Universidad del País Vasco
Euskal Herriko Unibertsitatea

APPLICATION OF SINGING SYNTHESIS TECHNIQUES TO BERTSOLARITZA

a dissertation presented

by

Xabier Sarasola Aramendia

Thesis director: Dr. Eva Navas Cordón

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

UPV/EHU

Bilbo

October 2020

Acknowledgments

I would like to thank the following people for helping with this research project:

- Eva nire tutoreari, laguntza eta pazientzia guztiarengatik. Itsasuntzi guztiek behar dute lema. Flotatzea eta aurrera egitea ez da inoiz nahikoa.

- Aholabeko taldekide guztiei. Ezinezkoa da neurtzea lan hau egin bitartean jasotako erronkak, laguntza eta babesa. Inma, meritoa dauka hau bezelako talde poliedrikoa bat gainbegiratzea eta zer esanik ez honekin helburu koerenteak lortzea. Eskerrik asko gidaritzarengatik. Jon, eskerrik asko unibertsitate irakaskuntzarako sarreran laguntzeagatik. Makinista izatea ez duzu oraindik lortu baina bips bezelako lurrun-makina bat nola zaintzen duzun ikusita argi dago arazorik ez zenuela izango eskala handiagoekin. Ibon, erronka dialektiko etengabea, eskerrik asko txartel grafikoen eta pythonen arora sartzerakoan pauso bakoitzaren zehaztasun zientifikoa exijitzeagatik. Igor, eskerrik asko hizketaren ezagutzarekin laguntzeagatik eta edozein gairi buruzko ikuspuntu kritikoa emateagatik.

  Daniel, tan excepcional en el conocimiento de la voz humana y tan cercano en el día a día. Gracias por la introducción y ayuda en mis inicios en la síntesis de voz. Agustín, gracias también por la ayuda con la síntesis de voz en mis inicios y los cursos acelerados sobre el idioma y cultura japonesa. No creo que mi trabajo haya conseguido la calidad de Hatsune Miku pero espero haberme acercado lo suficiente. David, gracias por haberme respondido

con elegancia tantas dudas y preguntas en aspectos que han ido desde lo estríctamente profesional a estadísticas tan polémicas como las asistencias post-rebote en el fútbol. Gracias también por implantar y haber sostenido la cultura de las palmeras. Las idiosincrasias se hacen, no se nacen. Luis, gracias por la amabilidad y ayuda desde el primer momento en el que llegué. Tu conocimiento sobre el laboratorio y todo lo que lo rodea se acerca a tu enciclopédico saber sobre el cómic. Ambas habilidades han sido esenciales durante estos años.

Sneha, thank you for introducing to our research group the amazing world of the human brain and your help with all my musical questions.

Itxasne, eskerrik asko laborategiari emandako gatz eta piperrarengatik eta laguntzarengatik. Palmeren kultura ere zurea da.

Victor, aunque hayamos compartido poco tiempo de laboratorio, te tengo que agradecer haber puesto encima de la mesa las tecnologías y desafíos que marcarán el futuro de la síntesis de voz.

Eder, ez dugu asko elkar topatu denbora honetan baina musikari buruz izandako galdera askoren erantzunak topatzen lagundu didazu. Eskerrik asko.

- Giorgos and Maria, for all the help and advise offered in that little paradise called Creta.

- Bertsozale Elkartea eta Xenpelar dokumentazio zentroa, eskerrik asko hainbat urtetan profesionaltasun handiaz jasotako datuak erabiltzen uzteagatik. Datu hauek izan dira lan honen subjektu material nagusia eta espero dut bertsolaritzaren etorkizunerako bide berriak irekitzen lagundu izana.

Laburpena

Tesi hau bertsolaritza kantu-ahots sintesi sistema berri baten garapenean zentratzen da, zuzeneko grabazioak oinarri gisa erabiliz. Lan honen erronka ez da soilik kantu-ahots sintesi sistema bat martxan jartzea. Bertsolaritza grabazioen corpusak bertso inprobisatuen transkripzioak ditu, baina grabazio artxiboek kantu-ahotsa ez diren hainbat elementu dituzte. Grabazioa gehienak zuzeneko saioak direnez, hizlarien ahotsa, jendearen txaloak eta zarata datu-basearen parte dira. Gainera, kantu-ahotsa ez dago etiketatua datu-basean. Ezaugarri hauek dituen datubase batekin, lan honen helburua bertsolaritza audioak garbitu, segmentatu eta etiketatzeko metodoak sortzea da, ondoren datu horiek kantu-ahotsaren sintesi ereduak sortzeko balio duten aztertzeko.

Grabazioetan kantu-ahots segmentuak automatikoki lortzeko metodoak garatu ditugu, hizketa eta kantu-ahotsa bereizten dituzten algoritmoak sortuz. Interbentzioen eta fonemen segmentazioa burutu dugu kantari anitzeko datubasean. Proposatutako segmentazio algoritmoek etorkizunean ager daitezken bertsolari berrien grabazioak segmentatzeko gaitasuna dute. Ondoren, bertsolaritza artearen propietate musikalak aztertu ditugu eta datu-baseko melodia teorikoak eta haien interpretazioa alderatu ditugu. Sistema automatikoak definitu ditugu bertsolarien kantu-ahotsa musikalki etiketatzeko. Etiketatze honek vibratoa kontuan hartzen du eta honen erabilera aztertu dugu bertsolari bakoitzean. Lortutako etiketatze sistema guztien ebaluazioa egin dugu prozesuan zehar.

Etiketatutako bertso grabazioen datu-basea sortu ondoren kantu-ahots sintesi sistemak sortu ditugu HMM eta DNN-ak erabiliz. Sistema hauei pitch nor-

malizazioa, tempo egokitzapena eta vibrato iragarpen teknikak gehitu dizkiegu. Metodo bat definitu dugu partiturak bertsolari bakoitzera automatikoki egokitzeko hauen pitch tartea kontuan hartuz. Bertsolari desberdinetarako sortutako sintesi ereduak modu subjektibo eta objektiboan ebaluatu ditugu emaitza onak lortuz.

Tesi honen ekarpenak bertsolaritza eta kantu-ahotsaren sintesiarekin erlazionatzen dira. Informazio maila berriak gehitu dizkiogu bertsolaritza corpusari, kantu-ahotsaren segmentazioarekin, interbentzioen segmentazi- oa, fonemen segmentazioa eta etiketatze musikalarekin. Etiketatzeko metodo hauek ez dute inolako kontrolik behar eta, beraz, etorkizunean datu-base etiketatuak handitzeko tresnak sortu ditugu. Kantari anitzeko datu-base bat sortu dugu, artearen egoerako beste kantu-ahots datubaseak baina handiago dena. Azkenik, bertsolaritza kantu-ahotsa sintetizatzeko sistemak definitu ditugu kantari eta teknologia desberdinekin, emaitza positiboak lortuz.

Abstract

This thesis focuses on the development of a new bertsolaritza singing voice synthesis system using as base original bertsolaritza live session recordings. The challenge of this work is not only the implementation of a singing voice synthesis system. The recorded corpus of bertsolaritza contains the transcriptions of improvised verses, but the audio files contain multiple elements that are not singing voice. As the majority of the recorded audios are live sessions, the voice of a speaker, applause of the public and noise are part of the database. In addition, the musical labeling of the singing voice is not included in the database. With a database of these properties, the aim of this work is to create methods to clean, segment and label the audios in the bertsolaritza and analyze the possibility of using them to create synthesis models for bertsolaritza singing voice synthesis.

We have developed methods to automatically obtain the singing voice segments in the recordings, creating new speech and singing voice classification algorithms. The segmentation of bertso utterances and phonemes has been performed in a multi-singer database. The segmentation algorithms proposed have the capacity to align material from unseen bertsolaris in the future. After that, we analyzed the musical properties of the bertsolaritza art and compared the theoretical melodies in the database with the actual interpretation of them. We defined automatic systems to musically label the bertsolaritza singing voice generating a fully labeled bertsolaritza database. Musical labeling included vibrato and we analyzed the use of it in each bertsolari. We evaluated all automatic labeling systems in the process.

After creating a labeled database of bertso recordings we generated singing voice

synthesis systems using HMMs and DNNs. We included f_0 normalization, tempo adaptation and vibrato prediction techniques in these systems. We defined methods to automatically adapt music scores for each bertsolari considering the pitch range of each bertsolari. We evaluated synthesis models created for different bertsolaris in a subjective and objective way obtaining good results.

The contributions of this thesis are related to bertsolaritza and singing voice synthesis. We added new information levels to the bertsolaritza corpus with the segmentation of singing voice, the alignment of utterances and phonemes and the subsequent musical labeling. These labeling methods need no manual supervision and therefore we created tools to increase the labeled database in the future. We created a multi-singer singing voice database that is considerably bigger than any state of the art singing voice databases. Finally we defined systems to synthesize bertsolaritza singing voice using different singers and technologies obtaining positive results.

Contents

Listing of figures

Listing of tables

1

Introduction

Bertsolaritza is an art of improvised poetry singing that is very popular in the Basque Country. This art is always performed in Basque and the recordings and documentation of the live sessions since the 60s are considered an important literary and musical corpus of Basque language and the Basque culture. The combination of longevity and non literacy of the Basque language has made the Basque popular literature mainly oral and bertsolarism can be considered a sub-genre of it [50]. In this thesis we will use the terminology in Basque language to address different elements of bertsolaritza. These are the definitions of these terms:

- **Bertso**: A bertso is a strophe of the improvised lyrics.
- **Bertsolari**: The name of the improviser and singer of the bertsos.

Given the importance of this art in the Basque society, multiple research areas have been opened related to it. Literary [49][52], sociological [117], musical

[54][55], Natural Language Processing (NLP) [1][7] and robotics [6] related researches have been developed in the analysis of bertsolaritza. Considering the growth of voice synthesis in the 21st century and specially of singing voice synthesis in the decade of 2010s, the development of bertsolaritza singing voice synthesis became a new research option.

The increase of digital data and data driven methods in the 20th century brought new ways of understanding the singing voice synthesis research. The highest quality systems and commercial systems like Vocaloid [72] are based in concatenative synthesis systems but the synthesis systems based on statistical models have been improving their results since their first appearance in 2006 [123].

The first bertsolaritza voice synthesis system was developed in 2015 to add singing voice to Bertsobot, the Bertsolari robot [6]. That singing voice was created by speech to singing voice conversion, modifying the parameters generated by an speech synthesis system. The research on bertsolaritza singing voice created the need of a deeper analysis of this art and the next steps have been taken in this thesis. This work analyzes the recordings of a bertsolaritza corpus and the potential applications of this corpus in the singing voice synthesis research. For this purpose we have used state of the art technologies. The analysis of bertso recordings from a signal processing perspective and the creation of a singing voice synthesis system based on these recordings are research paths that can bring a whole new possibility both to Basque society and singing voice research. This chapter is a summary of the work and contains the motivation, research lines and objectives of the work. The last part of this chapter explains the structure of the thesis document and the content of each section.

1.1 Motivation

The motivation of this work can be explained as the convergence of three main concepts: the bertsolaritza art from the Basque Country, the singing voice synthesis research and the trajectory of Aholab research group. These three elements

converge in the need of creating new tools and analysis methods.

1.1.1 Bertsolaritza

Bertsolaritza is an improvisation poetry art from the Basque Country. With the first examples located at the end of the 18th century [81], this art is strongly rooted in the Basque culture and with the increase of recorded and transcribed material, in the 20th century it became a very important element in the Basque literature and culture. Performed in cider houses, bars, squares and stadiums this popular art achieved a professional status since the 80s and is taught to the kids since early ages. With the popularization of recording tools in the 60s the recordings of bertsos started to increase and the organization Xenpelar Dokumentazio Zentroa was created to save all the data related to bertsolaritza: music scores, transcriptions, recordings and the related metadata. The growth of the newly created corpus pushed the emergence of new research areas such as its sociocultural impact [117], automatic poetry generation [1], musical analysis [54][55] and literary style analysis [49] [52] among others. In the area of signal processing research applied to bertsolaritza, a singing voice generation system had been created before the beginning of this thesis, but without analyzing the recordings of original bertso sessions.

1.1.2 Singing voice synthesis

Starting from its first public exposition in 1961 as marketing stunt of Bell laboratories, the singing voice synthesis evolved in the second half of the 20th century in parallel with the digital signal processing and data science. During the 20th century, rule based methods [167][147] made big improvements and many researchers tried to analyze all the phenomena that happen in singing voice. At the end of the 20th century concatenative methods started to be applied in singing voice synthesis [89]. In the 21st century, data driven approaches started to improve results and recent singing voice synthesis research has been focused in HMM [123][139] and Neural Network [103][16] approaches. Although the direction

of research is on data driven approaches, nowadays concatenative singing voice synthesis systems are the ones with best results and the ones used in commercial systems [72].

1.1.3 Aholab signal processing laboratory

Aholab is a signal processing laboratory from the University of the Basque Country specialized in speech processing created in 1992. Aholab developed the first Basque Text-to-speech (TTS) system [118][62] and it is the reference research group in terms of speech processing for Basque language. Aholab took part in the project of bertso singer robot Bertsobot [6]. In this project Aholab created the singing voice of the robot, by adapting a speech synthesis system.

1.1.4 Convergence of paths

Considering the relevance of bertsolaritza for the Basque Country and Basque language, any research line about this art has scientific and cultural interest. The data compiled by the Xenpelar Dokumentazio Zentroa includes hours of recordings of this art and before this thesis, this data had not been analyzed from a signal processing point of view. Aholab research group has experience analyzing Basque speech and extending the research area of Basque language from speech to singing voice would open a whole new research area around a topic with growing interest in the signal processing field. Although the starting motivation of this work is an improvement of bertsolaritza singing voice synthesis, defining the technology and standards for the laboratory in singing voice would create possibilities for recording analysis, score prediction, singing voice synthesis and conversion.

1.2 Research overview

If we analyze the structure of a generic singing voice synthesis system in Figure 1.2.1, we can see that musical scores and their respective interpretations of singing

voice are needed to create synthesis models.

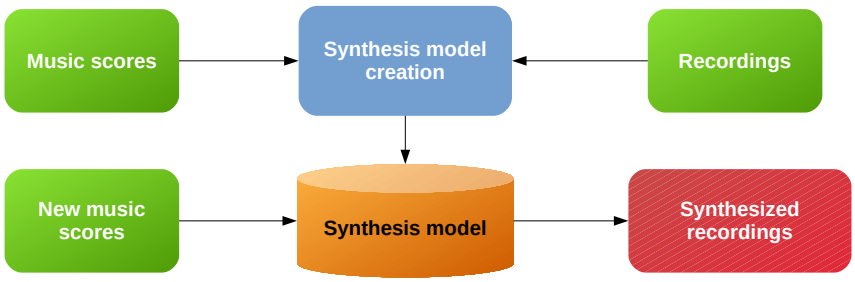


Figure 1.2.1: General architecture of a singing voice synthesis system

In the case of bertsolaritza, the audio files are recordings of live sessions, therefore, together with the singing voice, they include speech from the host who sets the improvisation themes, applause of the public and other extra noises. A preprocessing of the audio files needs to be done to obtain only the singing voice segments that will be used to create synthesis models. The data collected by Xenpelar Dokumentazio Zentroa include the orthographic transcriptions of the sung parts. However, due to the improvised nature of bertsolaritza, there is no music score with the musical information. The improvised verses are conditioned to specific meters and usually bertsolaris use predefined melodies for each meter. Nevertheless, because of the improvisation and also because bertsolaris are not professional singers, every melody can vary even when maintaining the meter. Meter and melody are the metadata parameters directly related with the musical structures and are incomplete in the provided archives. Taking into account that every recording has orthographic transcriptions and there are missing meter and melody annotations, we decided to create methods to predict these missing metadata features. The whole research performed in this thesis can be divided in three different phases:

- **Metadata prediction**: Meter and melody are the most relevant metadata

related to musical information and they are crucial to build the music scores used as input in singing voice synthesis systems. As they are missing in some files, new methods for predicting meter and melody must be created and tested.

- **Music score creation**: With the complete metadata obtained in the previous phase, novel methods to define the music scores corresponding to the recordings will be developed and evaluated.

- **Bertsolari voice synthesis**: Using the music score created in the previous phases, different singing voice synthesis models will be built and their quality will be compared and assessed using objective and subjective measures.

The complete process is represented in Figure 1.2.2.

1.3 Objectives

With the defined research structure, the objectives of this thesis are the next ones:

- **Singing voice synthesis research**: Collect information about the historical trajectory of singing voice synthesis and about the main state of the art technologies to select the best technologies to create a bertso singing voice synthesis.

- **Data collection**: Collect all the singing voice databases available to test and compare different singing voice processing methods that will be developed in this work.

- **Metadata prediction**: Create classification systems for the metadata of bertso recordings with transcriptions in order to make metadata annotation in recordings with missing information easier and to automatically annotate future recordings. Test the systems and evaluate their reliability for future improvements.

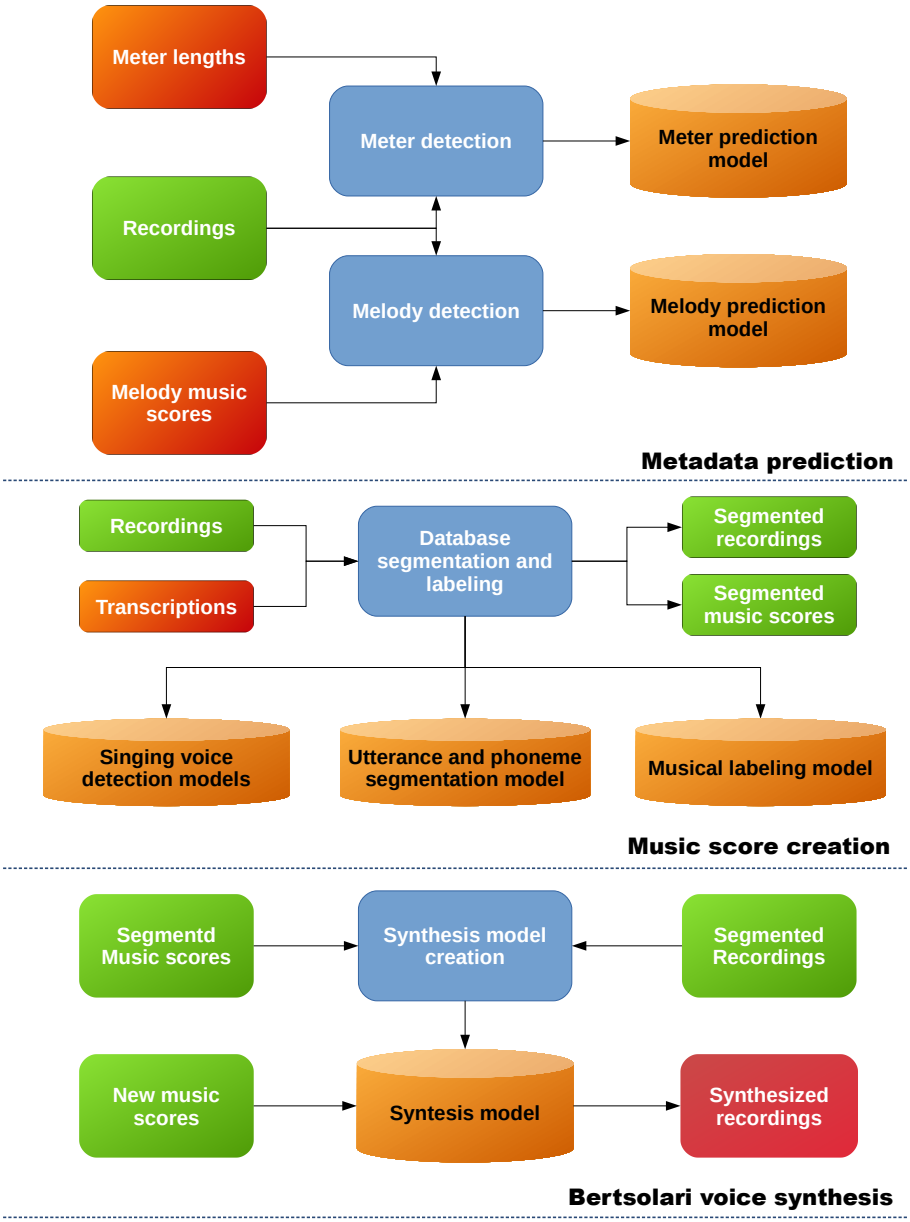


Figure 1.2.2: Main tasks to be covered by the thesis work

- **Database standardization**: Define the standard formats and type of files for the diverse information contained in the Bertso database considering the differences with a speech database.
- **Bertso melody evaluation**: Evaluate the musical coherence of the bertso recordings with standard melodies.
- **Music score prediction**: Using the bertso recordings, the orthographic transcriptions and the bertso melodies, create methods to define the music score represented by the singing voice in the recording.
- **Creation of an annotated bertso singing voice database**: Combining the manual and automatic methods proposed in this work define a fully annotated bertso singing voice database.
- **Music score synthesis standardization**: Define methods to standardize music scores for different singers with different data distributions.
- **Bertsolari voice synthesis**: Create singing voice synthesis models for multiple singers and compare them by means of objective and subjective evaluations.

1.4 Thesis structure

The structure of this work is designed to achieve every goal of the thesis. In **Chapter 2** singing voice generation, analysis, synthesis and labeling methods are described along with a general description of bertsolaritza art. History and analysis methods of speech are included in this chapter because multiple elements are shared with singing voice. First, singing voice generation by humans is explained in Section 2.1 and the representation of music in music scores is analyzed in Section 2.2. Then, we summarize the history of speech synthesis before the appearance of the singing voice synthesis in Section 2.3. In Section 2.4 we analyze different methods of modeling singing voice and in Section 2.5 we explain the spectral

methods used in the second half of the 20th century for the singing voice analysis and synthesis. Then, different singing voice synthesis system types are explained in Section 2.6. Next, different methods for automatic singing voice labeling are described in Section 2.7. Finally, the history and characteristics of bertsolaritza art are described in Section 2.8. This chapter meets the goal of analyzing the needs of bertsolaritza art and the optimal technologies to achieve the objectives in the current context.

In **Chapter 3**, the available databases of singing voice are described in Section 3.1 along with standard files and formats in singing voice databases. In this section transcriptions are properly cleaned and edited to adjust them to the recordings. The edited transcriptions are used to evaluate meter prediction systems as part of the goal of creating a metadata prediction system. In Section 3.2, the state of the art about the available software that can be valuable to perform the work proposed in the thesis is defined.

In **Chapter 4**, we propose new methods to define the music score representation of the bertso recordings. We start with the separation of singing voice from the rest of audio events present in the recordings. This process is described in Section 4.1. Then, utterance and phoneme segmentation are applied and described in Section 4.2 and Section 4.3 respectively. In Section 4.4 we define different ways to musically represent the singing voice in amateur recordings without music scores and describe the results of testing the annotation system in other databases. In Section 4.5 we analyze the music scores created by the method proposed in Section 4.4. The range, year of the recordings, note durations and the use of vibrato are analyzed in different bertsolaris. We also analyze the singing voice ranges of the singers of National University of Singapore (NUS) database.

In **Chapter 5** we use the data and annotations created in Chapter 4 to model different singer voices with multiple singing voice synthesis systems. In Section 5.1 we define the general scheme of the synthesis systems and in Section 5.2 we propose adaptation techniques for note pitch and duration to obtain synthesis models with more flexibility and better quality. In Section 5.3 we define the reconstruction of the vibrato from the parameters predicted with the Deep Neural Network

(DNN) system. Next, in Section 5.4, we define the model dependent conversion of the music scores used to adapt music scores to the range of each singer in an automatic way. After this, we describe the representation of the audio recordings and music scores that we use to create singing voice synthesis models in Section 5.5. Next, we explain the HMM-based and DNN-based singing voice synthesis systems in Section 5.6 and Section 5.7. In Section 5.8, we assess the synthesis systems by evaluating synthesized samples both with objective and subjective measures.

In **Chapter 6** we describe the conclusions of this thesis. The contributions and obtained goals are listed in Section 6.1. In Section 6.2 we define the possible improvements of the synthesis systems created in this work and the technologies that could be used to achieve these improvements.

