y^{-1}(F_x(x_n)) = \mu_y + \frac{\sigma_y}{\sigma_x}(x_n - \mu_x)$$

Which is the mean-variance scaling transformation.

A.4 Classification and Regression Trees

Classification and regression trees (CART) [Bre84] are machine learning tools that are used to predict the outcome of a variable. The goal of a CART is to predict the final value of this variable based on the known values of a certain number of other variables called *predictors*. If the predicted variable has a finite set of possible values (*classes*), the task is then classification, otherwise, the CART performs regression.

The CART addresses the problem by creating recursive binary partitions of the input data. It proceeds by finding the split that minimizes a certain measure of *impurity*. For each possible split based on the values of the predictors, the impurity is calculated, and the data is split in two subsets using the question that minimizes the impurity. The process is repeated for every subset recursively. This splitting sequence constitutes a binary tree structure, where each split can be regarded as a *node*, and the original set can be represented as the *root node*, as illustrated in figure A.1.

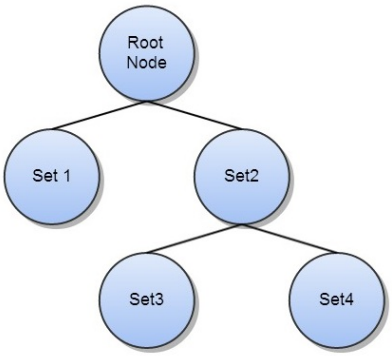


Figure A.1: Split sets as a binary tree

In classification tasks, the impurity measure is usually calculated as the *entropy* of the split data:

$$I(S_t) = -\sum_{x \in X} p(x) \log_2(p(x))$$

where S_t is the subset that would remain after the split at node t, X is the set of classes in S_t, and $p(x)$ is the proportion of elements in class x related to the number of elements in S_t. From this definition, it is clear that when the classification is perfect (i.e, all the elements in S_t belong to the same class), the impurity of the subset is null.

On the other hand, if the purpose is to perform regression, the impurity measure is defined as the MSE of the split subset:

$$I(S_t) = \frac{1}{N_t} \sum_{k=1}^{N_t} (y_k - \mu_t)^2$$

where N_t is the number of elements in the subset S_t, $\{y_k\}$ are the values of the dependent variable at node t, and μ_t represents the mean of the dependent variable at that node.

The size of the tree is an important issue, as very large trees can fit very well the training data, but generalize poorly to new data (known as the *bias-variance dilemma*). In order to determine the optimal size of the tree, a set of sub-trees of different sizes are built by means of *pruning*. The full tree (i.e. that in which no more splits are possible) is pruned by removing the node that provides the lowest change in purity, and the process is repeated recursively, generating a set of pruned sub-trees.

In order to select the optimal sub-tree, first, a cost function called *error-complexity measure* for the depth of the tree is established [Bre84]:

$$EC(T) = MSE(T) + \alpha L$$

where $MSE(T)$ denotes the mean squared error of the tree T, and L denotes number of leaves of the tree. The parameter α is free choice, and for each α, there is a pruned sub-tree that minimizes the error complexity measure.

The optimal parameter α is selected using *cross validation*. The training dataset is divided into M different subsets, and the optimal EC trees (varying α) are built for the $M-1$ training sets. Then, the optimal α is chosen as the one that minimizes the MSE in the testing set.

Once α is selected, a tree is grown from the complete dataset, in a way that minimizes the error-complexity measure using the selected parameter.