2

State of the art

In this chapter we describe the previous research related to different aspects of the singing voice, the research area related with this thesis. We also describe the history and characteristics of bertsolaritza art. The chapter starts explaining the singing voice generation by humans in Section 2.1 and the representation of music in music scores in Section 2.2. Then, we summarize the history of speech synthesis before the appearance of the singing voice synthesis in Section 2.3 given the close relation between these two technologies. In Section 2.4 we analyze different methods of modeling singing voice and in Section 2.5 we explain the spectral methods used for the singing voice analysis and synthesis. Then, different singing voice synthesis system types are described in Section 2.6. Next, different methods for automatic singing voice labeling are summarized in Section 2.7. Finally, the bertsolaritza art and a summary of research lines related to it are detailed in Section 2.8.

2.1 Singing voice

Singing voice is the generation of musical notes using the human vocal apparatus. Considered as one of the oldest musical instruments ever used, the cultural and social presence of the singing voice in our societies is undeniable. The high expressiveness of the singing voice combined with lyric poetry makes it a perfect tool to define cultural identities and idiosyncrasies. Singing voice has also been an oral method of music tradition transmission for many cultures.

The generation of singing voice is similar to that of speech. In Figure 2.1.1 we can see the structure of the human voice generation system.

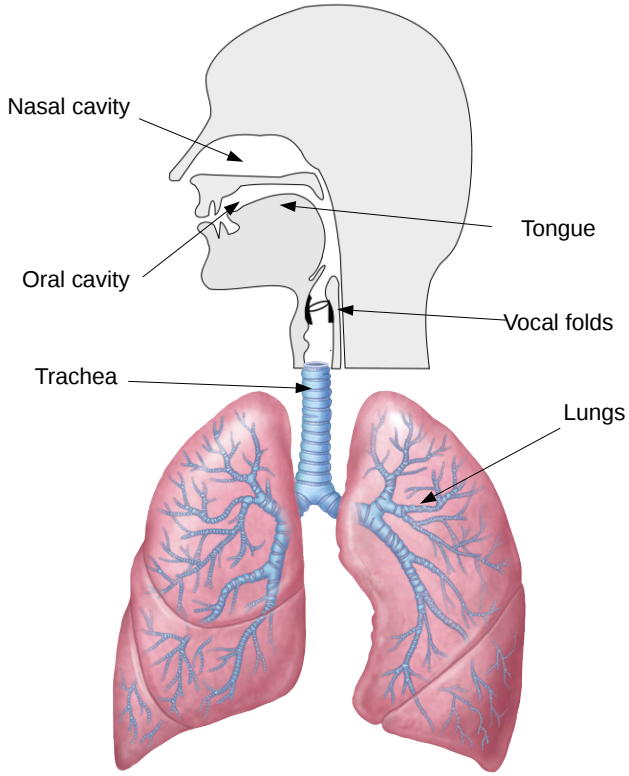


Figure 2.1.1: Structure of the human voice generation system

All the sounds in human speech or singing voice are generated with this system, therefore the flexibility and versatility of the system is very high. This versatility makes the singing voice one of the most important elements in music perception of humans [32]. In the voice generation process, the lungs create air pressure waves that, after crossing the trachea, arrive to the vocal folds. The vocal folds are a set of muscles that can vibrate under this pressure. The speed of this vibration (i.e. the number of times per second that the vocal folds open and close under the air pressure wave) can be considered the fundamental frequency of the speech or singing voice. The term pitch is used to refer to the perceived frequency instead of the actual frequency that has been generated by the vocal folds. This pitch frequency controls the intonation in speech or the singing pitch in singing voice. Pitch is related to the length of the vocal folds and we can control this length in a certain range around their natural length. The length of the vocal folds is related to the size of the neck, therefore there is a variety of pitch ranges to sing and speak. Longer folds produce a voice of a lower pitch. The range of pitch that a human can generate can also vary and can be trained to sing higher range songs. An image of the vocal folds can be seen in Figure 2.1.2.

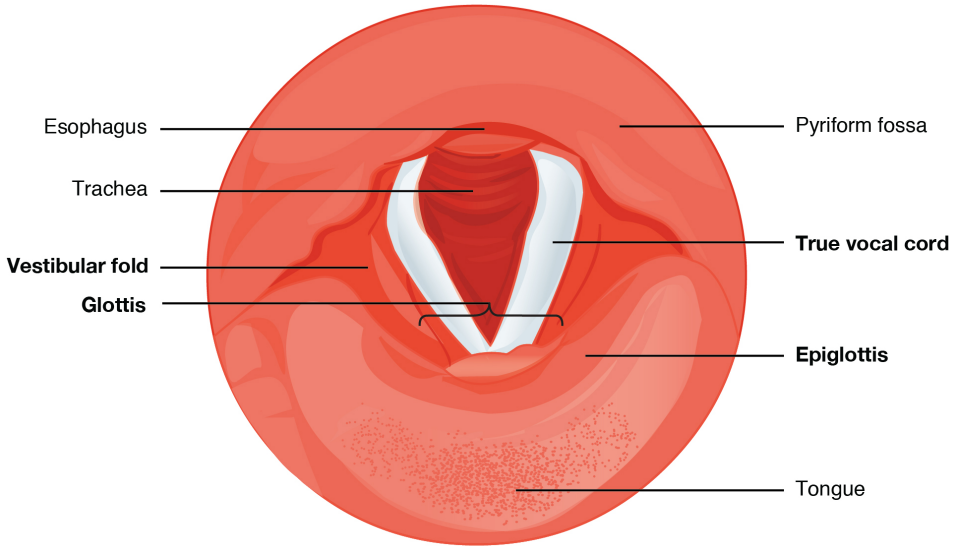


Figure 2.1.2: Representation of the vocal folds

When the vocal folds are used to create a sound, the generated sounds are defined as voiced. But unvoiced sounds can be generated too with the human voice generation system. In unvoiced sounds, the vocal folds do not vibrate in a harmonic way and the resulting signal is considered as noisy. In both voiced and unvoiced sounds, the resulting signal is defined as glottal voice source and it then passes through the nasal and oral cavities. With the control of the nasal and oral cavities, different transformations can be applied to the glottal voice source creating different sounds.

Multiple models tried to reproduce the human voice generation process through history but the most popular nowadays is the source-filter model. This model represents speech as the result of a source signal that passes through a linear acoustic filter. The glottal voice source is considered as the source and the nasal and oral cavities are considered as a linear filter. The generation of a voiced sound with the source-filter model is represented in Figure 2.1.3. The glottal voice source (source) is modeled as a train of pulses when the sound is voiced (as shown in the image) and as white noise in unvoiced sounds. The glottal source passes through the oral and nasal cavities (acting as filter) before reaching the output. The oral and nasal cavities create acoustic resonances at certain frequency bands, called 'formants'. They are seen in the spectrum as prominent frequency bands. Vowels are characterized by having stable formants and can be identified with only two of them. Although these formants are not identical for each realization, mainly because the position of the formants can vary depending on the surrounding phones, humans are capable of identifying them. For unvoiced sounds, white noise goes through the vocal-tract and generates non periodic turbulent noises. Although these turbulent noises are not periodic, humans can define different phonemes creating different noises.

In the source-filter model the source signal, represented as glottal voice source, is convolved with the impulse response of the filter, represented as the vocal and nasal tract. In Figure 2.1.3 we can observe the frequency response of the vocal and nasal tract and its resonance frequencies around 650, 1200 and 2300 Hz. These peaks are considered the formants. In the source-filter model, the generation of the

sound can be represented as the multiplication of the frequency representation of the source and impulse response of the filter in frequency space. In the unvoiced sounds white noise passes through the vocal tract generating different turbulent noises that the human hear can identify.

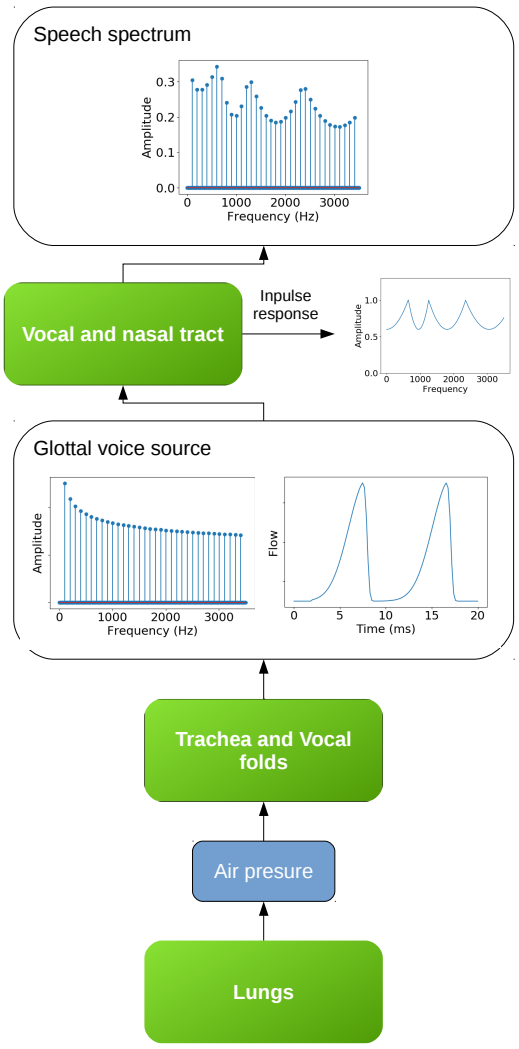


Figure 2.1.3: Voice production process from the signal processing point of view

With the source-filter model we can distinguish three different parameters: power, pitch and timbre. The power is related to the power envelope of the waveform, the pitch is related to the glottal source periodicity and the timbre relates to the form of the oral cavity.

The speech and singing voice share the same generation system and both usually express linguistic phonemes. Nevertheless, for humans speech and singing voice are two different types of sound and are used in different contexts. At the acoustic level, the differences between speech and singing voice are the next ones:

- **Voiced/unvoiced ratio**: In singing voice usually each note defines the average pitch and duration of a syllable of the lyrics. The pitch of a note is defined using the fundamental frequency and therefore is closely related to the voiced phonemes. The duration of the notes is not always longer than the duration of the syllables in speech, but this is commonly the case. Combining the importance of voiced phonemes for the pitch and the increase of syllable durations, the ratio of the duration of voiced to the unvoiced segments is higher in the singing voice comparing to speech.

- **Vocal Dynamics**: In musical terms, vocal dynamics is used to define the volume or power range of the voice. The singing voice tends to be more expressive and therefore the power of the singing voice has bigger range than that of speech. The average power of the singing voice is also higher than that of speech.

- **Range of fundamental frequency**: In the singing voice the fundamental frequency is driven by the notes in a music score. This makes the range of the fundamental frequency in the singing voice dependent on the music scores. If we consider the range needed to sing common music scores, this range is usually greater than the range needed for speech. The range of the fundamental frequency in speech goes from 80 Hz to 400 Hz and although it is complicated to define the range of all the possible sung melodies, we can put the example that the range of a soprano singer can go from 250 Hz to 1,500 Hz.

- **Vibrato**: In singing voice, sometimes a periodic modulation of the fundamental frequency is used to add expressiveness to notes with long duration. This modulation is called vibrato and it is a phenomenon that does not appear in speech.

- **Formants**: The purpose of speech and singing voice are different. The purpose of speech is the communication of a message and the singing voice purpose is closer to aesthetics and expression. In the transmission of a message the definition of the phonemes is a crucial part because these are the base of our acoustic linguistic channel. The definition of these phonemes is related with the formants as explained earlier in this section. The singing voice tends to give more importance to the expressiveness and the interpretation of the notes instead of to the understanding of the phonemes. These priorities sometimes change the formant positions of the voiced phonemes in singing voice. Opera singing voice also has an extra formant around 3 kHz that does not appear in speech [146]. This formant is closely related to the higher vocal loudness of this singing style.

If we consider the modeling of the singing voice from a source-filter perspective, four main elements can be predicted from a music score: phoneme durations, energy, timbre and f_0. Many analyses have been realized to determine energy contours [124] and the delay of the phonemes from the notes in the music score [123]. According to Saino et al. [123], the synthesis of singing voice sounds unnatural if we use the note durations to strictly calculate phoneme durations. This happens because there are time-lags between start timings of notes and real singing voice phonemes. Research about the timbre in singing voice has also been developed. The differences of timbre between different singing styles has been analyzed by Thalén and Sundberg [149] and singer formants have been analyzed by Sundberg [142]. The modeling of the f_0 is one of the key parts of the singing voice modeling, and this is why we explain it in a deeper way in the next Section 2.1.1.

2.1.1 Singing voice f_0

Compared to the other elements of the singing voice, f_0 has attracted special attention in the research of the singing voice synthesis. The reason for this attention is the importance of the f_0 in the musical melody and the expressiveness of the singing voice. The f_0 of the singing voice must represent a melody in the correct way, fill multiple mechanical conditions related to sung phonemes and add multiple expressive elements that are not included in music scores like vibrato for instance. It has been proved that changes in f_0 have greater effect in the perception of the quality of the singing voice if we compare it with changes in spectral characteristics [126].

The main elements in the singing voice f_0 where researchers have focused their efforts are the next ones:

- **Note onset positions**: The note onset position is placed in the first vowel onset inside the note in rule based systems like that of Kungliga Tekniska Högskolan (KTH) [147] and in expressive synthesis systems like the one described in [125].

- **Small fluctuations**: The small fluctuations in f_0 have been proved to be important for the naturalness of synthetic speech by Akagi et al. [4] and these fluctuations have been measured also in singing voice synthesis and been defined as fine fluctuations [3]. Saitou et al. created a singing f_0 synthesis method with second order filtering of the staggered curve of the pitch values that included fluctuations with white noise [127]. Ohishi et al. used a white Gaussian noise in their singing f_0 synthesis system [106].

- **Microprosody**: The deviation from the melody present in the f_0 curve because of the pronounced phonemes is called microprosody. Consonants tend to create f_0 valleys that complicate the relationship between the notes in the music score and the sung pitch values. Saino et al. [124] used aligned phonemes and interpolation to separate the phoneme dependent variations

and the melodic f_0. According to Saino et al. the voiced f_0 segments inside consonants suffer deviations from the melody. Removing this deviation by interpolation creates smoother f_0 curves that are easier to model considering the sung melody.

- **Transitions between notes**: If we consider the note transitions, Mori et al. observed that after the transitions, the f_0 values tend to surpass the destination pitch in the direction of the transition [101]. They defined this excess as overshoot extent. One year later Saitou et al. defined that before the transitions there was also a deflection in the opposite direction of the note change and called it preparation [127]. Saitou et al. created a f_0 generation model taking into account the preparation and overshoot, processing a staggered melody curve with a second-order system.

- **Vibrato**: Vibrato is defined as a periodic, rather sinusoidal, modulation of the f_0 curve. The modulation rate is usually between 5 and 8 Hz and the depth can vary between different singing styles. The depth needs a minimum of $\pm$ 0.5 semitones to be noticed and vibratos that exceed the $\pm$ 2 semitones are perceived as a bad singing technique [145]. The smoothness of the f_0 curve that statistical prediction methods assume makes the vibrato modeling a complicate task. This is why it has been modeled in a separate way in multiple statistical parametric singing voice synthesis methods [127][108][63].

- **Portamento**: The portamento is a pitch slide between two notes that is realized slower than a normal transition [134]. The portamento can be addressed in music scores, but the singers can also use it as an expressive mechanism without being signaled in the music score. This expressive mechanism has been taken into account in multiple singing voice synthesis systems [106][158] and note transcription systems [76][98][152].

2.1.1.1 Modeling strategies

Many of the research proposals in singing voice synthesis respond to the aforementioned phenomena. In 2005, Saitou et al. proposed a dynamic second-order system to process an initial staggered curve of the pitch values with defined durations [127]. This system took into account the generation of preparation, overshoot, fine-fluctuations and vibrato. In 2010 Saino et al. proposed the separation of phoneme-dependent components and melody components from the f_0 curve. The system modeled the notes from the melody component with different HMM configurations for different note structures. To address the vibrato problem, they created a method to remove the vibrato from the melody component, model it separately and reconstruct it in synthesis [125]. In 2013 Umbert et al. used unit selection combined with transformation and concatenation processes to obtain natural and smooth f_0 curves [157]. Each unit contained the f_0 curve segment inside three consecutive notes or silence. The synthesized music score defines an ideal contour where each note has specific conditions of pitch and vocal dynamics. With the pitch and vocal dynamics conditions, the units are selected to fill the constraints on each note. After the selection of the desired note units, vibratos are removed and saved using rate, depth and reconstruction error contours. The selected units are transformed to adjust them to the conditions and crossfading is used to smooth the transitions. After crossfading the vibrato is reconstructed using saved parameters. Also in 2015, Ardillon et al. used B-spline curves to create melodic, microprosody, vibrato and fine-fluctuations [5]

In the last decade, neural network based strategies have been proposed to synthesize the melody. In 2015 Özer modeled the f_0 and vibrato with separate Long Short-Term Memorys (LSTM) using as input a time aligned staggered pitch curve and information about the pitch and duration of the current and contiguous notes [110]. In 2016 Nishimura et al. applied f_0 normalization combined with DNNs and discussed different interpolations procedures for the normalization of unvoiced phonemes and silence frames. After the success of Wavenet autoregressive structures to predict directly the audio waveform from fixed parameters, two dif-

ferent f_0 prediction models employed this structure, Blaauw and Bonada in 2017 [15] and Wada et al. in 2018 [158]. Blaauw and Bonada used one-hot encoding of phonemes and notes as conditional features and Wada et al. used features related to the melody and out of tune penalizing loss. In 2018 Hua proposed parametrical note models with separated modeling of transitions and vibrato using DNNs [68].

As it can be seen, intensive research has been developed around the synthesis of melody, which demonstrates the importance of this component in the singing voice.

2.2 Musical scores

Musical scores are the written interpretation of music and they can represent musical melodies with and without lyrics. The music score without lyrics are prepared for musical instruments and when lyrics are included the singing voice is needed to interpret it. In the music scores with lyrics, the text from the lyrics is split between the notes in the melody. This split usually separates syllables and each note contains the orthographic symbols of one syllable. If we consider that any orthographic representation can be converted to a phonetic representation, we can consider a music score as a sequence of notes in which each note contains one or more phonemes. Each note defines the average fundamental frequency and the duration of a set of phonemes and these sets of phonemes are concatenated generating the song. The silence is also a possible note symbol in music scores. Although music is a very expressive art and can vary from literal interpretations of the scores, any melody base can be written in a score. The Western musical notation system offers many tools to represent musical events in a song. There are many specific symbols, but we are going to define only the ones common in our bertso music scores. Examples of these elements in a music sheet can be seen in Figure 2.2.1.

- **Staves**. The staves represent relative pitch positions. Notes are positioned in these levels to represent a specific pitch level.

- **Clef**. The clef indicates the absolute position of the staves in a Western music note/octave scale.
- **Accidental**. Symbols preceding a note that lowers or raises a note in a semitone resulting in a "non natural" pitch. The most common ones in Western music are the sharp (♯), flat (♭), and natural (♮) symbols.
- **Bar line**. Vertical lines in staves signaling melodic stress.
- **Time signature**. Symbol indicating the duration of notes between two bar lines and stress patterns in that space.
- **Note**. Base musical unit. It defines a pitch position and a duration symbol. These two parameters are a relative value in a musical scale.
- **Rest**. Note equivalent to silence. It has no pitch and it is a representation of a duration symbol.
- **Slur**. Connection of two or more notes of different pitch position that means they have to be played without separation.
- **Tie**. Connection of two or more notes with the same pitch position that means they have to be played as a single note summing the duration of all of them.
- **Tempo**. Representation of the number of times a specific duration symbol can fit in a minute.
- **Key signature**. A symbol written after the clef that represents a shift in the representation of the notes in the stave.
- **Lyrics**. Orthographic representation of the phonemes that have to be interpreted in each note.

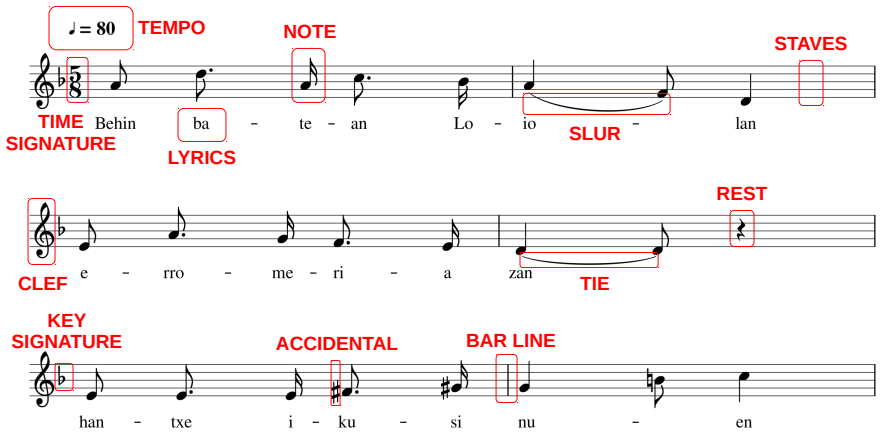


Figure 2.2.1: Example of elements considered in music notation

With these elements we can define the phonemes, the identity in note/octave scale and the duration of each note. These notes can be simplified as a sequence of pitch values and silences of specific duration. If the representation was ideal, we would obtain the sequence of notes where the phonemes within each note would fulfill the pitch and duration conditions of the note. The pitch should be represented with the f_0 of voiced phonemes and the sum of the durations of the phonemes should be equivalent to the note duration. This would result in a voice with a staggered f_0 representing the notes. Nevertheless, we explained in Section 2.1 that the singing voice is a very expressive human tool and the expressiveness comes precisely from the deviations from the music score that humans realize in the performance. Although the deviation from the music score creates expressive singing voice, deviating too much from the melody can result in bad quality singing voice. The melody of the music score has to be recognizable and for this purpose it is essential to perform sustained f_0 segments in the pitch values signaled in the music score. The durations of the phonemes are also modified from the durations signaled in the music score but the note transitions must respect the tempo of the music score. In conclusion, the duration and pitch conditions from the music scores are not strictly followed in the singing voice performances, but these conditions

are essential for the performance.

To obtain the phonemes that correspond to each note we have to convert the orthographic representation of the lyrics into phonemes. This conversion depends on the language of the lyrics. Each language has different phonetic rules and normally these rules are context dependent. To be context dependent means that the same orthographic symbol has different phonetic representation depending of the contiguous symbols. This is why it is essential to join the split lyrics in the music score to obtain the phonetic representation. We can see in Figure 2.2.1 that phrases and words are split in different notes making it impossible to obtain the phonetic transcription for each note in a independent way. In this thesis we used the phonetic rules of Basque and English because we have used databases with lyrics in both languages.

The pitch and duration values of the notes are represented in music scores in a ”relative” way and can be translated to real frequency (Hz) and time (s) values using standard references or references signaled in the music score. In the next two sections we will explain the process we need to follow to get the pitch and duration values of the notes from the score. Although these may seem to be standard known processes, the explanation will help us to show the problems that will appear when trying to label the bertso recordings.

2.2.1 Frequency of a note

The real frequency of a note can be obtained from the corresponding note and octave. These two parameters can be calculated using:

- Note position in the staff.
- Clef.
- Accidental.
- Key signature.

Using these elements the note and octave of a note symbol can be obtained from the music score and we can also obtain the Musical Instrument Digital Interface (MIDI) number of the note. The MIDI representation of a note is a technical representation of musical notes created in the 1980s for electronic music [69]. This scale is a standard way to represent all notes between C_{-1} to G_9 using integers from 0 to 127. This scale allows simplifying the representation of each note using only one integer instead of two (octave and note). We use Equation 2.1 for the conversion from note and octave to MIDI number

$$m = (12k + n + 12) \tag{2.1}$$

where n and k are the note and octave of the note we want to calculate respectively and m is the corresponding MIDI number. In the MIDI scale, the octave value range goes from -1 to 9 and the note value range goes from 0 to 11. With the MIDI number we can calculate the physical frequency of the note using Equation 2.2.

$$f(n, o) = 2^{(m-m_{ref})/12} f_{ref} \tag{2.2}$$

where m_{ref} and f_{ref} are the MIDI number and frequency of the note we selected as reference for the scale. According to this equation, the relation between note and frequency is always dependent on the note defined as reference, mostly known as tuning pitch. The standard in Western music for this reference note is the ninth note of the 4th octave, defined as A_4, and the frequency assigned to this note is 440 Hz. Applying the standard tuning system to Equation 2.2 produces Equation 2.3, which is the one we have applied in this thesis to calculate the frequency of any note in a music score.

$$f(n, k) = 2^{((12k+n+12)-69)/12} 440 \tag{2.3}$$

2.2.2 Duration of a note

The duration of a note is calculated using the symbol of the note and the tempo of the music score. The note symbols have no standard numerical representation, therefore we defined a numerical pairing scale that we have applied through the thesis. Although more symbols exist, we defined only the ones that usually appear in musical scores. We also considered only one dot symbol, although more than one can be used. This is again because it is very rare to see more than one dot in a music score. The symbol names and relative duration values are explained in Table 2.2.1.

American name	British name	Quarter value	Dotted value
whole note	semibreve	4	4 + 2
half note	minim	2	2 + 1
quarter note	crotchet	1	1 + 1/2
eighth note	quaver	1/2	1/2 + 1/4
sixteenth note	semiquaver	1/4	1/4 + 1/8
thirty-second note	demisemiquaver	1/8	1/8 + 1/16
sixty-fourth note	hemidemisemiquaver	1/16	1/16 + 1/32

Table 2.2.1: Note symbol names and relative duration values

As it can be seen in the table, the duration values of the notes in music scores are also relative temporal values represented in quarters. Although these are the most common used symbols in a music score, it is very usual to use ties and sum these symbols to obtain new duration values for the notes. This is why we will use the duration in quarters as a single number to represent the relative duration of a note. The real duration in seconds of a note symbol can be obtained with the formula defined in Equation 2.4

$$d(q) = \frac{60}{t}\frac{q}{q_{ref}} \tag{2.4}$$

where q is the quarter value we want to obtain the duration for and t is the tempo defined for the reference quarter value q_{ref}.

In note duration there is no standard tempo reference as there was the standard tuning pitch for the frequency definition of notes. This happens because this time discretization system has not enough flexibility to define all music styles. This discretization defines a recommended structure once a rhythm is defined for the melody. Therefore, this equation changes from music score to music score. Any musical symbol can be used to define the tempo in a music score but we decided to use the quarter note as as the unique reference to simplify terminology. Therefore when we use a tempo value in this work, it will always make reference to the number of quarter notes that can fit in a minute of time. With this reference set, the new expression to get the duration of a note symbol is shown in Expression 2.5

$$d(q) = \frac{60.q}{t} \tag{2.5}$$

2.3 History of speech synthesis before singing voice synthesis

The history of speech synthesis and singing voice synthesis have been closely related although singing voice synthesis appeared later than artificial speech. Taking into account that both speech and singing voice are created by the same articulatory system, it is normal to share system components when synthesizing them artificially. In this section we want to summarize the main speech synthesis attempts that took place before the singing voice was synthesized. This can contribute to locate the beginning of the singing voice synthesis in context and explain the differences in motivation of speech and singing voice synthesis.

The first efforts in speech synthesis have been dated as early as 1779 in a competition organized by The Imperial Academy of St. Petersburg, to explain the physiological differences between five vowels and their artificial production [45]. The winner, Kratzenstein, constructed a vocal tract-like resonator and he activated it using interrupted air streams similar to those generated by the vocal folds. In 1791 von Kempelen published his speech synthesis machine [38] he had been working

on for 20 years. The work of von Kempelen inspired subsequent works made by Charles Wheatstone, Alexander Graham Bell and his father Alexander Melville Bell. In 1937, Riesz developed a new synthesizer [24] with a tube that could be modified by means of nine movable components representing the lips, teeth, tongue, pharynx, and velar coupling. The mechanical and semi-electronical synthesis research continued till the 1960s with no great success [82] and although there are some modern mechanical systems [48] they are a minority in the domain of artificial speech and cannot compete in quality with digital synthesizers.

In parallel to the development of these mechanical systems, the first attempt for a electronic synthesizer appears in 1922 by Stewart [74]. In his system, the two lowest formants of vowels were created with resonant circuits. The Voder (Voice Operating Demonstrator) [39] was the first fully capable electronic speech synthesizer. It was developed by the Bell Telephone Laboratories, and a human was needed to control the synthesis process. The Voder [133] was inspired by the Vocoder (Voice coder), a method also invented in Bell Laboratories to reduce the bandwidth of a voice signal by parametrizing its slowly varying acoustic parameters and reconstructing the signal after the transmission of these parameters. Although the vocoder did not achieve the expected bandwidth optimization level, it was used during the World War 2 for secret communications [154] and brought a concept revolution to the speech and sound signal processing research area.

In 1962 Kelly and Lockbaum published a software version of the acoustic tube model, a digitized vocal-tract analog model [87]. Using this model, one year earlier in 1961, they synthesized the song "Daisy Bell (Bicycle Built for Two)" with the help of Max Matthews to create the musical accompaniment, also produced digitally. The long way made by research in speech synthesis before arriving to singing voice synthesis shows us the difference of motivation and objectives between the areas of speech and singing voice synthesis, although both signals are created by the same human vocal tract. The research of speech is related to the communication channels used by humans in everyday life, therefore is more related to the communication of a message. The first public example of synthesized singing voice appeared with a marketing purpose and singing a popular song. This shows

that the singing voice synthesis searches aesthetic and cultural purposes more than communication.

2.4 Digital analysis of singing voice signal

In the second half of the 20th century, the constant improvement of computing capacities allowed many digital methods for the synthesis of audio signals. We can classify these methods into two classes depending on what part of communication system they focus on: the input side, i.e. generation or the output side i.e. perception. Models that focus on generation are known as physical models and the physical models of human voice try to imitate the voice generation system of the humans. The defined control parameters are similar to the ones humans use in their voice generation system. These seem to be optimal systems for intuitive control of the synthesis but guessing the positions of multiple muscles and articulations in speech is not an easy task for humans. The digital acoustic tubes from Kelly and Lockbaum can be clearly classified as an example for the physical model. As previously mentioned, Kelly-Lockbaum tube was the first digital simulation of the vocal tract, created in 1962 [87]. The Kelly-Lockbaum tube is a concatenation of cylindrical tubes of the same length but different radius that attempt to model the profile of the vocal tract. The first singing voice synthesis system using the vocal tract model was made in 1993 and it was called Singing Physical Articulatory Synthesis Model (SPASM) [29]. SPASM used transition times, vocal tract shape and glottal configurations including glottal frequency (note) and vibrato.

On the other hand, spectral models are an example of the other kind of models, those focused on the output of the system. The objective of the spectral models of human voice is to generate the spectrum of the human voice. The spectral models are also known as parametric synthesis models because they extract a compact set of acoustic control parameters from the waveform. These parameters can be used to analyze the waveforms and the models create mechanisms to generate the waveform back from these parameters. The main difference with generation mod-

els is that the extracted acoustic features do not have to represent physically the human voice generation systems. Nevertheless, the source-filter theory of the human voice generation system is the base of multiple spectral models and can be considered as semi-physical. This is because the decomposition of the spectrum is based in the assumption that the human voice generation system is a source-filter system. The source-filter methods are considered as spectral models because the acoustic parameters do not have a direct physical representation in the human body. Some example of spectral models are the Frequency Modulation (FM), Formant wave functions, vocoders and sinusoidal models. Given the weight that spectral models took in voice synthesis in the second half of the 20th century, we explain their history, functioning and relation with the singing synthesis in Section 2.5.

2.5 Spectral modeling of singing voice

As we have explained in Section 2.3, the objective of spectral models is to characterize the spectrum of the human voice with parameters that do not necessarily have a direct representation in the human voice generation system. Multiple methods have been created through the second half of the 20th century that have been applied to singing voice.

2.5.1 Formant synthesizer

The formant synthesizer for speech applies the source-filter model and defines resonance filters at formant frequencies which filter the excitation. This excitation can be of two types: a noisy signal to generate unvoiced sounds and a glottal-pulse-train for voiced sounds. Additional pole and zero modules can be used to generate fricatives and nasal sounds. The first speech formant synthesizer was published in 1968 [116].

The first singing voice synthesis research based on formant synthesis was based on the OVE (Orator Vox Electrica) speech synthesizer from KTH and Fant and it was called Music and Singing Synthesis Equipment (MUSSE). According to

[147], initially the system was designed to sing just vowels but later on it was able to pronounce consonants and in 1984 [167] a system for the solmization of syllables was presented. The main parameters of the system could be controlled with knobs and were five formant frequencies and bandwidths, vibrato rate and extent, pitch-synchronous glottal noise, random variation of fundamental frequency f_0, and rate of f_0 changes between notes. The initial hardware version was substituted by a software version during the 1990's. The system has been mainly used to investigate the acoustic correlates of dynamic variations in the singing voice and on pitch perception.

2.5.2 Linear predictive coding

In 1970 Linear Predictive Models appeared revolutionizing both speech and musical technologies [8]. This method of speech coding ended up being one of the most used speech synthesis methods at the end of the 20th century. The first singing voice synthesis system using Linear Predictive Coding (LPC) was created in 1982 [47].

Based in the source-filter theory of speech generation, LPC supposes that the vocal tract is a linear filter of n coefficients and tries to calculate the inverse of this filter. Applying the inverse filter to the waveform, the effect of the formants can be removed and the remaining signal (called the error prediction or residual signal) can be analyzed to estimate the power and the fundamental frequency. With the inverse filter of the vocal tract (timbre), the fundamental frequency and the power, the three elements needed to characterize a source-filter system are obtained.

2.5.3 Frequency modulation

In 1973 John M. Chowning demonstrated the possibility of creating complex audio synthesis using FM, commonly used in radio communications [26]. Chowning used FM synthesis to create various synthetic sounds and also implemented a singing voice synthesizer [27]. This method uses multiple carrier/modulator pairs

to generate spectra with an arbitrary shape. It has been very successful in commercial computer music generation, offering very good simulation of musical instrument sounds, using very few computational resources. To synthesize vocal sounds, the carriers are placed near the formant frequencies, and a common modulation oscillator at the fundamental frequency is used.

2.5.4 Sinusoidal modeling

Sinusoidal modeling is based in the theory that speech may be constructed as the sum of multiple sinusoidal waves with time-varying amplitude and frequency. The first attempt to construct a speech analysis/synthesis system with sinusoidal modeling was made in 1986 [94] and used peak detection in the Short-Time Fourier Transform (STFT) to estimate the parameter's values. This modeling had great success in music synthesizers and the first singing voice synthesizer using sinusoidal modeling came in 1997 [89]. The singing voice synthesis model used Analysis-by-Synthesis/Overlap-Add (ABS/OLA) and added vibrato and vocal intensity variations to the original speech controls. The success of sinusoidal modeling for the synthesis of music drove the researchers to improve the modeling creating the sinusoidal plus residual model [137]. In 2001, Bonada et al. created a singing voice synthesis system using a new parametrization called Excitation plus Resonance (EpR) [20]. The EpR model combined the sinusoidal plus residual model with the source-filter model. In 2007 the Vocaloid was released, a commercial singing voice synthesis systems with EpR parametrization [72].

2.5.5 Formant wave functions

Formant wave functions (known as FOF because of the definition in French Fonction d'Onde Formantique) simplify the source-filter definition of signals by a time-domain decomposition of the signal in a finite number of FOFs [119]. This decomposition can be faster than source-filter methods because it avoids source-filter separation. With this method, the singing voice synthesizer CHANT was created in

1984 [121].

2.5.6 Mel cepstral analysis

The concept of cepstrum was originally used for pitch detection [104]. The cepstrum is the Fourier transform of the log-magnitude spectrum. This transformation is very efficient finding periodicities in the signal. Considering that voiced speech is created with a glottal pitch, the cepstrum coefficients show a peak at the index corresponding to the fundamental frequency of the voice. Mel cepstrum synthesis uses the fundamental frequency and the mel log spectral envelope codification to reconstruct the speech signal. The first speech synthesis system to use mel cepstral parameters was developed in 1983 [70] and reduced the speech encoded data in 60-70 % while maintaining the quality. The first singing voice synthesis system using mel-cepstral coefficients arrived in 2006 [123]. This system arrived relatively late compared with other systems because during the 90s the sinusoidal modeling had been the most popular vocoding method for singing voice synthesis.

2.6 Artificial singing voice synthesis

The voice parameters or the concatenative units are the base for speech generation, but which units to choose or how to generate the correct parameters has been a research topic of growing interest through the second half of the 20th century. We have to take into account that the origin of synthetic speech and singing voice are text and music scores respectively and this means that a transformation has to be made to create an audio signal from a symbol sequence. In a music score, while musical notes define the mean fundamental frequency of the notes, duration values and loudness, the lyrics define the phonemes pronounced in each note by means of the orthographic text. The creation of the signal from this information is what defines a singing voice synthesis system. We have explained in Section 2.5 that there are many methods to express a singing voice signal in a parametric way; in this section we analyze what type of methods have been developed during the last

years to convert a sequence of symbols into audio. Usually the input sequence of symbols will be converted into a sequence of parameters representing the voice, and then finally synthetic voice will be reconstructed from the parameters. We also analyze the concatenative method because of its importance in the singing voice synthesis area. The concatenative method concatenates waveform units of natural singing voice to generate new synthetic audios without any parameter step. We described here three main methods to generate audio from music scores; the rule based parametric method, the concatenative method and the statistical parametric method.

2.6.1 Rule based parametric synthesis

In rule based parametric systems, the sequence of parameters to generate the singing voice is created from the information given by the music score using a set of rules. These rules are created observing real interpretation of music scores. These rules have to define the duration of notes and the phonemes inside each note, the articulatory parameters of each phoneme and the pitch characteristics defined by the notes. Although the note sequence is a discrete sequence, its translation to a natural fundamental frequency of a singing voice cannot be done as a staggered curve. The variation of the pitch from the nominal mean, the vibrato and the transition between notes have to be taken into account. A general scheme of a rule-based singing voice synthesis system can be seen in Figure 2.6.1.

In the second half of the 20th century, the majority of the singing voice synthesis systems that used spectral parameters were rule based. This is why many of the systems cited in Section 2.5 are rule based. Singing voice synthesis systems defining rules for FM [27], formant synthesizers [167], LPC [47] and FOF [121] had been published through the 80s. The research group Speech Transmission Laboratory (STL) of KTH, developers of the singing voice synthesis system based in formant synthesis [167], developed the most important contributions to the rule-based singing voice synthesis [144][13][147].

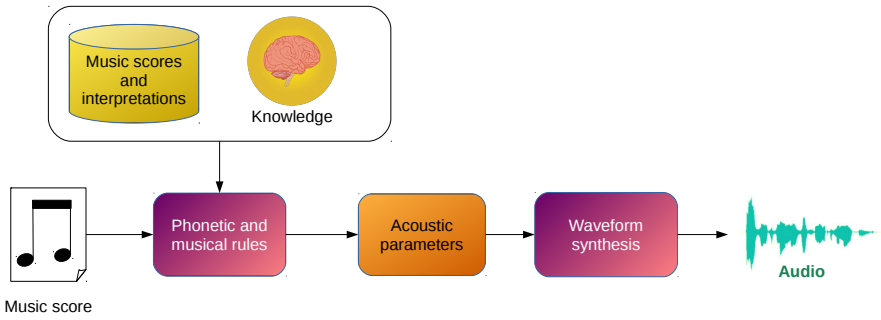


Figure 2.6.1: Rule based singing voice synthesis scheme

2.6.2 Concatenative synthesis

While all voice parametrization methods tried to synthesize speech with the best possible quality using parametrization of audio signals to reduce the number of features, methods using natural audio segments were also under study in the last decade of the 20th century. In concatenative synthesis systems, parameters are extracted from music scores with fixed rules as in rule based parametric systems, but in a second step these parameters are used to select the optimal combination of natural singing segments to represent the music score. The optimal units are selected considering the cost of including a unit in a specific position of the sequence. This cost usually includes two main elements, the target cost and the concatenation cost. The target cost calculates how the selected unit fills the conditions of the position. The concatenation cost takes into account the amount of change that the unit will need in the boundaries to obtain acceptable transitions in the resulting audio. After the selection of the units, the transitions between independent units are smoothed using different techniques. The structure of a concatenative singing voice synthesis system can be seen in Figure 2.6.2.

The quality of concatenative systems is dependent on the database used for the selection of samples. There is a compromise between the challenge of using a big

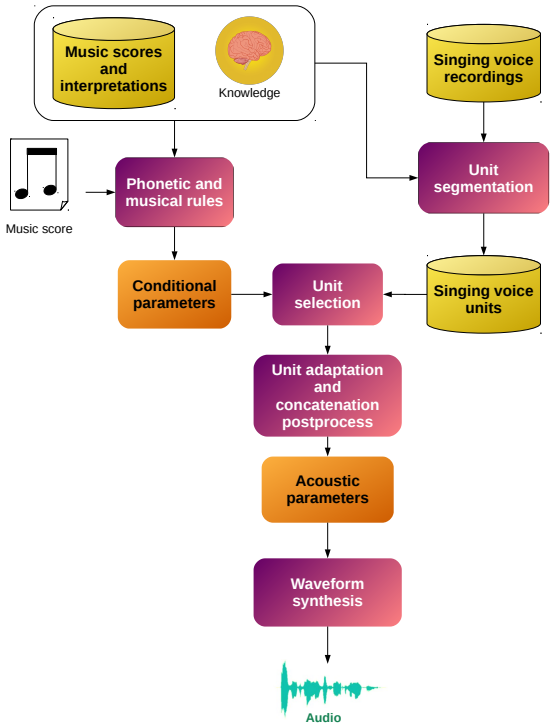


Figure 2.6.2: Concatenative singing voice synthesis scheme

database and the quality of the synthesis. Small databases are easier to create but the synthesis quality is limited due to the lack of segments for many combinations of contextual factors in the scores (i.e. pitch values, lyrics, and durations). A bigger database can cover more music score symbols and contexts but is harder to prepare; it also makes the unit selection problem more difficult because the search space is bigger.

In 1958 the first concatenative system for speech synthesis was published by GE Peterson [112]. In that system transitions and stable onsets of phonemes were concatenated. Concatenative speech synthesis systems evolved from using databases made up of diphones and rule-based simple selection to concatenate them (usually using signal processing to smooth the discontinuities), to databases of many hours of speech and elaborated unit-selection algorithms.

Singing voice synthesis has different prosody rules comparing to speech synthesis, because music scores set the pitch values that every phoneme must have. Nevertheless, the range and expressiveness of the singing voice is wider than that of speech. The first concatenative singing voice synthesis was presented in 2002 [85]. Although it is not the first singing voice synthesis system that used natural recordings as concatenative units, it is the first system that used original signals without parametrization. In 1997, the first sinusoidal model synthesis system also used natural signal segments as base [89], but the sinusoidal modeling of these segments was used to create new signals.

As it is impossible to cover all the possible contexts in singing voice, adapting the units to the needs of the synthesis is a common practice. In the system Lyricos, natural voice recordings were adapted with ABS/OLA method to create more natural singing voice with concatenation [89]. The first version of Vocaloid, the state of the art commercial synthesis system, adapted the singing voice units using spectrum scaling methods to modify the pitch of the singing voice units [72]. Concatenative synthesis achieved the best results at the beginning of singing voice synthesis and the databases started to become bigger and more sophisticated to cover different linguistic and musical contexts. Umbert et al. added modules to control the f_0 and vocal dynamics to improve the expression of concatenative singing voice systems [157].

2.6.3 Statistical parametric synthesis

Statistical parametric synthesis is a singing voice synthesis method that models parametric representation of the singing voice in a statistical way. The statistical modeling is made analyzing the distribution of the acoustic parameters of similar signal segments. The general scheme of this system can be seen in Figure 2.6.3. These systems started to grow with the use of HMMs. The HMMs also helped to segment the phonemes in the databases in an automatic way. First automatic segmentation of databases for speech and the use of these databases for synthesis started in 1998 with two tools for automatic segmentation: Whistler, created

by Windows [113] and another system published by International Business Machines Corporation (IBM) [36]. The Blizzard challenges of 2005 and 2006 proved that although the ceiling quality of the Statistical Parametric Synthesis Systems (SPSS) for speech was lower than that of the unit selection synthesis systems, the overall quality was better. These first HMM systems were based on mel-cepstral parameters. The first singing voice synthesis system using statistical parametric synthesis arrived in 2006 [123] and used a time-lag model to adjust the phoneme boundaries to music tempo conditions. This system also used mel-cepstral parameters as the first HMM speech synthesis system. The mel-cepstral parameters are the most popular in the statistical singing voice because the fundamental frequency parameter can be controlled independently. We explained in Section 2.1.1 the importance of the fundamental frequency for the tuning and expression of the singing voice.

In 2010 the system described in [123] was improved using pitch-shifting extensions of the database [91] and vibrato [108]. In 2012 pitch adaptation training was introduced [109] and singer adaptation [139] in 2014.

In 2013 the first DNN speech synthesis system was presented [166] and in 2016 the same mechanism was applied to the singing voice [103]. In 2017 Wavenet was published with architectures to create raw audio with neural networks directly from parameters and in 2017 the Neural Parametric Singing Synthesizer (NPSS) presented a singing voice parameter generation using a similar architecture [15]. In 2018 Neural Network Tacotron introduced the first Wavenet that generated speech waveform from mel-spectrum [138]. This brought a revolution to the speech synthesis because opened the possibility to synthesize speech without using the historical parametric analysis of speech signals. Nevertheless, the majority of statistical parametric singing voice synthesis systems continued to use mel-cepstral parameters to have more control of the pitch of the singing voice.

In the following years many neural network architectures have been proposed for singing voice synthesis: Convolutional Neural Network (CNN) [102], Conditional Generative Adversarial Network (CGAN) [64] and Wasserstein Genera-

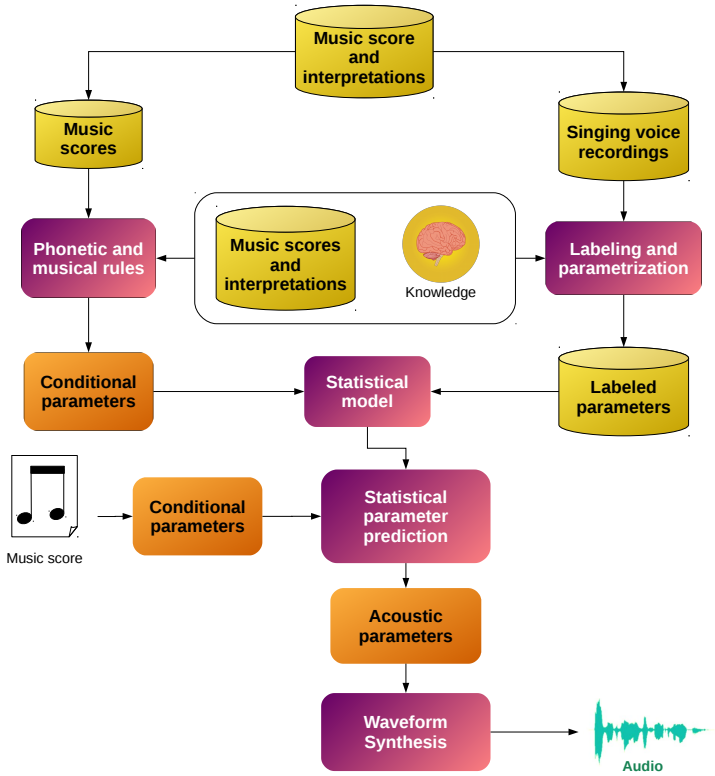


Figure 2.6.3: Statistical parametric singing voice synthesis scheme

tive Adversarial Network Sing (WGANSing) [25] are the most remarkable ones. Efficient singer voice cloning has also been achieved using neural networks [17].

2.7 Singing voice labeling

We explained in Section 1.2 that one of our objectives is to segment and label a singing voice database that contains multiple elements apart from singing voice in the recordings and has no music scores of the singing voice segments. This is why we present the state of the art of different aspects related to the labeling of

singing voice recordings. We analyze three main areas: the segmentation of the audio signal, the alignment of lyrics and audio and the musical transcription of singing.