A.5 Example of listening test query

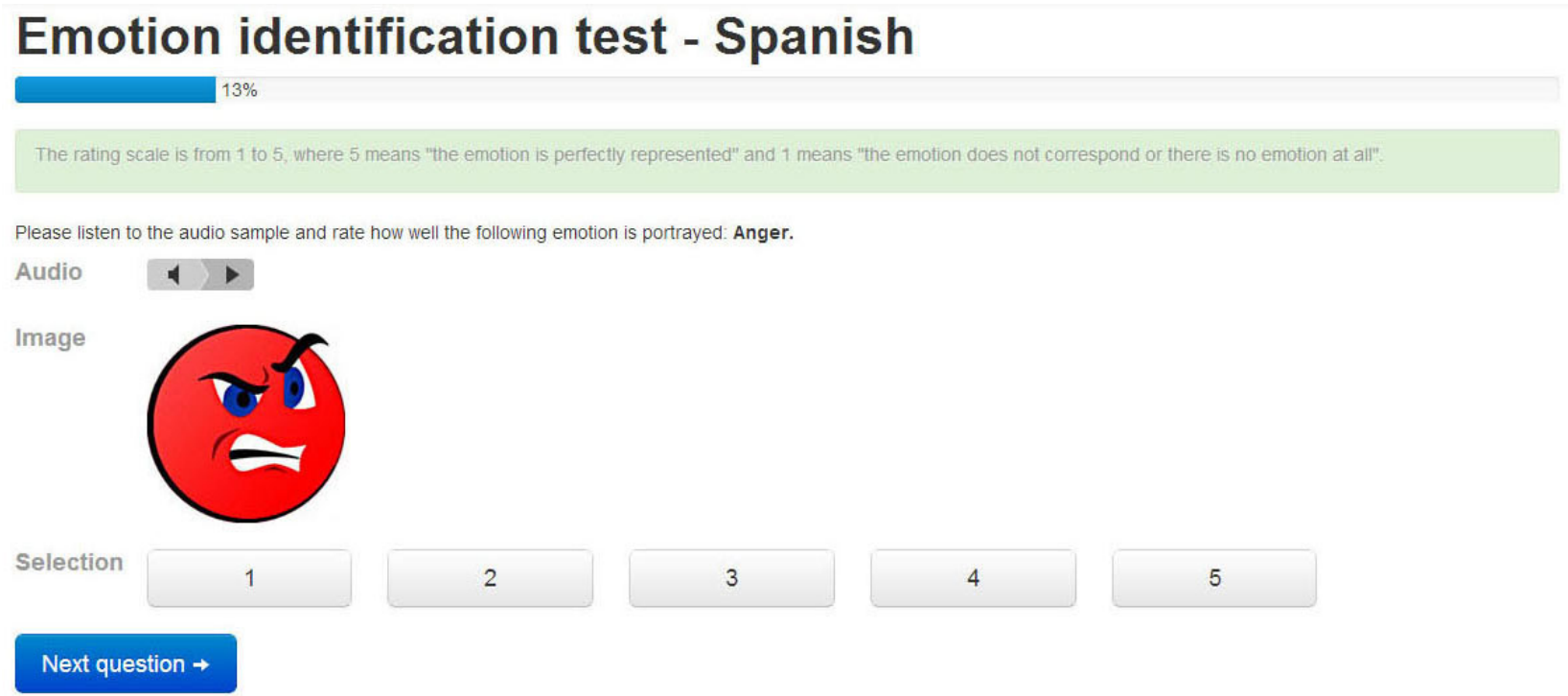


Figure A.2: Example of listening test question, querying for the emotion "Anger". The image has been rotated to enhance resolution.

References

[Ace93] A. Acero. *Acoustical and environmental robustness in automatic speech recognition.* Springer, 1993.

[Amb00] D. C. Ambrus. "Collecting and Recording of an Emotional Speech Database". Tech. Rep., Faculty of Engineering and Computer Science, Institute of Electronics, University of Maribor, 2000.

[Ban96] R. Banse and K. Scherer. "Acoustic Profiles in Vocal Emotion Expression". *Journal of Personality and Social Psychology*, Vol. 70, No. 3, pp. 614–636, 1996.