2.7.1 Audio segmentation

One of the common problems in the research about the singing voice is that the majority of the recordings are polyphonic audios with musical instrument accompaniment. Bertsolaritza normally has no musical accompaniment but the recordings contain applause, noise, silence and speech. We call audio segmentation to the separation of all these components. The separation of applause, noise and silence from speech and singing voice is a common Voice Activity Detection (VAD) problem. In this section we focus on the discrimination of speech and singing voice, both classified as voice by VAD.

Discrimination of speech and singing voice is not an easy task even for humans, who need approximately one second long segments to discriminate singing voice and speaking voices with more than 95% accuracy [105]. When speech is repeated, rhythm patterns appear and repeated spoken segments are perceived as singing [34], [44]. Although singing voice and speech are closely related and difficult to be distinguished by humans, the technologies developed for spoken speech are not directly applicable to singing voice. In general, results deteriorate heavily in tasks such as automatic speech recognition [95] or phonetic alignment [88]. Therefore when dealing with recordings that contain both speech and singing voice a tool to discriminate among them must be first used. Afterwards, the technique or strategy suitable for each type of voice can be safely applied.

There are two different tasks related to the automatic discrimination of singing voice and speech: classification, when each segment (or file) belongs to only one class and segmentation where both classes are present in the same file and have to be first separated and then classified. For the classification step short-term and long-term features extracted from the audio signal have been traditionally used. With the use of short term features the signal is classified at frame level and then

decisions for different frames must be combined to obtain a single decision for each segment.

Most works, however, rely on long-term parameters related to pitch to discriminate speech and singing voice. For instance, the distribution of different pitch based features was tested in [53] and most of them achieved to correctly classify more than half of the items in the database. The best results were obtained for correlation of pitch between syllables with 78 % accuracy. This feature tries to find the similarities of the pitch between syllables, considering that syllable repetitions are more probable in singing voice.

Pitch and energy related features are also used in [136] where they are feed to a multilayer Support Vector Machine (SVM) obtaining a 99.22 % accuracy. The pitch-related features were the ones that contributed most to discrimination. Thomson [150] proposed to use the Discrete Fourier Transform (DFT) of the pitch histogram to estimate the distribution of pitch deviations. These deviations should have lower variance for singing voice than for speech. He obtained the best results using segments of 16 seconds with 98 % accuracy.

Some works combine short-term features related with the spectral envelope and long-term features related with prosody to train GMMs and distinguish speech and singing voice. In [105] the authors found that short term features work better for segments shorter than 1 second while pitch-related features obtain the best results for segments longer than 1 second. The best result obtained in this work nevertheless used segments of 2 seconds with 94 % accuracy. On the contrary, the work in [156] also uses a GMM classifier and finds that spectral features alone work better than pitch based ones alone when using segments between 17 and 26 seconds. Regardless, he obtained the best results when combining both types of features. The obtained best result is 94 % for accuracy. In [135], a large set of 276 attributes related with spectral envelope, pitch, harmonic to noise ratio and other characteristics and a ensemble of classifiers are proposed to classify singing voice, speech and polyphonic music. Using all the features to train an ensemble classifier with a Naive Bayes, a k-nearest neighbor, a SVM, an unpruned C4.5, a bagging C4.5 and a boosting C4.5, the system gets 99.43 % accuracy.

2.7.2 Lyrics alignment

Lyrics alignment is one of the key parts of the singing voice synthesis process. As in speech synthesis, the alignment of phonetic representation of lyrics text with the recording is necessary to model the spectrum and duration of the different elements in the singing voice. Because both speech and singing voice use the same phonemes lyrics and text alignment methods are very similar, but lyrics alignment includes musical information in the process.

Comparing to speech phoneme alignments, the singing voice poses a problem because of the background music. The majority of singing voice recordings are polyphonic because the singing voice is combined with multiple instruments in music. This is why the lyrics alignment process includes the detection of the singing voice in the songs and the separation of the voice from the music to align the phonemes. The phoneme alignment techniques applied in monophonic singing voice recordings or after the detection of the singing voice signal in polyphonic recording are closely related to the methods used in speech.

The first automatic phoneme alignment in speech was a result of the forced alignment technique used for speech recognition [71]. With the growth of the speech databases and the increase of data-driven speech synthesizers, the manual segmentation of speech databases became highly time consuming and the research of automatic phoneme alignment became essential. Different methods of phoneme segmentation were developed: Dynamic Time Warping (DTW) alignments using parallel synthesized signals [90], HMM based alignments using Viterbi algorithm to optimize the phoneme positions [162] and two stage alignment methods that refined initial alignments obtained by HMMs using SVM [86] [79] or Neural Networks [80]. Different variations of segmentation systems have been developed considering multi-speaker scenarios, for example using Maximum Likelihood Linear Regression (MLLR) or Maximum a Posteriori (MAP) algorithms [168]. In 1999, two years after the publication of the HMM based speech alignment algorithm by Wightman and Talkin [162], Loscos et al. created an HMM alignment system for singing voice synthesis including aspiration states and du-

ration conditions predicted from music scores. In 2004 the LyricAlly system was published[160]. This system used the song structure, multi-model HMMs and different types of parametrization of audio segment to align lyrics lines to the signal. Later, the musical chords information in the phonemes [92] and musical duration information of syllables [40] were used for lyrics alignments. In 2018 Gong and Serra used CNN onset detection combined with Hidden Semi-Markov Models (HSMM) with musical duration condition to segment phonemes in singing voice [58].

End-to-end speech synthesis systems like Tacotron [161] introduced speech synthesis methods that do not need previous phoneme alignments. These systems model the alignments between input characters and acoustic frames during the training, using attention mechanisms. Instead of using attention mechanisms to align the phonemes with the acoustic parameters in an automatic way, Blaaw and Bonada decided to force the attention alignment in the training using pre-defined note duration models [16]. The hypothesis is that the network can learn the deviation of the acoustic parameters starting from the theoretical alignment although the alignment is not perfect. This method is applied because of the importance of the tempo and the note timing in singing voice synthesis.

2.7.3 Musical singing transcription

Singing transcription is an important part of the area of Music Information Retrieval (MIR). The objective of singing transcription is to represent the notes that constitute the sung melody in the recording. In MIR, applications like query by humming [30], singing scoring [97], singing tutors [66] and musicology studies [77] need the singing transcription data for research. The work of manually annotating the notes in the singing voice is challenging and the definition of standards is complicated even for experts [83]. Systems that label the notes in singing voice in an automatic way with no manual supervision have become a subject of interest with the growth of big data and MIR. This area is named Automatic Singing Transcriptions (AST) and is specially challenging because the diversity of styles, the

difficulties to sing in a perfect tune for the majority of humans and the expressive elements that appear in singing voice to add expressiveness (vibrato, portamento, etc.).

Considered an unresolved problem still in early 2000s [73], different approaches have been developed to automatize this process in the 21st century: Pollastri used range sequences with histogram data to adjust tuning [114], Wang et al. used Adaptive Round Semitones (ASR) to correct out of tune notes and musical grammars to obtain musically coherent sequences [159], Mulder et al. used the energy boundaries to separate potential notes and applied a special treatment to process portamentos [31], Ryynänen and Klapuri used HMMs to model notes and musicological rules [122], Gómez and Bonada optimized paths across matrices with frame/note values and iterative methods to annotate flamenco recordings [57], Mauch et al. modeled HMMs and probabilistic pitch combinations to detect the most probable notes and Molina et al. used power and aperiodicities to detect note boundaries and dynamic averaging to detect the note pitch values [99].

The lack of a standard evaluation method has also been discussed by Molina et al. [98]. The evaluation has to take into account note onsets, offsets, pitch and note purity. All these parameters result in a multivariate evaluation and considering that pitch values are not always predicted in a discrete scale with 440 Hz A_4 reference, the acceptable error margins have to be defined.

2.8 Bertsolaritza

Bertsolaritza or bertsolarism, is the art of singing improvised songs called bertsos in Basque. The improvised verses are conditioned by previously defined rhythm patterns, structures and melodies. Bertso singing is performed in many different social contexts: at local festivals, celebrations lunches and tributes to outstanding people. However, the most relevant events related to Bertsolaritza are the competitions which take place every year. Considered as improvised literature, its early origins and constant reinvention makes it a relevant reflection of the society of

each time. In this section we will explain a resumed history of bertsolarism and the current state of this art. We will also address the research made about this art in the area of signal processing until this moment.

2.8.1 History

The history of bertsolarism is closely related to the history of the Basque language which is considered as the last remaining descendant of the pre-indo-european languages from Western Europe [155]. In spite of this longevity, the written documents of Basque have been few till early 20th century, although the first Basque book, a poetry collection named *Linguae Vasconum Primitiae*, was published in 1545. This combination of longevity and non literacy makes oral tradition very important. Basque literary culture has been scant until the beginning of the 20th century and various artistic expressions have been classified as "Basque popular literature". Basque popular literature is mainly oral and bertsolarism can be considered a sub-genre of it [50].

If we try to define the origin of bertsolaritza, we can find that the long oral history makes it difficult to know it with high precision. The hypotheses about its origins can be classified in three main types.

- **Ancestral theory**: A theory influenced mainly by Manuel Lekuona, the first real scholar of bertsolaritza. It argues a prehistorical creation of bertsolaritza using as argument the poetic activity of Basque people from the origins and the origins of Basque language [60].

- **Modern bertsolaritza theory**: In opposition to the ancestral theory, a modernist theory analyzes that the first mention of bertsolaritza occurs at the end of the 18th century [81]. Nevertheless, these first references present bertsolaritza as a fully developed and mature cultural expression by this time.

- **No explicit root theory**: This theory, developed by Luis Michelena [96], marks the first documented references of this art in the 15th century. Al-

though the term bertsolaritza is not used, laws banning street singing written in 1452 exist in Ancient Charter for Bizkaia. These laws refer explicitly "women who are shameful and agitator, and who sing in the street".

Although there are many theories about the origin of bertsolaritza, there is consensus that it is at the end of the 18th century that first records talking explicitly about bertsolaritza appeared. The beginning of the 18th century is considered as the popularization period of the bertsolaritza competitions. The spread of the enlightenment in this century developed Basque literature and by the end of the century, records of bertsolaritza and the first well known bertsolari appear, Pernando Amezketarra (1754-1823). The 19th century is better documented and the bertsolaris from this century are considered as "classical". The improvement of documentation of this art did not expand to the improvisation world. The documentation refers mainly to non-improvised poetry and biographical data of the bertsolaris. We have to wait until mid 20th century and the availability of audio recording devices for the appearance of considerable corpus and trustworthy transcriptions of improvised sessions.

In 1935, the first-ever Basque Country Bertsolari Championship was held by the group Euskalzaleak and 19 bertsolaris took part on it. This was a first strong step in the professionalization of the bertsolarism, an art related mainly to environments of cider houses and bars until then. Topics and styles started to expand from that year on. Although we can find in the 19th century bertsos with topics like the loss of historical rights of Basque people, the professionalism brought a deeper analysis of the art and improvisation about fictional situations that contributed to expand the art. Competition also brought the bertso scoring system and criteria deepening the quality analysis. During the Spanish Civil War (1936-1939) and Franco's dictatorship (1939-1975) bertsolaritza went from censorship in the 40s and 50s to become a powerful denounce of repression and appearing regularly in the media in the 60s and 70s. In the 80s, after the end of the dictatorship, bertsolaritza was modernized gaining attention from a broader public. First bertso schools appeared at the time. Since the 80s, the bertso in stage has not stopped growing

without abandoning cider house and bar culture in parallel. Nowadays, the bertsolari championship is celebrated every 4 years where the champions from 7 different provinces compete. In the last edition 14.600 people attended to Bilbao Exhibition Center to the championship final.

In 1991 the Xenpelar Documentation Center was created to compile all the digital information related to bertsolaritza such as melody scores, recordings, transcriptions as well as meta-information of these records. This center has continued with the systemic compilation of the bertso session till this day, and the majority of the compiled data has been used in this work. The details of the used data are explained in Section 3.1.1.

More extended information about the history of bertsolaritza can be found in [51] [107] [10][50].

2.8.2 Women in bertsolaritza

The presence of women in bertsolaritza scene has not been ample during early documented years. Although one of the origin theories arguments that first references of improvisation singing are set in the 15th century with women singers, it is hard to find many more examples. All the "classical" bertsolaris from the 18th century are men and the first participation of a woman in a bertsolaritza championship happened in 1985. Nevertheless, the growth of women in bertsolaritza has not stopped since the 80s and in the main championship in 2017 a woman won the final for the first time in history.

2.8.3 Bertso structure

Bertsolaritza is an art of poetry improvisation and therefore we can consider that any creation is constructed with strophes, lines and rhymes. In bertsolaritza the term bertso is used to define the strophe and each improvisation can have multiple bertsos. The bertso is the basic unit of bertsolaritza and the terminology refers to different structure and elements of the bertsos. In Figure 2.8.1 we can see the main

elements of a bertso.

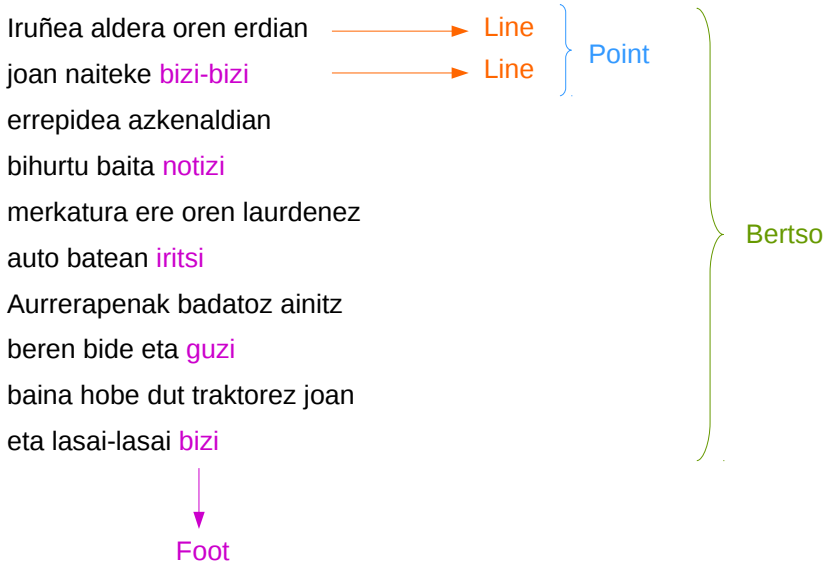


Figure 2.8.1: Main elements in the bertso structure

The definition of these elements and other definitions about the structure of the bertso are given below:

- **Bertso**: Strophe of the improvisation.
- **Line**: A line from the bertso.
- **Foot**: The last word of each line including a rhyme.
- **Point**: Two consecutive lines ending with a foot.
- **Meter**: A structure defining the number of lines of a bertso and the number of syllables in each line. Every different meter has its name. The most common meters are listed in Table 2.8.1.

- **Melody**: Melody used to sing the improvised lyrics. Closely related to meters, each melody can be used to sing bertsos with a specific meter as the number of syllables has to be adjusted to the number of notes in the melody.

Name	N. of lines	Syllable structure
Zortziko minor	8	7/6/7/6/7/6/7/6
Zortziko major	8	10/8/10/8/10/8/10/8
Hamarreko minor	10	7/6/7/6/7/6/7/6/7/6
Hamarreko major	10	10/8/10/8/10/8/10/8/10/8
Hamaseiko equal	16	8/8/8/8/8/8/8/8/8/8/8/8/8/8/8/8

Table 2.8.1: Definition and names of most common meters

2.8.4 Bertso performance types

The bertsolaritza art is closely related to live shows. Non-improvised bertsos and melodies can be seen in music albums, press or homages, but the art is considered to have its base on the improvisation. The improvisation can be completely free on meter and theme, but usually conditions for the improvisation are set by an external host or by the organizers in live shows. Depending on the conditions given for the improvisation, we can classify different types of bertso performances. The first characteristic of an improvisation is that bertsolaris can perform solo or in pairs. In the solo performances these categories can be performed:

- **First point based**: The host sings the first point of a bertso and the bertsolari has to improvise the rest of the bertso. The meter and melody conditions for the improvisation are implicit in this first point. This exercise is usually made with bertsos using the zortziko minor meter.
- **Feet based**: The bertsolari is given four feet and a meter. All four feet must be used to create a bertso respecting the meter.

- **Final point based**: The host sings the last point of a bertso and based on this, the bertso must be completed. The meter and melody conditions for the improvisation are implicit in this last point.
- **Topic based**: A topic and a meter are given to the bertsolari and one or more bertsos have to be improvised around the theme with the specific meter.

In pair performances we find the next categories:

- **By trades**: The host gives to each bertsolari a trade or role to play and defines a meter. The bertsolaris alternate turns to improvise a single bertso responding to each other.
- **By points**: The host gives to each bertsolari a trade or role to play and defines a meter. The bertsolaris alternate turns to improvise a single point in each turn.

2.8.5 Research in bertsolaritza

As explained in the previous sections, bertsolaritza is nowadays a strong improvisation literary art with a corpus big enough for analysis. Many research areas have been opened around bertso art. In this thesis we have focused in the audio recordings and signal processing but we can observe research in many other areas.

- **Literary research**: The growing number of transcriptions from improvisation sessions and championships have increased the size of the corpus available for the analysis of poetic and rhetoric aspects of bertsolaritza [49] [52].
- **Sociological research**: As explained earlier in this document, the influence of bertsolaritza in the Basque society after hundreds of years of presence is deep enough to analyze it as a sociological phenomenon [117].

- **Musical analysis**: The multiple melodies used in bertsolaritza have been collected by Xenpelar Organization. The analysis of these melodies has opened the research in music genre classification [54] and melody generation [55].

- **Natural Language Processing (NLP)**: The goal of NLP is the understanding and generation of natural languages by computers. Although literature and poetry generation are one of the hardest challenges in this research area, multiple works have been developed to create bertsos with structure and rhyme conditions in an automatic way [1] [7].

- **Robotics**: The work of automatic bertso improvisation evolved into the development of an on-stage robot imitating the behavior of a bertsolari to sing automatically generated lyrics [6].

When it comes to the analysis of the bertsolaritza singing voice, we can only find published work about the the prosody of bertsolaritza [2]. Although there are many recordings of historical bertsolaritza sessions, not much research has been done about the singing style of bertsolaritza. The research of the on stage robot by Astigarraga et al. [6] created the need of a singing voice for the robot. In that first work, a statistical speech synthesis system was used with forced durations and pitch to match the requirements of the melody.

2.9 Chapter conclusion

In this chapter we covered the relevant research related to this work and resumed the bertsolaritza art. It has been made clear that singing voice synthesis systems take reference technologically from speech synthesis system. However, a different expressiveness challenge appears in singing voice synthesis. In the second half of the 20th century, a revolution in all areas related to digital information has been produced and speech and singing voice have not been an exception. The increase of digital data and computer capacities have ended up giving the advantage to data

driven statistical systems in the 21st century, but concatenative systems are still competitive. The increase of corpus size in bertsolaritza art since the decade of the 60s fits perfectly in the narrative of big data in the modern era of statistical signal processing and this thesis tries to fill the gap of the missing research around the world of bertsolaritza. This increase in database sizes also created the need of systems to label singing voice synthesis databases at recording, phoneme and musical levels. The adaptation of these systems for bertsolaritza can open new research lines in musicology and signal processing areas.

3 Materials

This chapter presents the basic materials utilized to develop the singing synthesis system applied to bertsolaritza. We have used several databases to train and test the different tools developed in the thesis and diverse existing software tools. First, in Section 3.1 we will describe the three different databases of singing voice we have used for the development of this thesis: the Bertso database obtained from the BDB database compiled by Xenpelar Dokumentazio Zentroa, the NUS database and the NITech database. For each database, information about the audio characteristics, quantity of audio and available metadata will be provided. In Section 3.2 the main software tools used in this work will be briefly described.

3.1 Databases

3.1.1 Bertso database

The Bertso database is a modified version of the database Bertsolaritzaren Datu-Basea (BDB) (https://bdb.bertsozale.eus), collected, organized and sustained by Xenpelar Dokumentazio Zentroa. In this section we will explain the information that the original BDB database already provided and the information we have added to it in order to make it suitable for the development of singing speech synthesis systems. We have augmented the BDB database with two types of information: on the one hand we have added complementary information not related with audio processing and completed missing metadata. On the other hand, we have segmented and labeled the original audio signals and added this information to the database. In this section, we will describe the procedure followed to add the first type of information and the segmentation and labeling process will be explained in Chapter 4.

3.1.1.1 Original BDB database

The original BDB database is composed by recordings of bertsos and melodies of bertsolaritza. All recordings have metadata attached to them, but some recordings have more information than others. We made two lists to define the metadata parameters provided in the database: one lists the features common to all recordings and the other gathers the features that are defined only in some recordings. The features defined in all recordings are the next ones:

- **Date and Place**: Place and date where the recording has been made.
- **Bertsolari**: Name of the bertsolari who sings in the recording.

The features present only in some of the recordings are the following ones:

- **Melody**: Melody used in the recording.

- **Meter**: Metric of the sung bertsos.
- **Identity of the host**: Name of the host who sets the theme for the improvisation.

The initial files contained in the BDB database, the corresponding file formats and the database structure can be seen in Figure 3.1.1. Each recording has an associated MP3 file with the audio recording and a text file with the orthographic transcription of the singing segments of the recordings. The melodies have a graphical representation of the score in Tagged Image File Format (TIFF) or PDF format, a MIDI format file and a piano interpretation in a MP3 sound file. The bertso recordings that have associated information about the sung melody include a Melody ID that relates the recording with the used melody.

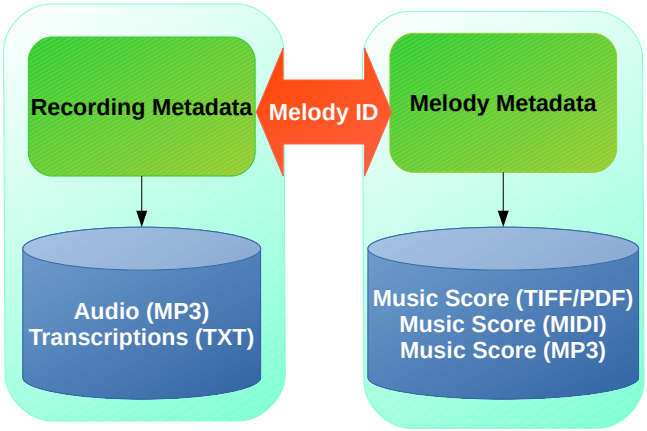


Figure 3.1.1: Structure of the original BDB database

We analyzed the original BDB database and defined the modifications in the metadata, transcriptions and file formats that were necessary to accomplish our goals. These modifications are described in the next subsections.

3.1.1.2 Audio pre-processing

The original recordings in the BDB database are MP3 files with 22.05, 44.1 or 48 kHz sample rate. We converted and down-sampled them to obtain Waveform Audio File Format (WAV) files with 16 kHz sample rate. As the original database contains audio signals recorded with different recording equipment and in different environments, we applied a loudness normalization algorithm, according to EBU R128 [41]. The process is visualized in Figure 3.1.2.

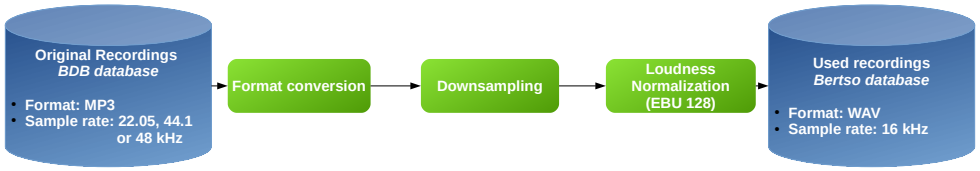


Figure 3.1.2: Pre-processing steps applied to the recordings

3.1.1.3 Music score conversion

The graphical musical scores in TIFF format have been converted to PDF format, to have all the graphical representation of the scores in the same format. The music scores of the 30 most used melodies have been manually converted to Music extensible Markup Language (MusicXML) format with the help of Audiveris [14], an Open Source Image recognition program that converts printed music scores to machine readable formats. We realized that the music score have no tempo information, mostly because bertsolaris can choose freely the tempo they want to use in the competitions.

3.1.1.4 Transcription correction

To prepare the data for its use in singing synthesis, automatic segmentation processes must be applied. For these segmentation processes to get good results, it is crucial that the transcription matches the actual content of the recording. The

original transcriptions in the BDB database do not match the recording due to two sources of problems: encores and errors in the transcriptions. A transcription modification procedure has been developed to tackle these problems.

Encores are very common in bertsolaritza but the transcribed lyrics are saved without them because the encores are a musical ornament more than a poetic figure. In some cases the original orthographic transcriptions provide an indication of the line of the lyrics where the encore is made, but other transcriptions do not have this indicator present. The type of encore is never signaled, so the number of lines that are repeated and other musical humming effects used in encores are not known. We manually determined all the encores listening to the recordings and then labeled the type of encore to create an accurate orthographic transcription of the singing voice.

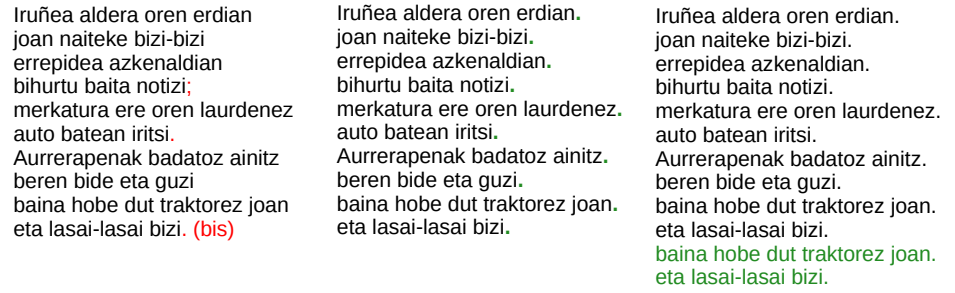
As our goal is to work with singing synthesis of closed structure melodies, we saved the intermediate files generated in the transcription correction process for analysis. Faithful orthographic transcriptions are important in order to achieve three goals:

- **Structure detection**: As we have already mentioned, not every recording is labeled with the corresponding melody or structure in the original BDB database. Knowing the meter reduces the set of possible melodies a bertsolari may be singing in a recording, making the process of structure detection easier.

- **Note alignment**: When singing, and even more frequently in bertsolaritza, the text sung inside a note corresponds usually to one syllable. Knowing the transcriptions allows getting syllable information and can make the alignment of the notes in the audio easier.

- **Phoneme alignment**: To train the synthesis systems the phonetic transcription must be aligned with the audio data. A reliable orthographic transcription is crucial to get an accurate alignment between phonemes and the audio signal.

The process designed to get the accurate orthographic transcription is composed of three steps. In Figure 3.1.3 we can see the three different types of orthographic transcription we generate during this transcription correction process.



(a) Original transcription **(b)** Meter transcription **(c)** Transcription with encores

Figure 3.1.3: Transcription types

The transcription represented in Figure 3.1.3a is the format of the original transcription provided in the BDB database. It is written in a literary way using punctuation marks, cites and encore marks (bis). The encore indicators do not define the number of lines that are repeated. This number is usually defined in the melody used to sing the bertso, but the use of melodies is irregular in bertsolaritza and on occasions the number of lines repeated is not respected and has to be manually revised. The intermediate orthographic transcription shown in Figure 3.1.3b adds no new lines to the original transcription but cleans all punctuation marks and "bis" references. We also add periods in the end of each line, because our intention is to split the recordings in bertso lines and consider each line as an utterance.

After this initial cleaning process, the final version of the orthographic transcription displayed in Figure 3.1.3c adds the encores. In some cases the encores repeat only one line and in other cases, two contiguous lines in the lyrics are repeated. The line duplication can occur at any line of the lyrics but usually happens in the final and middle lines. Finally, the humming is added. Each melody has an associated humming but this standard humming can have small variations from bertso to bertso and sometimes bertsolaris might avoid using them. To make the annotation

process less time consuming, we standardized the transcription of the humming and we wrote the same line every time that it is sung. We manually revised the position and verified the use of humming in every recording. We have observed two different types of humming in the BDB database. The first humming is used before a encore of a single line and adds the transcription "Larai larai larai larai" as seen in Figure 3.1.4a. The second type of encore creates a new line in the original transcription with the text "Ai ai ai ai" as seen in Figure 3.1.4b.

bidasoan han doa erreka.
harria eta hondarra.
oraindik ez da guztiz aldatu.
xalbador zenan oharra.
bitan zatitzen bitan puskatzen.
baitu gure herri zaharra.
ta gaur erreka ikusi dut nik.
haundia eta azkarra.
mendiek ez ezik geronek ere.
egiten degu negarra.
larai larai larai larai.
egiten degu negarra.

(a) Humming type 1

zarzuelan urduri.
antzean dabiltza.
antieju beltzakin.
hala da bizitza.
erregian gerua.
hain degu bortitza.
hobe luke handikan.
ai ai ai ai.
urrun joan balitza.

(b) Humming type 2

Figure 3.1.4: Humming types

In addition to these corrections, we manually corrected all the orthographic transcriptions that did not match with the corresponding recording. The final version of the orthographic transcription is the one that matches the singing voice in the recording. This final transcription is the one we are going to use for the phonetic alignment.

3.1.1.5 Metadata completion

As we have already commented, many recordings in the original database lack metadata. To properly extract the labels needed to train the synthesis system, metadata should be as complete as possible. We have developed some procedures to deal

with this missing metadata problem.

- **Meter detection**: We created a meter detection algorithm to label the transcriptions in an automatic way. This procedure will be explained with detail in the next Section 3.1.1.6.

- **Host detection in point exercises**: In the point exercises, the host sings the first point of the bertso. As our objective is to match the singing voice of the recordings with the lyrics transcriptions, the identity of all the singers has to be defined. This host identification has been manually made.

- **Recording errors**: Some recording files have errors or are very noisy. These recordings are manually detected and properly labeled.

3.1.1.6 Meter detection

To generate the proper labels for singing synthesis, we need to associate the lyrics sung in the bertsos with the corresponding music information. Some of the recordings have information about the meter used in the bertsos. We have analyzed the provided meter labels: the total number of sung bertsos is 37830 and 3140 of them have a meter or melody annotation. Thus, only the 8.3% of the bertsos provide some music information. In addition, we observed that some annotations are not coherent with the real length of the sung bertsos. Therefore, the information provided in these cases must be discarded and the final number of bertsos with correct annotation reduces to 2945, 7.78% of the total number of bertsos. This is a very small number, so we need to devise a method to complete this information and automatically add meter information to the recordings.

As we have previously commented, bertsolaritza is a singing style that uses predefined melodies and syllable structures when singing called *neurria* (meter). Breaking these rules makes the improvised verses worse and usually bertsolaris try not to break them in order to get the maximum score. However, in practice it is very difficult to improvise bertsos that fit perfectly in the predefined melodies.

Therefore, it is common practice to use more than one note in a syllable or to include more than one syllable in one note. We analyzed our database to find how often and in what way singers use syllable adjustments to fit in melody structures. We developed an automatic measure classifier algorithm based on syllable count and measure use, applying Expression 3.1 to perform the classification.

$$z = \arg\min_{i=1}^{n} |x_i - y| \tag{3.1}$$

where n is the number of possible meters with the same number of lines of the sung bertso, y is the vector with the number of syllables in each line of the sung bertso and x_i is the vector with the number of syllables in each line of the meter i (i.e., x_i is the reference vector). We call 'syllable distance' to the distance between each x_i and y.

The melody part of the BDB original database contains 157 different meters. However, only 29 of them are associated to some recording. The meter annotator we have developed only considers these 29 meters and automatically selects for every bertso the candidate meter with the same number of lines and the smallest syllabic distance, as defined in Equation 3.1. Bertso recordings are built with a variable number of bertsos that ranges from 1 to 9. The number of lines that a bertso can contain varies between 4 and 20. We considered bertsos with length of 4, 6, 8, 9, 10, 11, 12, 13, 14 and 16 lines. Taking into account that in the annotated bertsos having 4, 6 and 9 lines only 1 structure is used for each of them, we only applied our automatic measure classifier in bertsos with 8, 10, 11, 12, 13, 14 and 16 lines. The meter annotator uses the syllabic distances to compare between candidates and this method may produce ties between candidates.

In order to evaluate the performance of the meter annotator, we have calculated the accuracy of the algorithm. As the method may produce ties between candidates and there is no other available information to untie the candidates, we have left the tied cases out of the evaluation. We manually labeled a total of 2751 bertsos and use them to evaluate our proposed algorithm. The results can be seen in Table 3.1.1.

Bertso length	Possible structures	Bertsos	Ties	Accuracy (%)
8	6	1259	180	90.79
10	2	997	0	100.00
11	2	25	0	100.00
12	5	148	8	100.00
13	3	11	1	90.91
14	5	108	3	99.07
16	3	184	2	100.00
All	26	2751	194	95.71

Table 3.1.1: Automatic meter classification results

We can observe that bertsos with 8 and 10 lines are very popular as they represent the 82% of the total. The classification of these bertsos is good, although we obtain many ties in the cases of bertsos with a length of 8 lines. These ties appear because there is more variety in possible structures in bertsos with length 8. The annotation in bertsos with other lengths has good scores with few ties. We consider that this algorithm is good enough and have used it to complete the meter information in our database.

3.1.1.7 Database size

After the cleaning and the correction of the transcriptions we measured the size of the raw database. The Bertso database contains 2095 audio files and has a total duration of 59 hours, 10 minutes and 40 seconds. As many speakers and singers are recorded in the database, we defined the following terminology to refer to each subgroup:

- **Participants**: The term relates to all the persons whose voice is included in the database.
- **Hosts**: It refers to the persons that do not sing a whole bertso in all the database. It usually corresponds to the presenters of the shows.

- **Bertsolaris**: It applies to the persons that sing at least one whole bertso in the database.

In the database there are 187 different bertsolaris of which 32 are female and 156 male. There are multiple hosts too, but in the raw database there is no information to measure the total number and genre.

If we consider each line of bertso lyrics an utterance, the database has 45055 utterances. From now on, we will use the term utterance to refer to a bertso line. As the database explained in this section is not segmented and labeled, we are not going to show any distributional information. More detailed information of the singing voice in the database is given in Section 4.5.1.

3.1.2 NUS database

The NUS database is an English speech and singing voice recording database prepared for Speech-to-Singing voice conversion [37]. Music scores are sung and read by the same speaker to use them as parallel information for conversion. The singer are part of the NUS Choir and the amateur vocal community at NUS. As there are not many open access singing voice databases, we used the singing voices from this database to compare different analysis with the results obtained in Bertso database and to evaluate the algorithms developed in this thesis. The singing recordings of the NUS database provide speaker information and phoneme alignments, but there is no music information. As some of the sung melodies are popular, we searched the music scores of these songs and with them created the theoretic phoneme representation of the music scores. The phoneme alignments provided in the database have been corrected to represent the phonemes actually sung by the singers in a faithful way. This correction creates different phonetic representations of the same words, as each singer may have uttered a different pronunciation of the same word in the lyrics. This creates a problem to match the theoretic phonetic transcription of the music score and the phonemes labeled in the recording. This is why we manually modified the phonetic music scores of each recording to adjust them to the sung phonemes. From the 20 different music scores sung in the

database we only have been able to find 16 of them. We have not considered the recordings where the sung music score has not been located.

The resulting music score aligned database includes 12 singers and 39 songs and the total singing voice recording duration is 75 minutes. The recording duration in lines and in minutes per speaker is shown in Table 3.1.2. As each singer sings different music scores and we only obtained 16 music scores out of 20, the amount of recording data for each singer is different. The total duration for each singer varies between 4 and 9 minutes and there are six female singers and six male singers.

Singer	Duration (min)	Lines	Gender
ADIZ	5.69	89	F
JLEE	8.91	88	M
JTAN	5.43	77	M
KENN	3.93	53	M
MCUR	4.10	53	F
MPOL	8.75	104	F
MPUR	5.94	116	F
NJAT	5.26	77	F
PMAR	8.41	88	F
SAMF	5.53	92	M
VKOW	6.69	76	M
ZHY	6.36	115	M

Table 3.1.2: Recording duration for each speaker in NUS database

The duration of the notes for each speaker are shown in Figure 3.1.5. The image shows the boxplot that represents the range between the 25 and 75 percentiles with a colored rectangle. The red line corresponds to the median of the data. The total range of the data is represented with whiskers. We can observe that most values are concentrated in the range below 1 second for all speakers, although longer notes are used in some cases and values as high as 4 seconds are observed. These long notes are usually used at the end of songs and are common in bertsolaritza style too. The note pitch ranges cannot be analyzed because the notes in the phonetically aligned music scores may not be the ones sung by the speakers. Different

speakers can sing the same score in different octaves and we have not manually annotated the actual sung notes. A method to automatically label the musical information knowing the music score is proposed in Section 4.4.2 and the analysis of the resulting note labels in NUS database is presented in Section 4.5.2.

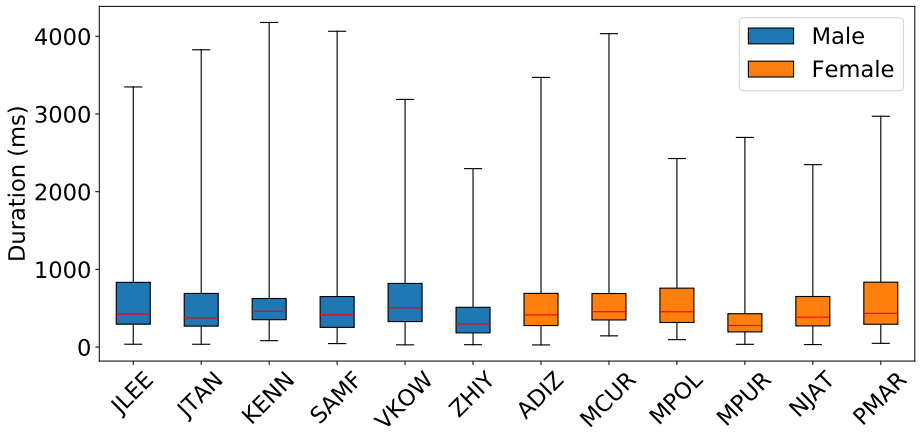


Figure 3.1.5: Note duration boxplot distribution per speaker in NUS database.

3.1.3 NITech database

The Nagoya Institute of Technology (NITech) singing database consists of Japanese children's songs sung by a female singer and has been previously used in the evaluation of HMM and DNN-based singing synthesis systems [109][63]. This database contains 70 songs sung by a female singer but only 31 of these songs can be publicly accessed. The database provides the singing voice recordings with note alignments and musical labels. We have used the alignments and the labels to create structured music scores.

The reduced version of the database of 31 songs contains 28 minutes of voice distributed in 392 music lines. The distribution of note duration and pitch values can be seen in Figures 3.1.6 and 3.1.7 respectively.

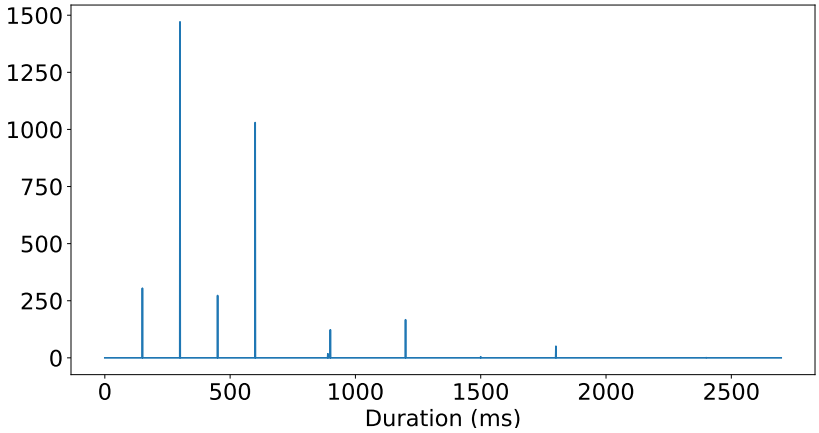


Figure 3.1.6: Note duration distribution in NITech database

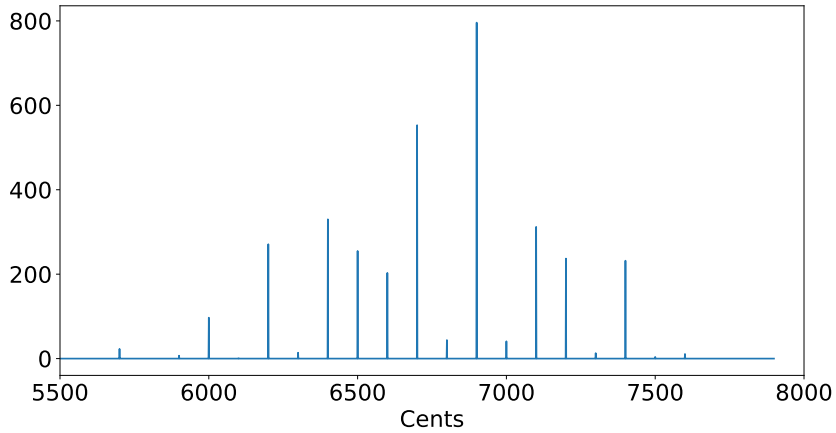


Figure 3.1.7: Note pitch distribution in NITech database

The duration values provided in the labels are not continuous, because they have been discretized to represent a musical score with the most commonly used symbols. In the pitch distribution, we can see that the range of the singer goes from 5700 cents to 7700 cents, which are equivalent to A_3 and F_5 respectively.

3.2 Software

In this thesis new software has been created to analyze, annotate and synthesize singing voice, but many open source external software tools have been used as well as a base for many functionalities. These tools have constituted a valuable assistance in the development of this thesis and they must be properly acknowledged, addressed and referenced. This section list and briefly describes these tools.

3.2.1 HTK

Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and using HMM [164]. Originally created for speech processing, it has been used in many areas that need HMM processing like Deoxyribonucleic acid (DNA) sequencing. The tool has scripts for audio feature extraction, HMM model training, decoding and evaluation. We used the HTK toolkit to force align the phonemes in the singing voice recordings in the single singer method presented in Section 4.3.1.

3.2.2 Kaldi

Kaldi is a speech recognition package that provides facilities to develop HMM and Neural Networks based applications [115]. Audio feature extraction, model training, decoding and evaluation tools can be used. A large online community is available for support of Kaldi's users. We used the Kaldi toolkit to force align the phonemes in the singing voice recordings of Bertso database with singer adaptation in Section 4.3.2.

3.2.3 Merlin

Merlin is a Neural Network speech synthesis toolkit [163]. The toolkit includes feature and linguistic label normalization tools and pipelined neural networks for

training, synthesis and evaluation. The DNN synthesis system we created in Section 5.7 is a modified versions of this toolkit.

3.2.4 Sonic Visualizer

Sonic Visualizer is a music audio file annotation Integrated Development Environment (IDE) [23]. Its command line tool Sonic Annotator has multiple signal processing and feature extraction libraries that include Tony note annotation algorithm [93] and YIN pitch extraction algorithm among others. The Tony algorithm uses HMMs with fixed transitions and a note grammar with threes states to detect notes from the f_0 curve. In Section 4.4 we compare our note detection algorithm with Tony algorithm in different ways.

3.2.5 Ahocoder

The Ahocoder parametrization [43] uses autocorrelation method [19] with Quasi Harmonic model refinement to calculate the f_0 of an audio signal. Using the harmonics of the f_0 in each analysis frame, Maximum Voiced Frequency (MVF) is defined using Sinusoidal Likeness Measure (SLM) [120]. The MVF is the higher harmonic frequency of f_0 in which the similarity of the harmonic with a pure sinusoid is higher than an empirical threshold. After defining MVF, cepstral coefficients are calculated. In signal reconstruction f_0, Mel-cepstral Coefficients (MCEP) and MVF are used. The spectrum envelope is reconstructed using the f_0 and the MCEP and in the aperiodic part of the spectrum, above the MVF, white noise is added to the spectrum.

3.3 Chapter conclusion

In this chapter, we characterized the Bertso database and described all the open access singing voice databases that can help us in this work, the NITech and the NUS databases. In the Bertso database, we adapted and normalized the audio, text

and music score files to standardize the data parameters. We also designed an algorithm to automatically detect the meter of bertsos with very good results. In external databases, format adaptation and data analysis have been done too. Both external databases provide phoneme alignments but only one of them has note annotation information. This is why we analyzed note duration in both of them but pitch analysis has been performed only in NITech database. The missing analyses are going to be obtained in Section 4.5 after the automatic annotation phase explained in Chapter 4. Finally, we defined the most important software we have used in this work as a base for the development and preparation of our own tools. The preparation of the Bertso database has been published in [128].

4

Data labeling

In the previous chapter we explained the problematic due to the presence of speech from different hosts, applause and environment sounds in the recordings of bertsolaritza. In the available transcriptions only the singing voice is transcribed, therefore, non-singing segments must be removed for proper processing and labeling. We have developed and compared different audio segmentation and classification systems in a data-set specifically created and labeled for this purpose, as explained in Section 4.1.

Once the new audio files containing only singing voice have been obtained, the labels to train the singing voice synthesis system must be created. These labels must combine linguistic information and music information. For the linguistic part, we propose a method to segment the database into utterances in Section 4.2 and into phonemes in Section 4.3 using forced alignment with the phonetic transcriptions. For the music related component, musical annotation methods using

and without using bertso melodies have been developed and tested in Section 4.4. The whole structure of the labeling procedure developed can be seen in Figure 4.0.1. After using the segmentation and labeling systems, we annotated the Bertso database and NUS database and analyzed the general characteristics of the labels in these databases in Section 4.5. We finalize the chapter drawing the main conclusions in Section 4.6.

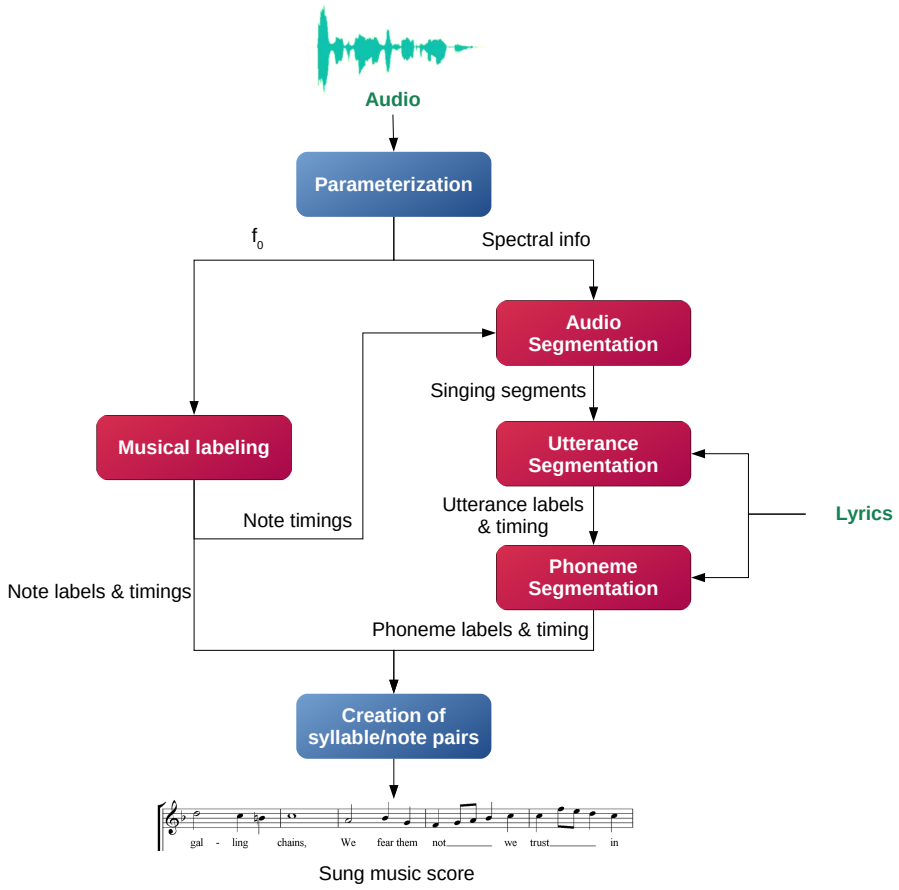


Figure 4.0.1: Overview of the data labeling procedure

4.1 Audio segmentation

The final objective of the database preparation procedure is to align the phonetic transcription of the lyrics with the audio recordings. However, the bertso audio recordings contain not only the singing voice, but also other acoustic elements like applause, speech and music. Therefore, the first step in the database preparation procedure must be the location and separation of the segments containing singing speech. Examining some sample audio files from the database, we found the following types of content: singing voice (sn), speech (sp), silence(sl), applause(a) and music (m). Applause also appears combined with other types of audio: we found applause overlapped with the singing voice (sna), with speech (spa) and with music (ma). Observing the audio files we took these decisions:

- We excluded the recordings with background music because they are very scarce and the continuous music background can cause errors in the alignments.
- We decided to remove all the speech segments that appear in the recordings.
- We kept the singing parts that overlap with applause.