[Bar07] R. Barra, J. M. Montero, J. Macias-Guarasa, J. Gutierrez-Arriola, J. Ferreiros, and J. M. Pardo. "On the limitations of voice conversion techniques in emotion identification tasks". In: *Proc. of Interspeech 2007*, 2007.

[Bat00] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Nöth. "Desperately Seeking Emotions Or: Actors, Wizards, And Human Beings". In: *ICSA workshop on speech and emotion*, pp. 195–200, 2000.

[Ben03] K. P. Bennett and M. J. Embrechts. "An Optimization Perspective on Kernel Partial Least Squares Regression". In: *Advances in learning theory: methods, models, and applications; Proceedings of the NATO Advanced Study Institute on Learning Theory and Practice*, pp. 227–250, IOS Press, Louvain, Belgium, 2003.

[Bre73] R. Brent. *Algorithms for Minimization Without Derivatives*, Chap. 4. Prentice-Hall, 1973.

[Bre84] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees.* Wadsworth Inc., 1984.

[Bru93] F. Brugnara, D. Falavigna, and M. Omologo. "Automatic segmentation and labeling of speech based on Hidden Markov Models". *Speech Communication*, pp. 357–370, 1993.

[Bur] J. Burdkart. "BRENT. Algorithms for Minimization Without Derivatives". `http://people.sc.fsu.edu/~jburkardt/m_src/brent/brent.html`. Last accessed on 16.01.2014.

[Bur05] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss. "A database of German emotional speech". In: *Proc. of Interspeech*, pp. 1517–1520, Lisbon, Portugal, 2005.

[Cen10] L. Cen, P. Chan, M. Dong, and H. Li. "Generating Emotional Speech from Neutral Speech". In: *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 383–386, 2010.

[Cor00] R. Cornelius. "Theoretical approaches to emotion". In: *ICSA workshop on speech and emotion*, pp. 3–10, Belfast, North Ireland, 2000.

[Des10] S. Desai, A. Black, B. Yegnanarayana, and K. Prahallad. "Spectral mapping using artificial neural networks for voice conversion". *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 18, No. 5, pp. 954–964, Jul. 2010.

[Dur60] N. Durbin. "The fitting of time series models". *Revue de l'Institut International de Statistique*, Vol. 28, No. 3, pp. 233–244, 1960.

[Eri05] D. Erickson. "Expressive speech: Production, perception and application to speech synthesis". *Acoustical Science and Technology*, Vol. 24, No. 4, pp. 317–325, 2005.

[Fel66] W. Feller. *An Introduction to Probability and Its Applications*, p. 166. John Wiley, 1966.

[Fuj05] H. Fujisaki, C. Wang, S. Ohno, and W. Gu. "Analysis and synthesis of fundamental frequency contours of standard Chinese using the command-response model". *Speech Communication*, Vol. 47, pp. 59–70, 2005.

[Hel07a] E. Helander and J. Nurminen. "On the importance of pure prosody in the perception of speaker identity". In: *Proc. of Interspeech*, pp. 2665–2668, 2007.

[Hel07b] E. Helander and J. Nurminen. "A novel method for prosody prediction in Voice Conversion". In: *IEEE Proceedings on Acoustics, Speech and Signal Processing. ICASSP 2007*, pp. 509–512, 2007.

[Hel12a] E. Helander. *Mapping techniques for Voice Conversion.* PhD thesis, Tampere University of Technology, 2012.

[Hel12b] E. Helander, H. Silen, T. Virtanen, and M. Gabbouj. "Voice Conversion Using Dynamic Kernel Partial Least Squares Regression". *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 3, March 2012.

[Hof04] G. O. Hofer. *Emotional Speech Synthesis.* Master's thesis, University of Edinburgh, 2004.

[Hua01] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing. A guide to Theory, Algorithm and System Development.* Prentice Hall, 2001.

[Ima83a] S. Ima. "Cepstral analysis synthesis on the mel frequency scale". In: *Proc. of ICCASP 83*, pp. 93–96, 1983.