We manually excluded the recordings with background music at the beginning of the process. After that, we created and automatic segmentation system to segment the audio files and extract the segments of interest.

4.1.1 Proposed segmentation system

The architecture of our speech/singing segmentation system is shown in Figure 4.1.1. In the first step, a GMM-HMM VAD is applied to theMel Frequency Cepstral Coefficient (MFCC) features to locate the segments that contain voice. Then, the pitch contour is extracted and a smoothing process is applied on it to remove

vibrato effects. We identify the segments of the smoothed pitch contour that correspond to musical notes using our algorithm described in Section 4.4.3.1 and, making use of this information, we calculate the voicing and note percentages in each segment. Finally, a statistical classifier is used to assign the correct class to these percentage parameters.

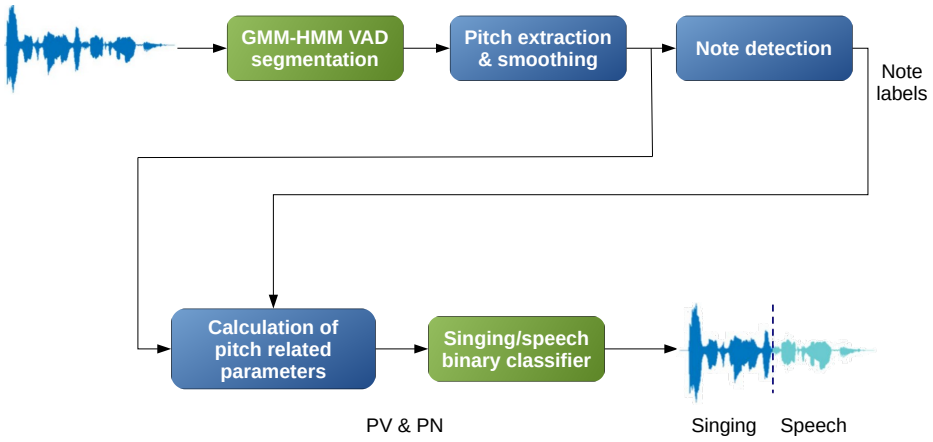


Figure 4.1.1: Structure of the proposed speech/singing voice segmentation system.

4.1.1.1 GMM-HMM based VAD

In the VAD of the proposed system, three possible classes are defined as output in each acoustic frame: voice, applause and silence. Considering that we manually excluded the recordings with background music, the only overlap of classes that can be found in the recordings are the singing voice with applause or speech with applause. We considered these overlaps as voice. We decided to split applause and silence into two separate classes because applause is common in our Bertso database and the acoustic nature of both classes is very different. The classification of the recordings is made using a frame-level GMM with an HMM post-smoothing [61]. We used 13 MFCC values with Δ and Δ^2 values calculated applying a 25 ms window and 10 ms frame period. For the initial frame classification, independent

GMMs are trained per each class using Expectation Maximization [33]. The optimal number of gaussians will be selected with a VAD experiment described in Section 4.1.1.5. Frames are classified using these GMMs, but this frame level classification can create fast label changes that do not fit well to the data. To avoid this, we have used an ergodic HMM of three states, one per class. This HMM forces the frame labels to remain in the same class for a minimum duration depending on the likelihood values of the GMMs. We used a probability of 0.0001 outside the transition matrix diagonal for this purpose. To classify the segments, the likelihood of observation provided by each model is calculated using expression (4.1):

$$P(o|s_i) = \sum_{j=1}^{M} w_{ij} N(o|\mu_{ij}, \Sigma_{ij}), \tag{4.1}$$

where o is the MFCC vector, w_{ij}, μ_{ij}, Σ_{ij} are the weight, mean and diagonal covariance of the component j of the state s_i and M is the number of Gaussian components. A representation of the proposed VAD method can be seen in Figure 4.1.2.

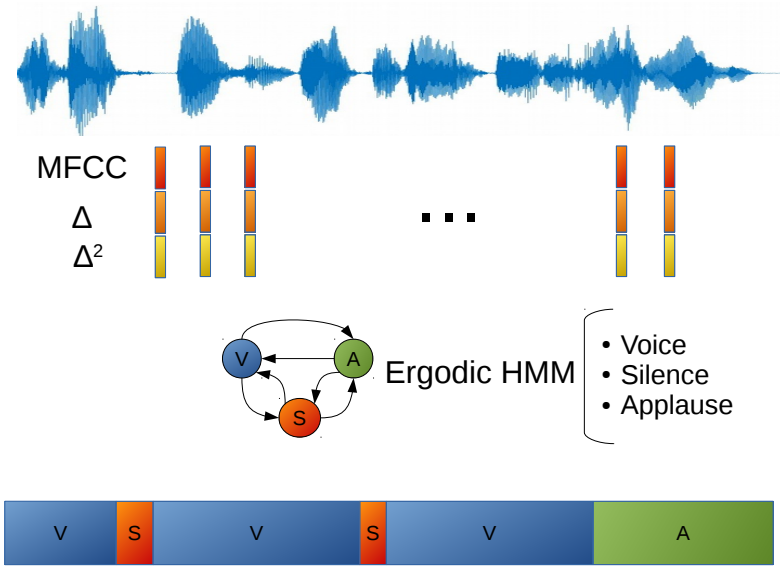


Figure 4.1.2: Scheme of the proposed VAD system

4.1.1.2 Speech/singing classification

In our Bertso database, the separation of speech and singing voice segments is clear, i.e., there are no adjacent boundaries between the two classes. This is why we address the problem as a binary classification of the voice segments detected by the VAD. The pitch parameters we propose for the classification of each segment are: proportion of voiced frames (PV) and percentage of voiced pitch frames labeled as a musical note (PN).

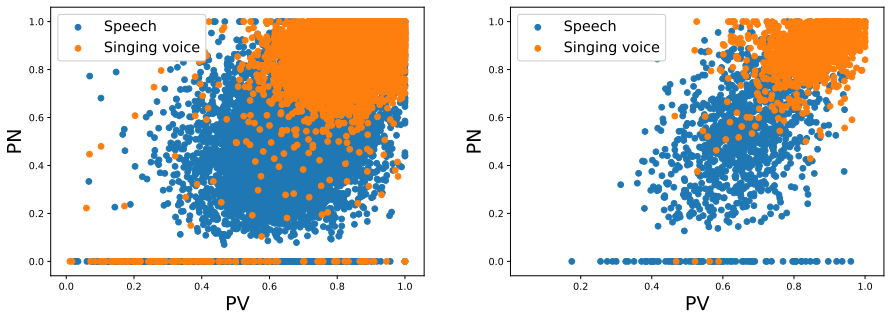
The pitch curve has been calculated using PRAAT autocorrelation method [19] with a frame period of 10 ms. Voiced/unvoiced segments are obtained directly from the pitch curve where the relative value of maximum autocorrelation is used to take this decision. Stable musical note segments are found using our algorithm explained in Section 4.4.3.1. The features for classification are calculated according to expressions (4.2) and (4.3):

$$PV = \frac{N_{VF}}{N_T}, \tag{4.2}$$

$$PN = \frac{N_{NF}}{N_{VF}}, \tag{4.3}$$

where N_{VF} is the total number of voiced frames, N_{NF} is the total number of frames labeled as a musical note and N_T is the total number of frames, all of them calculated within the segment to be classified.

Figures 4.1.3a,b show the distribution of the proposed classifying features PV and PN in Bertso and NUS databases described in Sections 3.1.1 and 3.1.2 respectively. In both cases, speech presents a more scattered distribution than singing voice. However, good discrimination can be achieved when considering both parameters at the same time. As a final step of the proposed algorithm, a classifier has to be trained with the vector containing these two parameters to obtain the final speech/singing classification. In the proposed system we have used an SVM [132, 148].



(a) **(b)**

Figure 4.1.3: Distribution of PV and PN parameters. (**a**) Bertso database (**b**) NUS database

4.1.1.3 Used databases

There are very few publicly available datasets that contain monophonic singing (MIR-1K [67] and the Singing Voice Audio Dataset [18] for instance) and even fewer datasets that contain both spoken and singing speech. In fact we have only found the NUS Sung and Spoken Lyrics Corpus [37], that is not completely suitable for our task because the files it contains are mono-class, i.e, they contain either speech or singing. Although the NUS database contains recordings with only singing voice or speech, as we focused our problem on voice segment classification, the database may be used for our experiment. Therefore, we have two databases to test our algorithms, the Bertso database explained in Section 3.1.1 and the NUS database explained in Section 3.1.2. The Bertso database is a large database, there are no open source singing voice databases with this size. As the manual labeling of the databases is highly time consuming, we prepared and labeled an smaller representative excerpt of the Bertso database. We named this excerpt as Bertso excerpt database and used it to train a first version of our speech and singing voice segmentation system. We used this first version to help us in the labeling of the whole Bertso database. The main characteristics of each database are summarized in the next sections.

4.1.1.3.1 Bertso excerpt database

The Bertso excerpt database is a subset of the Bertso database with 20 audio files from 19 bertsolaris with a total duration of 60 minutes and 40 seconds. These audio files contain 32.8 minutes of singing voice and 2.87 minutes of speech. In this excerpt the singing voice has longer durations than speech. The mean duration of singing segments and speech segments are 3.69 and 1.51 seconds respectively. The 20 files were selected to cover the variability of the original Bertso database, considering recordings from different decades and gender.

Table 4.1.1 shows the distribution of bertsolaris and hosts by gender in the Bertso excerpt database.

	Bertsolaris	**Hosts**	**Hosts who sing**	**Total**
Female	7	6	2	15
Male	12	6	2	20
Total	19	12	4	

Table 4.1.1: Number of hosts and bertsolaris in Bertso excerpt database

The distribution of the proposed classifying features *PV* and *PN* in the Bertso excerpt database can be seen in Figure 4.1.4. The behavior is very similar to that observed in Figures 4.1.3a and 4.1.3b with speech more scattered than singing voice.

For the experiments, we split the Bertso excerpt database into 10 subsets for cross-validation tests. All the partitions considered include different participants in the train and test subsets.

4.1.1.3.2 NUS database

The NUS database has been described in Section 3.1.2, but the description only included the analysis of the singing voice. In addition, only the singing voice recordings with available music score have been analyzed there. For the speech/singing discrimination task no original music score is needed. In this database, each record-

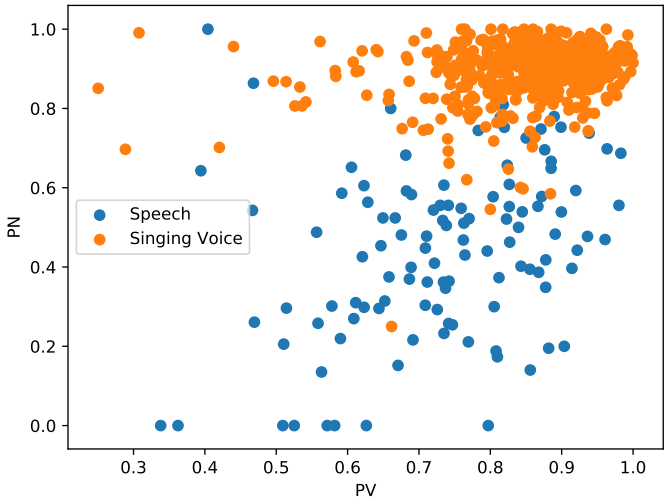


Figure 4.1.4: Distribution of the classes in the Bertso excerpt database

ing contains either speech or singing voice and we used the VAD defined in Section 4.1.1.1 to obtain the voice segments and labeled them with the type of the recording. The number of segments of each class and their duration are presented in Table 4.1.2.

Segment type	**N. of segments**	**Total duration (min)**
Singing	1437	92.51
Speech	1283	40.80
All	2720	133.31

Table 4.1.2: Quantity and duration of voice segments in NUS database

The duration distribution of the speech and singing voice segments in the NUS database can be seen in Figure 4.1.5b. The average duration and standard deviation of the singing voice and speech are 3.83 ± 2.21 and 1.92 ± 1.06 seconds respectively. Considering that in the NUS database the linguistic content is the same in singing and speech segments, we can clearly see that singing voice segments are longer than speech segments.

4.1.1.3.3 Bertso database

The Bertso database contains 2095 Basque audio files from 187 different bertsolaris and has a total duration of 59 hours, 10 minutes and 40 seconds. To label the database, we first trained the system proposed in Section 4.1.1.2 with the Bertso excerpt database and used it to segment the whole Bertso database. After the automatic segmentation, the segmentation errors have been manually corrected so that a bigger gold standard is available to measure the performance of our proposed segmentation algorithm.

In fact, we have no need to create a segmentation system once we already have the whole database manually segmented. However, all the tools produced in this thesis are prepared to process new bertso recordings that will be created in the future. We think that the evaluation of the proposed systems in a bigger and more diverse database provides better information about the limitations and advantages of the algorithms. The number of segments of each class and their duration in Bertso database are shown in Table 4.1.2 and the length distribution of each class is visualized in Figure 4.1.5a.

Segment type	**N. of segments**	**Total duration (hours)**
Singing	35419	53.55
Speech	1283	5.80
All	44463	59.35

Table 4.1.3: Quantity and duration of voice segments in Bertso database

The metadata provided with the recordings include information about all the bertsolari identities, but the host identity is only annotated if he or she also sings to introduce the topic for the verses. It was not feasible to manually label the identities of the rest of hosts, therefore we decided to use an approximate method to create host identity labels for the missing metadata. The recordings are separated into sessions that correspond to different places and dates. We decided to assign the same host identity to all the recordings of the same session, as usually there is only one host in each bertsolaritza show.

We also classified the genre of the hosts by applying a threshold to the average value of the f_0 in such a way that hosts with an average f_0 value higher than the threshold are considered female and male otherwise. The optimal threshold to define the genre of a host has been defined using the hosts whose identity was available in the metadata (31 different hosts in 54 sessions, with a total duration of 15 minutes and 5 seconds). Table 4.1.4 shows the classification results obtained using different thresholds, where 250 Hz is the value that gets better F-score, so we have used this value to automatically label the missing genre of hosts.

The genre classification and host identification algorithms have been used to process the audio files with missing metadata to complete the information. The final database contents are shown in Table 4.1.5. In most cases participants either sing or act as host in each file, but as we have already commented sometimes the hosts give the first foot for the improvised verses singing as well. We also can find bertsolaris acting as host in some recordings. This is why we created two different categories for hosts that have singing segments and bertsolaris that have speech segments in the database.

Threshold (Hz)	**Precision**	**Recall**	**F-score**
100	0.13	0.5	0.21
150	0.66	0.63	0.44
200	0.77	**0.84**	0.77
250	0.88	0.77	**0.81**
300	**0.9**	0.64	0.67
350	0.37	0.5	0.43

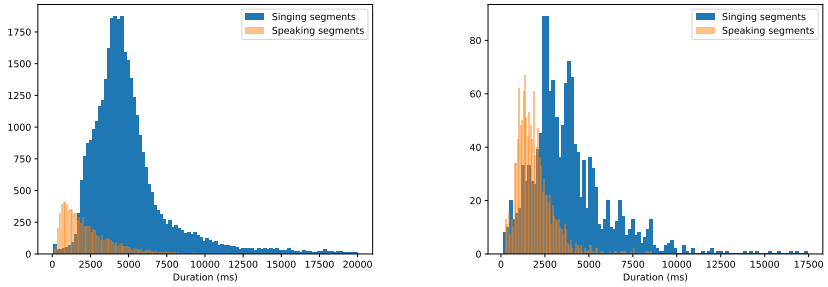
Table 4.1.4: Results of the automatic genre classification

The duration distribution of the speech and singing voice segments in the Bertso database can be seen in Figure 4.1.5a. The average duration and standard deviation of the singing voice and speech are 5.03 $\pm$ 2.67 and 2.30 $\pm$ 1.74 s seconds respectively. In the Bertso database the speech and singing voice do not contain the same linguistic content, but we can observe that the distributions follow a similar pattern

	Bertsolaris	Hosts	Bertsolaris who host	Hosts who sing	Total
Female	33	43	1	9	86
Male	140	528	13	28	709
Total	173	571	14	37	

Table 4.1.5: Number of hosts and bertsolaris in Bertso database

to the one found in the NUS database.



(a) Bertso database **(b)** NUS database

Figure 4.1.5: Distribution of singing and speech segment durations

4.1.1.4 Other speech/singing discrimination methods

To compare our algorithm with other methods we have selected methods that are suitable to work with segments of different duration as it is the case of Bertso database. On the one hand, we have trained GMM classifiers with the parameters suggested in [105] (Δf_0) and [150] (DFT-f_0). On the other hand, we have also built a GMM classifier based on MFCC parameters. These baseline methods are explained with more detail in the following subsections.

4.1.1.4.1 DFT-f_0

As commented before, f_0 provides very useful information to discriminate between speech and singing. The histogram of f_0 gives information about the range and the

distribution of f_0 values, which for singing voice will be concentrated around the frequencies corresponding to musical notes. In [150] GMMs are used to model the DFT of the f_0 distribution and detect the deviation of the instantaneous pitch value from its mean. f_0 values in Bertso database range from 75 to 500 Hz. A 100 bin histogram is calculated, therefore each segment corresponds to approximately 0.027 octaves. The histogram is normalized to have unit area and then modeled using 8 component GMMs for singing voice and speech.

4.1.1.4.2 Delta-f_0

The dynamics of f_0 are another feature that has been considered in speech and singing voice discrimination. In [105] the Δf_0 distribution of voiced segments is modeled with GMMs to discriminate speech and singing voice. We calculate the Δf_0 using a Savitsky-Golay filter [131] with a window of 50 ms. An histogram of 100 bins is made from -50 to 50 Hz. The distribution of Δf_0 in each voice segment is normalized to have unit area and then modeled with 16 component GMMs.

4.1.1.4.3 MFCC

In [105] short-term spectrum features are used motivated by the presence of an additional resonance characteristic of singing speech as addressed in [143]. We calculated 13 MFCC coefficients and their Δ with a frame period of 10 ms and a window of 25 ms applying a Cepstral Mean and Variance Normalization (CMVN) file-wise normalization. MFCC frames of speech and singing voice are modeled using 32 component GMMs. Each voice segment is assigned the class that gets the higher sum of log-likelihood for all the frames of the segment. We chose GMMs to model MFCC parameters due to the high dimensionality of the parameters.

4.1.1.4.4 Tony

Our proposed singing/speech classification method includes the use of our own note detection algorithm. To assess the effect of the note detection method, we decided to test our proposed speech and singing voice discrimination system using

an alternative note detection algorithm. The alternative note annotator we have selected is Tony, a state of the art note detection algorithm [93]. In summary, this new discrimination method is identical to the proposed one but Tony is used for note labeling instead of the algorithm explained in Section 4.4.3.1.

4.1.1.5 Results

4.1.1.5.1 Results of GMM-HMM VAD

We tested the VAD only in the Bertso database excerpt. The metric used to assess the VAD is the voice detection F-score defined as indicated in Equation 4.4.

$$F - score = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

where TP is the duration of speech classified as speech, FP is the duration of non-speech classified as speech and FN is the duration of speech classified as non-speech.

Table 4.1.6 shows the results for different number of Gaussian components. All of them get good results, over 0.96, and the number of components does not affect the performance significantly. We have selected the VAD with 32 components for the classification experiments.

Gaussians	F-Score
2	0.965 +/- 0.005
4	0.967 +/- 0.006
8	0.969 +/- 0.007
16	0.972 +/- 0.008
32	0.973 +/- 0.008
64	0.974 +/- 0.008

Table 4.1.6: Results of the GMM-HMM VAD for different number of Gaussian components

4.1.1.5.2 Results of speech and singing discrimination

We compared the proposed system with the ones explained in Section 4.1.1.4. To assess the generalization capability of the algorithm two different experiments have been performed: Bertso excerpt experiment and Bertso experiment.

In the Bertso excerpt experiment we split the Bertso excerpt database in 10 subsets for cross-validation tests. The final result is obtained joining all the test results. All the partitions considered include different participants in the train and test subsets. The NUS database is classified using the algorithms trained with the whole Bertso excerpt database. In this experiment we did not include the version of the proposed system with Tony note detection.

In the Bertso experiment we have used the 5x2cv paired *t* test [35] in the Bertso database with the structure shown in Figure 4.1.6. To achieve this, we first split the Bertso database into 10 sections with the only condition that all the segments corresponding to one specific participant must be in the same section. After this, we made five iterations of splitting the database into two blocks of 5 sections each, chosen randomly, and with no repetition of sections within the blocks. In each of the iterations, we obtained a first score by using one block as train and the other as test. A second score is calculated by rotating test and train: training set becomes test set and vice versa. This gives us 10 scores which are averaged to calculate the final score. The procedure ensures that no participant is present both in the training and test block in any iteration.

In the experiments, no model adaptation has been used to classify the segments from the NUS database. The performance of the classifiers has been measured using unweighted Precision, Recall and F-score, defined as the macro-averaged measure parameter for each of the classes. We have used the unweighted mean of the score because our Bertso database is heavily imbalanced, with more singing segments than speech segments as seen in Section 4.1.1.3.1. Macro-averaging considers all classes equally and is more convenient in the case of imbalanced datasets [141].

The results of the Bertso excerpt experiment for the classification of the Bertso

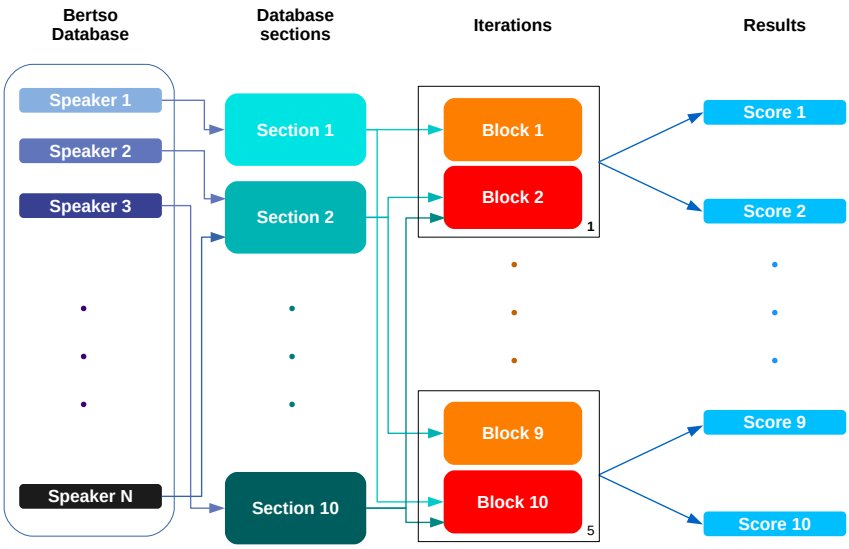


Figure 4.1.6: 5x2cv test structure

excerpt with cross-validation and the NUS database are shown in Table 4.1.7. The proposed method gets the best results and can compete with MFCC coefficients classified by a GMM. The methods that use pitch derived parameters as distribution of Delta-f_0 and DFT-f_0 get poorer results due to the short duration of the segments to classify. Although in other works they have proved useful, they are not suitable for the characteristics of the Bertso database. The experiment in the NUS database shows similar results proving the validity of our method with professional singers and a different style. The MFCC method results got worse for NUS database, probably because the audio files used to generate the models contain both speech and singing and the files in NUS database belong to one class. Therefore the normalization process affects both databases differently.