[Ima83b] S. Imai, K. Sumita, and C. Furuichi. "Mel log spectrum approximation (MLSA) filter for speech synthesis". *Electronics and Communications in Japan Part I-communications*, Vol. 66, No. 2, pp. 10–18, 1983.

[Ina03] Z. Inanoglu. *Transforming Pitch in a Voice Conversion Framework.* Master's thesis, University of Cambridge, 2003.

[Ina07] Z. Inanoglu and S. Young. "A System for Transforming the Emotion in Speech: Combining Data-Driven Conversion Techniques for Prosody and Voice Quality". In: *Proc. of INTERSPEECH*, 2007.

[Jon93] S. de Jong. "SIMPLS: An alternative approach to partial least squares regression". *Chemometrics and Intelligent Laboratory Systems*, Vol. 18, No. 3, pp. 251–263, March 1993.

[Jou13] R. Jourani, K. Daoudi, R. André-Obrecht, and D. Aboutajdine. "Discriminative speaker recognition using large margin GMM". *Neural Computing and Applications*, Vol. 22, pp. 1329–1336, June 2013.

[Kai98] A. Kain and M. Bacon. "Spectral voice conversion for text-to-speech synthesis". In: *Proc. of ICASSP*, pp. 285–288, May 1998.

[Kam95] T. Kamm, G. Andreou, and J. Cohen. "Vocal tract normalization in speech recognition: Compensating for systematic speaker variability". In: *Proc. of the 15th Annual Speech Research Symposium*, 1995.

[Kan06] Y. Kang, J. Tao, and B. Xu. "Applying Pitch Target Model to Convert F0 Contour for Expressive Mandarin Speech Synthesis". In: *Proceeding of Acoustics, Speech and Signal Processing. ICASSP 2006*, pp. 733–736, 2006.

[Kaw97] H. Kawahara. "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited". In: *Proc. of ICASSP-97*, pp. 1303 – 1306, 1997.

[Kaw99] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné. "Restructuring speech representations using using a pitch-adaptive time-frequency smoothing and a instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds". 1999.

[Kor03] G. Korchanski and C. Shih. "Prosody modeling with soft templates". *Speech Communication*, Vol. 39, pp. 311–352, 2003.

[Lev47] N. Levinson. "The Wiener RMS error criterion in filter design and prediction". *Journal of Mathematics and Physics*, Vol. 25, pp. 261–278, 1947.

[Mar79] K. Mardia, J. Kent, and J. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[Mcd98] J. W. Mcdonough. "Speaker Normalization with All-Pass Transforms". In: *International Conf. on Spoken Language Processing'98*, 1998.

[Nar95] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana. "Transformation of formants for voice conversion using artificial neural networks". *Speech Communication*, Vol. 16, No. 2, pp. 207–216, Feb. 195.

[Nur12] J. Nurminen, E. Helander, V. Popa, and M. Gabbouj. *Speech Enhancement, Modeling and Recognition - Algorithms and Applications*, Chap. 5. Voice Conversion. InTech, 2012.

[Opp65] A. Oppenheim. *Superposition in a class of nonlinear systems*. PhD thesis, Res. Lab. Electronics, Massachusetts Institute of Technology, 1965.

[Ost02] J. Ostermann. *MPEG-4 Facial animation*, Chap. Face Animations in MPEG-4, pp. 17–56. John Wiley, 2002.

[Pro07] J. Proakis and D. Manolakis. *Digital signal processing*. Pearson Prentice Hall, 2007.

[Roa96] C. Roads. *The Computer Music Tutorial*. MIT Press, 1996.

[Ros01] R. Rosipal and L. J. Trejo. "Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space". *Journal of Machine Learning Research*, Vol. 2, pp. 97–123, 2001.

[Sak78] H. Sakoe and S. Chiba. "Dynamic programming optimization for spoken word recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 26, No. 1, Feb. 1978.