The results of the Bertso experiment for the 5x2cv test of the Bertso database and the NUS database have been separated in two tables. Table 4.1.8 shows the

Method	Precision		Recall		F-score	
	B. excerpt	NUS	B. excerpt	NUS	B. excerpt	NUS
Δf_0[105]	0.78	0.76	0.83	0.74	0.80	0.74
DFT-f_0[150]	0.73	0.77	0.77	0.76	0.75	0.77
MFCC[105]	**0.95**	0.75	0.85	0.66	0.89	0.64
Proposed	0.91	**0.89**	**0.93**	**0.89**	**0.92**	**0.89**

Table 4.1.7: Results of speech/singing classification in the Bertso excerpt experiment

results of the singing voice segment classification and Table 4.1.9 presents the results of the speech segment classification. The method using MFCC coefficients classified by GMM is the best method for the classification of Bertso database in both singing voice and speech segments, although differences in results with our proposed method are not statistically significant (see Table 4.1.10). This MFCC method also gets very good results for the singing class in the NUS database, comparable to our method. In the Bertso database, singing and speech are mixed in the same file, while, in the NUS database, each file contains only one type of voice. In the MFCC method, file-wise mean and variance normalization is applied for the calculation of the MFCCs. If there are enough data with high participant variability in the training database, the MFCC method can learn the characteristics of both classes and generalize to other databases. However, if data are not enough, like in the Bertso excerpt experiment, this generalization is not good as it can be seen in Table 4.1.7. This bias is due to the unbalance in favor of the singing class in the training data.

The pitch-based methods, Delta-f_0 and DFT-f_0, do not get good results compared with our system in both experiments. As commented before, this is due to the presence of short voice segments in the database. These pitch based methods are suitable when long voice segments are present.

Regarding the use of a different note detection algorithm, Tony method gets worse F-score values than the proposed method in the Bertso and NUS databases. Tony only gets higher scores than the proposed method in the NUS database re-

garding recall for singing and precision for speech. The note detection algorithm of Tony is designed to analyze singing voice and not speech. Therefore, it is likely that it has a strong bias towards identifying notes even in speech.

Method	**Precision**		**Recall**		**F-score**	
	Bertso DB	NUS	Bertso DB	NUS	Bertso DB	NUS
Δf_0[105]	0.95 ± 0.00	0.79	0.85 ± 0.03	0.73	0.90 ± 0.01	0.76
DFT-f_0[150]	0.93 ± 0.00	0.76	0.83 ± 0.01	0.77	0.88 ± 0.01	0.76
MFCC[105]	$\mathbf{0.99 \pm 0.00}$	0.88	$\mathbf{0.97 \pm 0.00}$	**0.97**	$\mathbf{0.98 \pm 0.00}$	**0.92**
Tony	0.97 ± 0.00	0.83	0.92 ± 0.01	0.95	0.94 ± 0.01	0.88
Proposed	0.98 ± 0.00	**0.92**	0.96 ± 0.00	0.92	0.97 ± 0.00	**0.92**

Table 4.1.8: Results of singing classification in the Bertso experiment

Method	**Precision**		**Recall**		**F-score**	
	Bertso DB	NUS	Bertso DB	NUS	Bertso DB	NUS
Δf_0[105]	0.58 ± 0.04	0.72	0.81 ± 0.01	0.78	0.68 ± 0.03	0.75
DFT-f_0[150]	0.53 ± 0.01	0.73	0.76 ± 0.01	0.72	0.63 ± 0.01	0.73
MFCC[105]	$\mathbf{0.90 \pm 0.01}$	**0.96**	$\mathbf{0.97 \pm 0.00}$	0.85	$\mathbf{0.93 \pm 0.01}$	0.90
Tony	0.73 ± 0.04	0.93	0.88 ± 0.00	0.78	0.80 ± 0.02	0.85
Proposed	0.87 ± 0.02	0.91	0.94 ± 0.00	**0.91**	0.90 ± 0.01	**0.91**

Table 4.1.9: Results of speech classification in the Bertso experiment

To assess the statistical significance of the results, we have calculated the p-value for the results of all the alternative systems when compared with our proposed system. Table 4.1.10 shows the p-values for speech and singing detection. Considering these values and a significance level of $\alpha = 0.05$, the differences in performance of the proposed system are statistically not significant comparing to the MFCC system and they are significant comparing it with the rest of the systems.

Method	p (Singing)	p (Speech)
Δf_0[105]	8.099×10^{-5}	2.858×10^{-5}
DFT-f_0[150]	0.010	0.001
MFCC[105]	0.084	0.063
Tony	0.004	0.001

Table 4.1.10: p-value of the results of the proposed algorithm compared with the rest of the systems

4.1.1.5.3 Analysis of computation time

We measured the time needed by each speech/singing discrimination method to train and classify the Bertso database using 5x2cv cross-validation. The processes have been run in an Intel Xeon CPU E5-2660 v2. The results obtained are shown in Table 4.1.11.

We can see that all processing times are comparable, except for the one of the GMM built with MFCC. We can see that the slightly better results achieved by the MFCC system are produced at the expense of bigger dimensionality and computation time. In consequence, the proposed system is the best classification method overall.

Method	Time for training (h:mm:ss)	Time for classification (h:mm:ss)
Δf_0[105]	**0:04:03**	0:03:59
DFT-f_0[150]	0:04:15	0:03:51
MFCC[105]	7:12:33	0:17:04
Tony	0:04:31	0:04:13
Proposed	0:04:08	**0:03:46**

Table 4.1.11: Computation times for training and classifying in the Bertso experiment

4.1.2 Analysis of the segmented Bertso database

In this section we have described the method proposed for the labeling of the singing and speech segments in the Bertso database. The aim of singing voice de-

tection is to remove all the elements that are not singing voice from the audio files. The isolation of singing voice segments is essential to obtain phonetic and note alignments in an automatic way because only the singing voice segments are transcribed in the database. However, although the singing voice segments are separated in the database, each segment does not correspond to an utterance. Therefore we cannot separate the recordings into utterances using the identified singing segments. We decided to develop a method to clean the audio files in an automatic way, so that they are valid to be used in the next processing phase. The new clean version of each recording is the segment that starts from the beginning of the first singing segment and ends at the end of the last singing segment of the recording. Before the cleaning we decided to define certain conditions of the recordings to simplify future steps and create a more coherent dataset. The conditions of the recordings for the cleaning are the next ones:

- **Singing continuity**: There can be no speech segment between the first singing segment and the last singing segment in the file.
- **Bertsolari continuity**: The bertsolari that improvises has to be unique in all improvised singing segments.
- **Singing host continuity**: As explained in Section 2.8.4, in some exercises the host sings the first or last foot of the bertso and the bertsolari has to improvise the rest of the bertso. In this type of recordings only a unique host that sings is accepted.
- **Melody continuity**: Every bertso in the recording has to be sung with the same melody.
- **Meter continuity**: Every bertso in the recording has to have the same meter.

The selected recordings will then have a unique bertsolari, unique meter, unique melody and in case it appears, a unique singing host. We also removed from the selection some recordings with good quality but with incoherent transcriptions or

recordings that are cut before the end of the improvisation. If we consider the performance types explained in Section 2.8.4, in the selection we have only solo performances that have no multiple melodies, metrics and bertsolaris in the recording. The resulting database contents are presented in Table 4.1.12.

At this point we still do not know if each singing segment corresponds to a bertsolari or to a singer host. To estimate the recording times, we have used the proportion of utterances. The total time is proportionally divided between both bertsolari and singer host in the same proportion as are the utterances.

Singer type	N. of singers			N. of utterances			Voice segment dur. (min)		
	Male	Female	All	Male	Female	All	Male	Female	All
Bertsolari	147	29	176	35152	5237	40389	2483.42	385.23	2868.65
Singing host	114	30	144	922	286	1208	62.20	19.53	81.73
All	261	59	320	36074	5523	41597	2545.62	404.76	2950.38

Table 4.1.12: Number of singers, number of utterances and total duration (min) of the selected voice segments.

4.2 Utterance segmentation

As explained in Chapter 3 each of the recordings of Bertso database contains a single song structured in bertsos (strophes) and utterances. Having recordings of different length and with length of several minutes makes the processing and the indexing of information inefficient. In speech, it is possible to divide the orthographic transcriptions into syntactical sentences, but in singing lyrics sentences are not clearly defined. The singing voice prosody is mainly defined in the music score, although each singer can interpret a music score with variations. The structure of lyrics, divided in utterances in western music, gives the singer a rhythm structure and also a break to breath between the utterances. In bertsolaritza, this silence between utterances is even clearer than in popular music. Taking this into account, we decided to segment the recordings into utterances defined by transcriptions. Doing this in the first place creates a database closer to standard speech databases,

making it easier to use utterance centered speech processing software. In addition, this division of the singing signals into utterances makes the database more flexible for recurrent and parallel processing.

We used the phoneme segmentation in audios with multiple utterances to define the utterance segmentation. The phoneme segmentation may be used to define utterance segmentation marks between the end of the last phoneme from each utterance and the start of the first phoneme of the next utterance.

As the database contains singing voice, we considered the differences between the phoneme alignments of speech and singing voice. In the original BDB database, every recording comes with the corresponding orthographic transcription of the lyrics. In the Bertso database the 33.32 % of the recordings has the melody labeled, but we do not consider this melody as the gold standard music score of the recording. Bertsolaritza is a improvised art, lyrics are created live with the only restriction of meter and the singers choose one of many predefined melodies with the predefined meter to sing the created lyrics. The bertso scenario creates different distortions in the original melodies. First, the improvisation produces meter mismatches and different interpretations of the same melody. Second, the fact that the majority of bertsolaris are not professional singers creates tuning errors. Finally, although the note durations follow some standards, the singing is a cappella and the tempo precision is not needed; therefore it is not easy to predict the duration of each note. The relation between the music scores and recordings in the Bertso database is analyzed in a deeper way in Section 4.4.2.

Knowing that the provided music score information is not a gold standard information, our approach of the phoneme segmentation is very similar to the one used in speech, both at system and feature levels. Nevertheless, we tested different phoneme groupings considering words and syllables.

We used Kaldi [115] to align the phonemes. The procedure of segmentation consists of four phases:

- Monophone training: Acoustic features are modeled for phonemes without context.

- Triphone training: Models from the monophone training are used to generate models for triphones, using side context in each phoneme.
- Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) training: LDA transformation and MLLT model-adaptation are applied to the features and new triphone models are created.
- Singer-adapted triphone models: New models are created using Feature space Maximum Likelihood Linear Regression (fMLLR) transformation for the features of each singer.

The acoustic parameters used for modeling are 13 MFCCs extracted with a 25 ms window and a 10 ms frame rate. Singer-wise Cepstral Mean and Variance Normalization is also applied. Delta and Delta-delta features are used in the first steps.

The phonetic transcription of the utterances uses word context rules to take into account co-articulations. In the alignment, we used position-dependent phonemes with different configurations. The position-dependent phonemes have four alternative models for each phoneme depending on the relative position within specific phoneme groups called tokens. In speech recognition, the tokens used as reference are words because of the way speech is constructed [140]. The four models created for each phoneme with position information are the Begin, End, Internal and Single phonemes. The single position is used for the tokens with a single phoneme, singletons.

- Begin: Phoneme is located at the begging of a token.
- Internal: Phoneme is located inside a token.
- End: Phoneme is located at the end of a token.
- Single: Phoneme is a token itself.

In singing voice, the notes have more phonetic relevance than words and the syllables are closely related to notes [59][78][40]. In bertsolaritza, the meter is

an initial condition for improvisation and it is important to adjust the syllables to the meter to adjust the created lyrics to the predefined melodies. Considering this relation we decided to analyze the phoneme alignment system using three different tokens as reference for position-dependent phonemes: word, syllable and phoneme. In the alignment with the phonemes as reference token, each phoneme is considered a singleton and therefore it is equivalent to an alignment without position-dependent information.

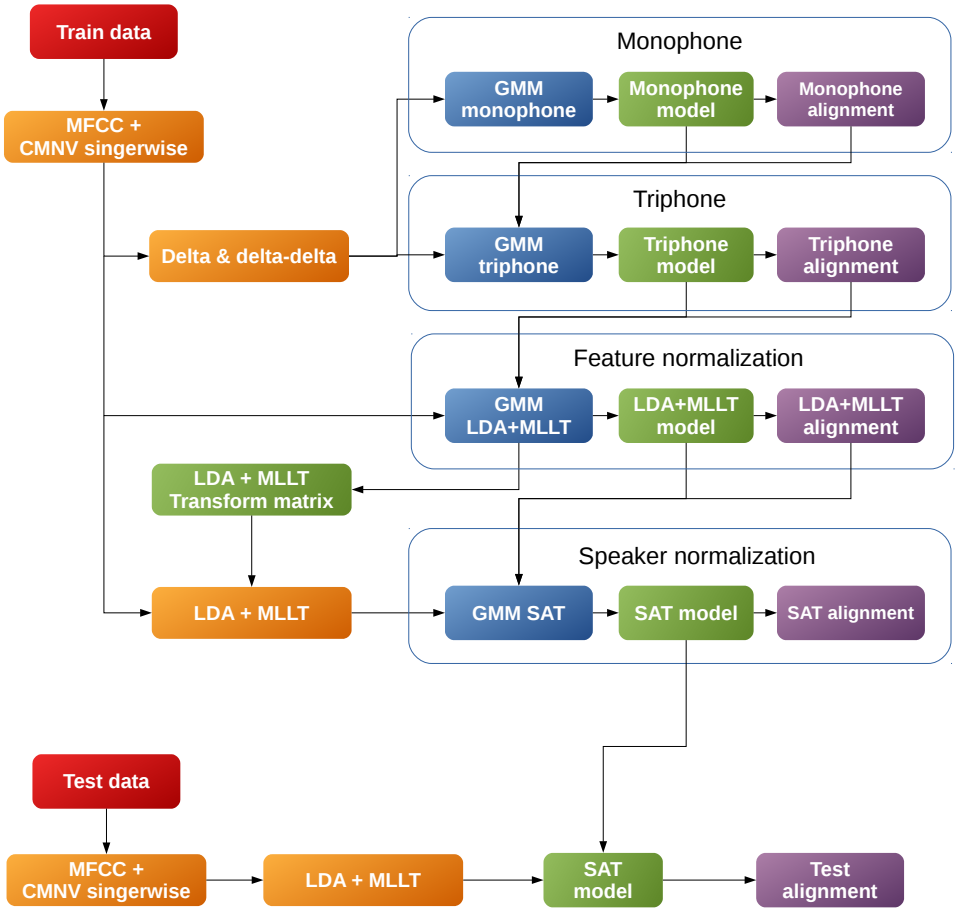


Figure 4.2.1: Multi-singer phoneme alignment process

The Kaldi phoneme alignment method is inspired in state-tying trees [165] and

uses top-down binary clustering instead of manually prepared questions to model data using a limited number of leaves and gaussians. The global structure of the phoneme alignment method can be seen in Figure 4.2.1.

With the phoneme sequence and acoustic features, monophone alignment is made using 40 iterations and 1000 gaussians for modeling. We split the iterations into 4 blocks of 10 iterations. In the first 10 iterations, realignment is made in each iteration. In the second block the realignment is made only in even iterations and in blocks 3 and 4 the realignment is made every 3 iterations. Using the obtained monophone models, tied state triphone models are created and the dataset is aligned again in 40 iterations. The use of triphones does not guarantee high improvement in alignment as seen in [111] and [22]. Then, the feature space is transformed using LDA followed by MLLT estimation. LDA compresses 9 frame context windows into 39 features and MLLT computes a diagonalizing transform over multiple alignment operations. This alignment phase uses fMLLR as a feature space transform to adapt acoustic features of each singer to the global model.

To align the audio files from the test set, the LDA+MLLT transform matrix is applied to the features and the singer adaptive model is used to align the phonemes.

The final phoneme segmentation has to be post-processed to define utterance segmentation. As we have already determined the beginning and ending phonemes of transcription utterances, the utterance segmentation marks are defined in the center of the silence between the last phoneme of each utterance and the first phoneme of its contiguous utterance. If there is no silence between these phonemes, the segmentation mark is set in the boundary between the phonemes.

4.2.1 Dataset

The proposed utterance segmentation system has been tested over a subset of the bertsolaritza singing voice recordings obtained in Section 4.1.2. Without the time marks of the utterances, multiple singer normalizations cannot be applied in the same recording. This is a problem in the recordings where the host sings and therefore more than one singer appear in the same recording. This is why we used only

the recordings with a single singer for this experiment. The manual labeling of all utterance boundaries is time consuming, therefore we aligned a subset of the utterances of the singer 0030b to use it as the test set. We also separated singer 0113b from the train database because we use it later in Section 4.3.1 as a test singer. The recordings of all the remaining singers are used as training data. The distribution of the train and test data for this utterance segmentation experiment is shown in Table 4.2.1. The duration provided for the data subsets corresponds to the duration of the voice segments found by the proposed VAD.

	Bertsolaris		**Utterances**		**Duration (min)**		**Phonemes**	
	Male	**Female**	**Male**	**Female**	**Male**	**Female**	**Male**	**Female**
Train	141	29	30672	4776	2248.55	365.18	552129	87909
Test	1	0	1239	0	86.25	0	22840	0

Table 4.2.1: Data used in the utterance segmentation experiment

4.2.2 Segmentation comparison and results

We have evaluated three types of models to make the phoneme alignments depending on the tokens used for the alignment. We used phoneme, syllable and word tokens. The tokens are the phoneme groups where transitions between phonemes cannot pass through silence states, forcing all the phonemes inside the token to be contiguous in time. Word token means that short pauses can be included only between words; syllable token means that short pauses can be included between syllables and words; and finally, phoneme token means that a short pause can be inserted between any pair of phonemes.

When syllable and phoneme tokens are used, we post-process alignments to remove intra-word short-pauses from the final alignment. In speech is very unusual to break words with short-pauses but in singing voice these rules are more flexible. Nevertheless, we observed that in Bertsolaritza singing style that it is a extremely uncommon practice.

For the evaluation of the utterance segmentation we separate utterance boundaries in two groups. The first group includes the utterance boundaries with a silence between the utterances in the reference. The second group comprises the utterance boundaries that in the reference have no silence in between. The evaluation of each group is made in a different way. When we have a silence between the utterances any position inside this silence can be accepted as the utterance boundary because no phonemes would be lost from any utterance. This is why we evaluate these boundaries as a binary label, the marks inside the silence are correct and the ones that are outside it are incorrect. For boundaries without silence, we evaluate these boundaries with the absolute time difference between the actual and the predicted mark. In the database there are 1268 utterance boundaries if we do not count the start and the end of the recordings. From these 1268 boundaries, 95 % contain a silence and the other 5 % have no silence. A graphic explanation of the evaluation is shown in Figure 4.2.2. It can be seen that the evaluation type depends on the presence of silence in the reference manual segmentation used to evaluate the system. The prediction of the silence in utterance boundaries is evaluated in Section 4.2.3. The results obtained in both evaluations are shown in Table 4.2.2.

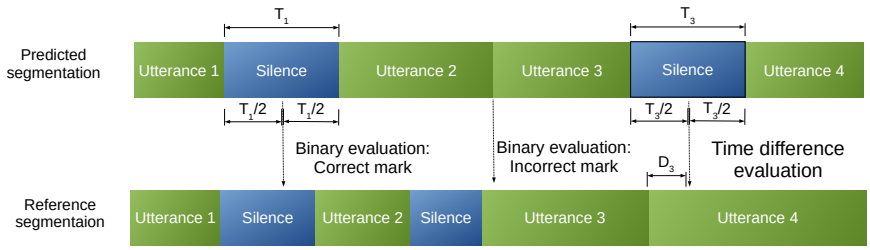


Figure 4.2.2: Utterance segmentation evaluation

We can see in the result table that the phoneme token obtains the best results in silence boundaries and the word token obtains the best results in the silenceless boundaries.

Token type	Silence boundaries accuracy (%)	Silenceless boundaries time difference (ms)
Phoneme	98.40	26.0 ± 19.15
Syllable	97.60	25.63 ± 21.03
Word	98.20	21.46 ± 17.79

Table 4.2.2: Utterance segmentation results

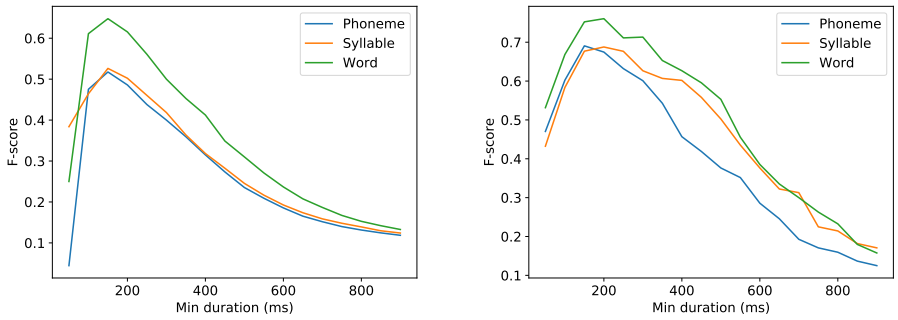
4.2.3 Silence detection

As utterance segmentation is a previous step to the phoneme segmentation, looking at the good results obtained, we decided to define the intra-utterance and inter-utterance silences with the alignments obtained in this step. We compared again the silence prediction score obtained using phoneme, syllable and word tokens. In the automatic phoneme segmentation, silences that did not appear at the initial transcription can be generated to optimize the phonetic models. These silences are created when the alignment with trained phonetic models cannot obtain an optimal phoneme sequence in the recording. They may appear because of two main reasons: either a silence that was not in the transcription exists in the recording or the pronounced phonemes differ from the average models of these phonemes obtained with the training data. These new silences contain important information that we can use in our database. There is no initial information of silences in any transcription of the database. We decided to use minimum length thresholds to decide if the silences generated in the phoneme segmentation have to be included in the transcription. We also compared different minimum lengths values to adjust this feature to the characteristics of our recordings.

To score the silence detection, we assign a binary label to each word. This label represents whether the word is followed by a silence. We exclude the initial and the last silence in each recording. Last words of utterances are classified separately from all the other words to score separately the intra-utterance and inter-utterance silences. We did this because the silences between utterances are very common while intra-utterance silences are scarce. Scoring separately these silences provides

a more accurate information of how well the method is performing.

As seen before in Section 4.2.2, the number of evaluated inter-utterance boundaries is 1268 and 95 % include a silence. In the case of intra-utterance silences, we have 4783 word boundaries and only 3 % include silence. We have used the F-score to evaluate the method because of the class unbalance. The score with different minimum duration thresholds can be seen in Figure 4.2.3. Table 4.2.3 and Table 4.2.4 present the final results.



(a) Detection of inter-utterance silences

(b) Detection of intra-utterance silences

Figure 4.2.3: Silence detection plots using different thresholds for the duration of the silence

Token type	Best score		Second best score	
	Duration (ms)	**F-score (%)**	**Duration**	**F-Score (%)**
Phoneme	150	51.75	200	48.58
Syllable	150	52.63	200	50.22
Word	150	64.77	200	61.53

Table 4.2.3: Results of inter-utterance silence detection

We can observe that the word token obtains the best results in both inter-utterance and intra-utterance silence detection. We think that the intra-utterance

Token type	Best score		Second best score	
	Duration (ms)	F-score (%)	Duration	F-Score (%)
Phoneme	150	69.06	200	67.46
Syllable	200	67.69	150	68.77
Word	200	76.06	150	75.23

Table 4.2.4: Results of intra-utterance silence detection

silence is more important than the inter-utterance silence due to the effect this has in the phoneme segmentation. The intra-utterance silences affect to the phonetic transcription of the utterances because of the coarticulations. Having silences in the edges of the utterances is a common feature in the speech or singing databases and therefore errors at the edges can be easier to fix in a future manual revision. We could use different minimum duration threshold to detect each type of silence, but we have preferred to define a single threshold for all silences to simplify the system. These are the reasons why we set the threshold in 200 ms.