[Sal10] G. Salvi, F. Tesser, E. Zovato, and P. Cosi. "Cluster Analysis of Differential Spectral Envelopes on Emotional Speech". In: *Proc. of Interspeech 2010*, pp. 322–325, 2010.

[San14] G. Sanchez Gasulla. *Modeling and conversion of prosody using wavelets*. Master's thesis, Tampere University of Technology, 2014.

[Sch01] K. R. Scherer, R. Banse, and H. G. Wallbott. "Emotion inferences from vocal expression correlate across languages and culture". *Journal of cross-cultural psychology*, Vol. 32, No. 1, pp. 76–92, Jan. 2001.

[Sch03] K. R. Scherer. "Vocal communication of emotion: A review of research paradigms". *Speech Communication*, Vol. 40, No. 1-2, pp. 227–256, 2003.

[Sch95] H. Schmid. "TreeTagger - a language independent part-of-speech tagger". `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`, 1995. Last accesed on 11.02.2014.

[Sil11] H. Silen, E. Helander, and M. Gabbouj. "Prediction of voice aperiodicity based on spectral representations in HMM speech synthesis". In: *Proc. of Interspeech*, Florence, Italy, 2011.

[Son11] P. Song, Y. Bao, L. Zhao, and C. Zou. "Voice conversion using support vector regression". *Electronics Letters*, Vol. 47, No. 18, pp. 1045–1046, 2011.

[Soo84] F. Soong and B. Juang. "Line spectrum pair (LSP) and speech data compression". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84*, pp. 37–40, 1984.

[SPT] SPTK working group. "Speech Signal Processing Toolkit version 3.6". `http://sp-tk.sourceforge.net/`. Last accessed on 31.01.2014.

[Sty98] Y. Stylianou, O. Cappe, and E. Moulinés. "Continuous Probabilistic Transform for Voice Conversion". *IEEE Transactions on Audio and Speech Processing*, Vol. 6, No. 2, March 1998.

[Sun03] D. Sündermann and H. Ney. "VTLN-based cross language voice conversion". In: *Proc. of the ASRU*, pp. 676–681, 2003.

[Sun13] A. Suni, D. Aalto, T. Raitio, P. Alku, and M. Vainio. "Wavelets for intonation modeling in HMM speech synthesis". In: *Proc. 8th ISCA Speech Synthesis*, pp. 285–290, 2013.

[Tod07] T. Toda, A. Black, and K. Tokuda. "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory". *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 8, pp. 2222–2235, Nov. 2007.

[Tok94] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai. "Mel-generalized cepstral analysis - a unified approach to speech spectral estimation". In: *Proc. of ICSLP-94*, pp. 1043–1046, 1994.

[Weg96] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin. "Speaker Normalization on Conversational Telephone Speech". In: *Int. Conf. on Acoustic, Speech and Signal Processing*, Atlanta, GA, 1996.

[Wel99] L. Welling, S. Kanthak, and H. Ney. "Improved Methods for Vocal Tract Normalization". In: *Int. Conf. on Acoustic, Speech and Signal Processing*, pp. 761–764, Phoenix, AZ, 1999.

[Wik14] Wikipedia. "Human vocal apparatus used to produce speech". `http://en.wikipedia.org/wiki/File:Illu01_head_neck.jpg`, 2014. Last accessed on 12.02.2014.

[Wu06] C.-H. Wu, C.-C. Hsia, T.-H. Liu, and J.-F. Wang. "Voice Conversion Using Duration-Embedded Bi-HMMs for Expressive Speech Synthesis". In: *IEEE Transactions on Audio, Speech, and Language Processing*, July 2006.

[Xu01] Y. Xu and Q. E. Wang. "Pitch targets and their realization: Evidence from mandarin chinese". *Speech Communication*, Vol. 33, pp. 319–337, 2001.

Total number of entries is 66