4.2.4 Analysis of the utterance segmented database

In the previous section, we have analyzed different systems for utterance segmentation and the resulting best system uses word tokens with a minimum threshold for the silence detection of 200 ms. Using this system, we also segmented the recordings with foot based exercises where also the host sings. In the alignment of multi-singer recordings, we labeled utterances of the host as they would have been sung by the main bertsolari in the recording. Despite this wrong labeling we empirically observed that the model can generalize enough to segment utterances. The resulting database has similar properties of the data characterized in Section 4.1.2, but now it includes time marks for each utterance in the recordings.

4.3 Phoneme segmentation

Speech forced alignment is the process that aligns the phonetic transcription of the speech with the voice audio. State of the art algorithms use HMM topologies with different acoustic modelings to predict the alignments. Most common acoustic features are MFCCs and are modeled using GMMs or DNNs.

In the previous section we have exposed how we made the utterance segmentation of the database. To obtain the phoneme alignments we aligned each utterance as an independent segment with the respective singer label.

Phoneme alignments present a harder challenge than the utterance segmentation for multiple reasons. The main reason is that the precision needed for phoneme alignment is higher than in utterance segmentation given the silence margins that we have between utterances. Another motivation is that obtaining manually labeled phoneme alignments is more time consuming than manually setting utterance boundaries. The ideal option would be to manually create multiple singer reference phoneme alignments and evaluate a multi-singer aligner. However, manually creating a large volume of multiple-singer phoneme alignments is a highly time consuming work. The solution we propose to this problem is to manually create alignments for a pair of singers, and evaluate the difference of aligning them with a model trained using only data from the same singer and with a multi-singer model in which they have not been part of the training.

4.3.1 Single singer phoneme alignment

Two bertsolaris, a male (0030b) and a female (0113b) have been selected for the first phoneme segmentation experiments. Using the transcriptions and part of the recordings, phoneme alignments have been obtained using triphone-based forced alignment in HTK. Recordings equivalent to 1000 phoneme labels including the silences have been separated from the training material and reserved for testing purposes. The reference alignments have been manually created for these 1000 reserved labels as well as for another 1000 labels from the training set in order to

check the influence of including data used during the training to test the system. The amount of data used for the experiments is shown in Tables 4.3.1 (for bertsolari singer 0030b) and 4.3.2 (for bertsolari singer 0113b) .

	Utterances	**Phonemes**	**Duration (min)**
Train data with no labeled reference	1138	23565	92.46
Train data with labeled reference	44	980	3.42
Test data	49	982	3.77

Table 4.3.1: Dataset of bertsolari 0030b

	Utterances	**Phonemes**	**Duration (min)**
Train data with no labeled reference	1572	32982	137.70
Train data with labeled reference	41	908	3.63
Test data	49	954	3.50

Table 4.3.2: Dataset of bertsolari 0113b

We can observe in the tables that the female singer (0113b) has more available data for the training. The experiment will also allow us to compare the variability of the result with the amount of data available for training.

We have evaluated the quality of the alignments by calculating the percentage of the marks that are closer to the reference mark than a threshold. These percentages are shown in Table 4.3.3 and Table 4.3.4 for the male and female singer respectively.

Seen during training	**$<$5ms**	**$<$10ms**	**$<$20ms**	**$\geq$20ms**
Yes	23.12%	25.23%	34.37%	65.63%
No	23.43%	26.61%	38.96%	61.04%

Table 4.3.3: Percentages of marks within a certain distance from the reference for 0030b singer

Seen during training	<5ms	<10ms	<20ms	≥20ms
Yes	34.32%	36.62%	43.21%	56.79%
No	35.74%	36.72%	41.56%	58.44%

Table 4.3.4: Percentages of marks within a certain distance from the reference for 0113b singer

There are no differences in the results between using the recordings to train the phoneme segmentation system and aligning new recordings not seen during training. Therefore, the models created can be applied to segment future new recording of the same singer without quality loss. The results are better for the female singer probably because there was more speech material available to train the models.

For read speech, the state of the art in phoneme level segmentation is around 80% within 20 ms from the reference segmentation [56]. The results achieved in our experiment are still far from this value, but are good enough to make the process of manual segmentation easier.

4.3.2 Multi-singer phoneme alignment

For multi-singer phoneme alignment each utterance extracted in Section 4.2.4 is aligned with the same system used in that section. This time, as utterance time marks in the recordings are defined, the singer adaptation can be applied in a better way because different utterances in the same recording can have been sung by different singers. This means that the singer adaptation in fMLLR is applied in a more precise way. Differences on the phoneme alignment considering different types of token have been tested again: phoneme, syllable and word tokens have been taken into account. The dataset used to test the system is the same one used in Section 4.2.1 with the difference that we manually aligned the phonemes in the test recordings for the evaluation of this experiment. For the manual alignment, we used the system explained in Section 4.3.1 as initial help.

4.3.2.1 Results

The percentage of labels with a distance to the reference smaller than a certain value calculated using different tokens is shown in Table 4.3.5.

Token type	**<5ms**	**<10 ms**	**<20 ms**	**<25 ms**
Phoneme	10.63 %	21.40 %	40.45 %	48.81 %
Syllable	11.59 %	23.33 %	45.01 %	54.85 %
Word	10.46 %	20.64 %	39.13 %	46.53 %

Table 4.3.5: Phoneme segmentation results

In the table we can observe that the syllable token alignment is the one that obtains the best result. We have to compare these results with the results obtained in Section 4.3.1 (Tables 4.3.3 and 4.3.4). The results in the 5 and 10 ms range are better in single singer alignments, but in the 20 ms range there is no clear optimal system. In the results of the bertsolari 0030b in Table 4.3.3 we can observe that the results in 20 ms are better in all variations of the multi-singer system. If we compare it with the alignments of bertsolari 0113b in Table 4.3.4, only the syllable token system obtains better results than the single singer results. We have to take into account that the the agreement between human aligners is not of 100 % in small ranges. It has been proven in databases of different languages and sizes that human aligners need a range of 20 ms to obtain agreements higher than 90 % [65]. As the reference labels have been created using the single singer system as help, the bias towards that system is an important factor in small error ranges. We think that the 20 ms range is a good measure to compare these systems.

Considering the 20 ms range, the results show the advantages of aligning singing voice with syllable tokens. This can be because phonemes placed in note boundaries share properties. We also can observe that the multi-singer syllable model improves the "<20 ms" results obtained for single singer training in Tables 4.3.3 and 4.3.4. From this we can conclude that the resulting alignment when including all the material in the training set and using the syllable tokens is equal or better

than the alignments obtained from single singer models.

4.3.3 Alignment refinement with novelty features

Considering that HMM alignment results do not reach the level of the state of the art systems of phoneme segmentation in singing voice, in part because of the lack of previous musical information [58][84], we considered the application of a post-processing to get an improvement in the results. We propose to use audio novelty feature [46] for this phoneme segmentation refinement. This feature has been previously used for text free phoneme boundary detection [9]. The novelty uses the signal itself without an external model and searches for self-similarity between parameter frames.

The parametrization we used for self-similarity search is MFCC features because it is the standard in phoneme segmentation. Once we have the parametrization we calculate a 2 dimensional representation of the cosine of the angle between all pairs of parameter vectors using Equation 4.5.

$$S(i,j) = \frac{v_i \cdot v_j}{\|v_i\| \, \|v_j\|} \tag{4.5}$$

where v_i and v_j are parameter vectors in frames i and j. The resulting matrix S of a sample utterance of the Bertso database can be seen in Figure 4.3.1.

We can observe that the maximum value of the matrix is at the main diagonal because the diagonal represents the similarity of each vector to itself. Similar segments in the signal create bright squares with high similarity to near frames and dark areas correspond to different contexts.

We apply an analysis window to detect boundaries between self-similar areas in this similarity matrix. The analysis window or kernel is slid in from the diagonal of the matrix. This kernel increases value to its position if the areas in either side of the center of the window are self-similar. It will reduce the value if the cross-similarity is high. The base kernel that fits these condition is defined in Equation 4.6.

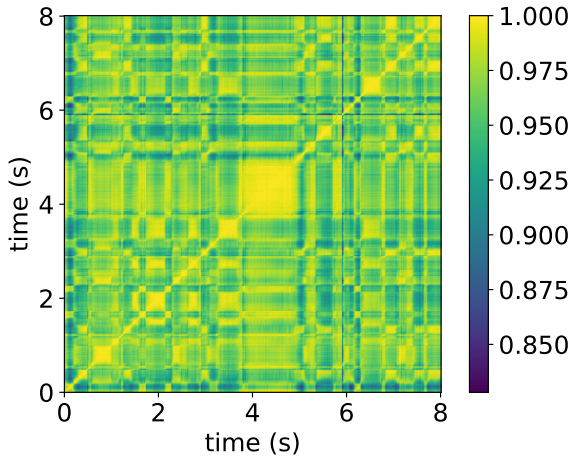


Figure 4.3.1: Similarity matrix of a sample utterance from Bertso database

$$C_B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{4.6}$$

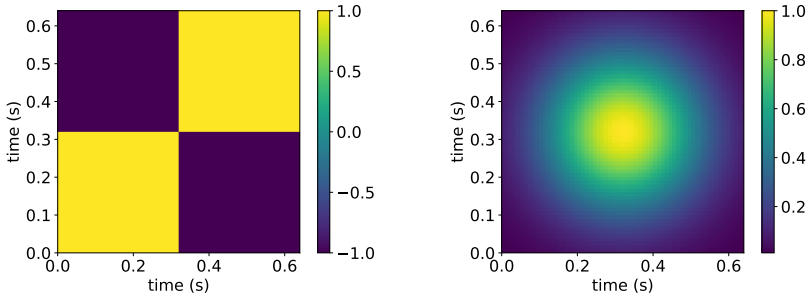
But the kernel has to be flexible to segment different levels of information, therefore we use the Kronecker product of the base kernel with a matrix of ones as shown in Equation 4.7

$$C_n = C_B \otimes J_n \tag{4.7}$$

where J_n is a *nxn* square matrix of ones defined in Expression 4.8.

$$J_n = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \tag{4.8}$$

A Kernel of $n = 64$, which can be considered of size of 0.64 seconds if we apply it to features calculated at 100 frames per second, can be seen in Figure 4.3.2a. To avoid edge effects, the kernel is multiplied by a 2D gaussian of the same size shown



(a) Chess Kernel **(b)** Gaussian Kernel

Figure 4.3.2: Kernel construction elements

in Figure 4.3.2b.

The final Kernel we have used is shown in Figure 4.3.3.

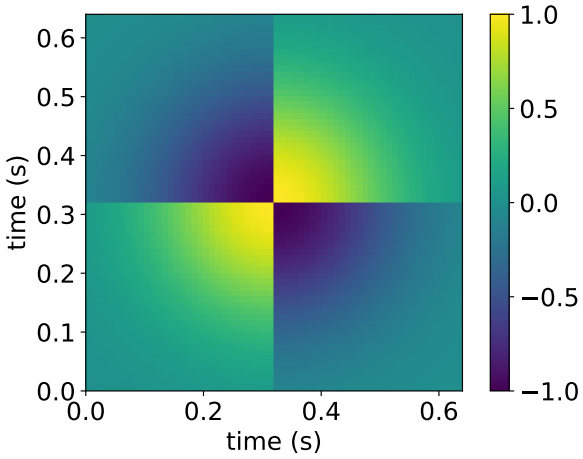


Figure 4.3.3: Chess Kernel multiplied with 2D Gaussian

The novelty feature is calculated applying Expression 4.9.

$$N(i) = \sum_{m=-L/2}^{L/2} \sum_{n=-L/2}^{L/2} C_n(m,n) S(i+m, i+n) \tag{4.9}$$

where L is the width of the kernel, C is the final kernel and S is the similarity

matrix. It can be seen that values in the similarity matrix further from the main diagonal than the kernel length are unused. Also the similarity matrix is symmetric, therefore both sides of the kernel create a repeated value that is summed to the result. As the novelty feature is a relative similarity value, removing this repeated values from the equation will not change the information of this feature. Taking into account these two factors, we propose to use limited similarity matrix and kernels. This optimizes the size and the computation of the similarity matrix and also the computation of the novelty feature. The final formula we propose to calculate the novelty feature is presented in Equation 4.10.

$$N(i) = \sum_{m=-L/2}^{L/2} \sum_{n=m}^{L/2} C_n(m,n) S(i+m, i+n) \tag{4.10}$$

The novelty feature represents the level of change that is happening in a parameter frame. In Figure 4.3.4 we can see that there is a strong correlation between high values of the novelty feature and phoneme onsets. Although all relative maximum values in the novelty curve do not correspond to a phoneme onset, all phoneme onsets are located in a local maximum or very close to a local maximum value in the curve of novelty. We can observe too that the last phoneme boundary is different and it is not associated with a clear relative maximum in the novelty curve. We observed that this absence of relative maximum is very common on the boundaries between silence and phonemes. The transitions from voice to silence tend to be slower and therefore there is no clear novelty in any area.

Considering this correlation between phoneme boundaries and novelty local maximum values, we decided to refine the phoneme boundaries located by the method described in section 4.3.2, searching maximums near the predicted boundaries. Considering that in the boundaries between silence and phonemes there is no clear maximum we decided not to refine these boundaries. The shift area for each boundary extends from halfway to the previous boundary to halfway to the next boundary. To exclude the noise of maximum values that are not related to phoneme boundaries we define a minimum value to consider a maximum as a po-

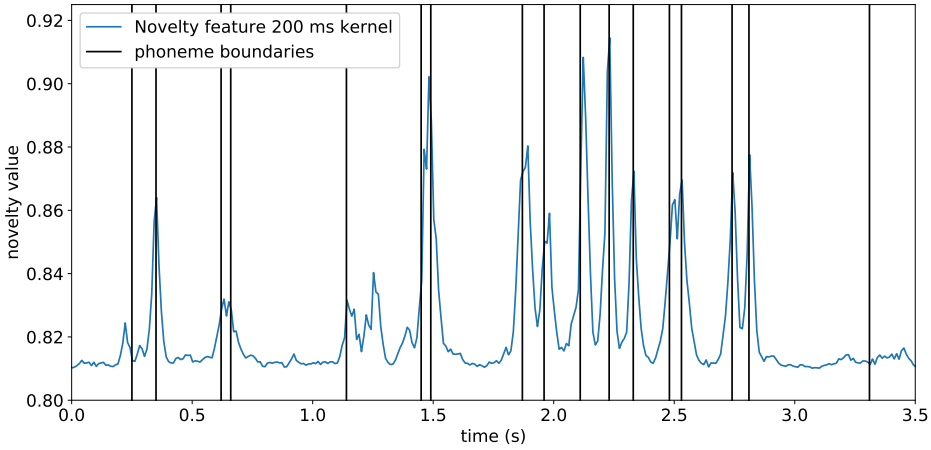


Figure 4.3.4: Novelty feature and phoneme onsets

tential phoneme boundary. To decide if a phoneme has to be shifted, we search for the potential boundaries with the highest novelty value in the shift area. If there are no potential shifts in this area, the boundary is not moved. This algorithm is defined in Algorithm 1.

We tested the improvement of the alignment using different values for the configuration parameters. We defined two trainable parameters for the configuration of the refinement process based on the novelty feature: kernel size and minimum value for potential shifts. The kernel size defines the analysis window used in each frame to define the novelty value and is related to the length of the segments we want to detect. The minimum value is related with the level of noise in novelty signals and undesired maximums.

We chose the optimal value for the kernel size and minimum value of potential shifts selecting the values with the best result in a subset of the data available for the test singer presented in Section 4.2.1. We evaluated the optimal values for the algorithm in the rest of this test singer data. The information of the data used for this experiment of segmentation refinement is described in Table 4.3.6. The differences in the figures presented now compared with the ones given in Section 4.2.1

Algorithm 1: Segmentation shift.

Data: Segmentation marks and novelty feature
Result: Refined segmentation

begin
  $M \leftarrow$ List of frame position of time marks $M = \{S_0, S_1, ..., S_K\}$;
  $N \leftarrow$ Novelty feature;
  $d \leftarrow$ Minimum value for potential shift;
  $M' \leftarrow$ Empty refined segment list;
  for $i = 1$ ***to*** $K - 1$ **do**
    $T_a = S_i - (S_i - S_{i-1})/2$;
    $T_b = S_i + (S_{i+1} - S_i)/2$;
    $n = \max_j(N(j)) \quad where \quad T_a < j < T_b$;
    if *$(n > d)$ and (S_i is not a boundary of a silence)* **then**
      $j = \arg\max_j N(j) \quad where \quad T_a < j < T_b$;
      add j to M'
    end
    else
      add S_i to M'
    end
  end
end

are due to the fact that here we calculated the actual phoneme durations, without taking silence into account. We randomly selected 100 utterances from the dataset and tested the percentage of labels within 25 ms from the reference label with different values for the configuration parameters.

	Singers	**Utterances**	**Total length (min)**	**Total phonemes**
Train	1	100	6.53	1924
Test	1	1090	76.63	20166

Table 4.3.6: Data for the segmentation refinement experiment

The results of the optimum configuration parameter search grid are shown in Figure 4.3.5. The image shows the segmentation mark percentage with an error

below 25 ms obtained with each parameter configuration.

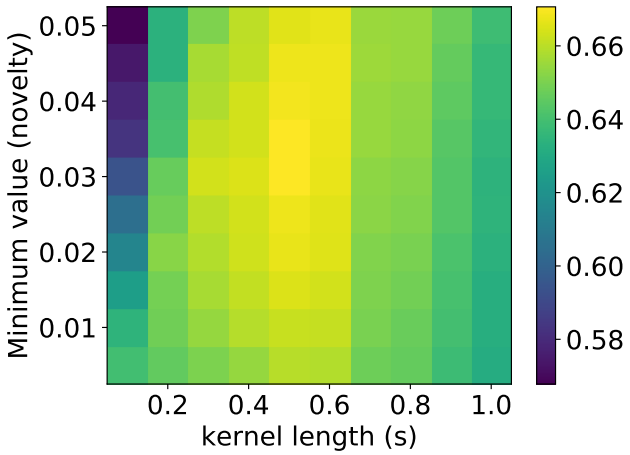


Figure 4.3.5: Segmentation score with a 25 ms margin and different refinement parameters

The best configuration parameters for our proposed method in this data-set are a kernel length of 0.5 seconds and a minimum novelty threshold for maximum values of 0.03. Using these parameters we calculated the new alignment results.

After defining the optimal values for the parameter we compared the results of the phoneme alignment refinement in the remaining 90 % of the manually aligned data. The alignment results before and after the refinement are shown in Table 4.3.7.

Segmentation	**<5ms**	**<10 ms**	**<20 ms**	**<25 ms**
Base	11.68 %	23.48 %	45.34 %	54.97 %
Refined	19.63 %	35.42 %	59.09 %	67.01 %

Table 4.3.7: Results of refined phoneme segmentation

The results show that we improved the alignment score with the novelty refinement method by 12 points. We have analyzed the segmentation improvement by

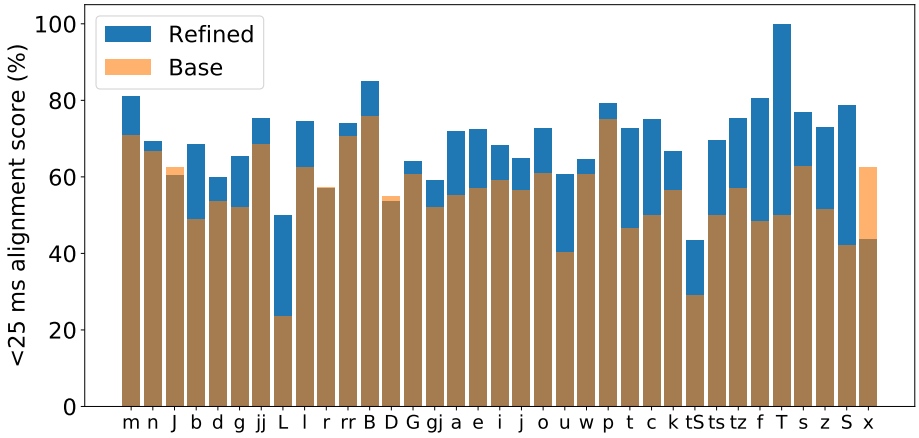


Figure 4.3.6: Analysis of the effect of segmentation refinement by phoneme

phoneme to ensure that the improvement is general and we are not getting a distortion in the mean scores with an improvement of certain phonemes and the degradation of others. The comparison is shown in Figure 4.3.6. Most of the phonemes improve the alignment result. We have a small degradation of the result in phonemes /*J*/, /*D*/ and /*x*/. A possible explanation of the degradation of the /*J*/ and /*D*/ phonemes is that these phonemes are the shortest ones, and therefore the selected kernel size has not enough temporal precision to detect the boundaries of these phonemes. In the case of the /*x*/, it is also a short duration phoneme, but we think that the main reason for the big difference with the original agreement is that this phoneme has very few appearances in the database. With few appearances the result of its alignment is highly volatile and not statistically significant.

4.3.4 Analysis of the phoneme segmented database

Considering the results obtained in this Section, we used the multi-singer alignment and the novelty refinement in all the database defined in Section 4.2.1. In the process 263 utterances have not converged to obtain an alignment. Considering that the total number of utterances is 41597, these utterances represent the

0.006 % of the total utterance. We observed that this happens because of errors in the transcriptions or high levels of noise in the recordings.

After the alignment, we split the multi-utterance audio files into utterances using the boundaries of the edge phonemes in each utterance. We added silence segments of 200 ms to all the new utterance files at the beginning and the end of the audio file because it is important for some speech modeling processes. We saved also the alignment of the HMM states of the phonemes in parallel with the phoneme alignment. The resulting database has similar properties to the one defined in Section 4.2.1 without the 263 utterances that could not be aligned.

4.4 Musical labeling

In the previous section we have described the process designed to obtain phoneme alignments without using musical information. This was necessary to define the linguistic part of the labels. In this section we will present the method proposed to create the corresponding musical information that will be combined with the phoneme alignments to obtain musical scores.

In Section 4.4.1 we analyze the selection of f_0 extraction algorithm considering that this parameter is fundamental in the representation of note pitch of the music score and general expressiveness of the singing voice. Then, in Section 4.4.2 we define a method to annotate a recording using the music scores of bertso melodies. Considering the improvisation nature of bertsolaritza and the variations of melodies that happen in live sessions we proposed a method to evaluate the similarity of the bertso musci scores and the recordings. We have also created a musical labeling method to label the recordings without music score information that is presented in Section 4.4.3. In Section 4.4.4 we explain the vibrato labeling and parameter extraction that we created to evaluate the use of vibrato in the Bertso database. Finally, in Section 4.4.5 we explore different methods to obtain the bertso melody from the recordings knowing that multiple variations occur in the improvisation.

4.4.1 f_0 calculation

For the musical labeling many different analysis over the musical pitch values and sung pitch values must be made; this is why the calculation of the fundamental frequency is a key part in the analysis of the recordings. We used a comparative study of algorithms for singing voices to decide what fundamental frequency extraction method to use [11]. In the study four algorithms are compared and each of them gets the best results in one of the four different error measures applied to f_0 detection. The analyzed errors are the next ones:

- **Gross Pitch Error (GPE)**: Errors higher than a semitone in f_0 calculation.
- **Fine Pitch Error (FPE)**: Standard deviation of error in f_0 values that have no GPE in f_0 calculation.
- **Voicing Decision Error (VDE)**: The proportion of frames in which the Voiced/Unvoiced decision has been incorrect.
- f_0 **Frame Error (FFE)**: Overall performance measure taking into account frames with the GPE and VDE errors.

The autocorrelation method obtained the second best overall score and the best voicing decision score. This is why we decided to calculate f_0 in our databases applying the autocorrelation method. The range of possible f_0 values is also an important configuration parameter that can reduce the errors in the f_0 calculation. This is why we empirically analyzed the range of the fundamental frequency in the whole Bertso database, searching for the upper and lower limits. Our conclusion is that the range of f_0 values of this database goes from 75 to 580 Hz. This frequency range in notes goes from $D_2^{\#}/E_2^{b}$ to $C_5^{\#}/D_5^{b}$ in English notation or 39 to 73 in MIDI numbers.

4.4.2 Musical labeling using music scores

The Bertso database provides the melody label in 33.32 % of the recordings, meaning that the melody the bertsolari used to create the improvised bertso is recognizable. The melody is selected by the singer that has to choose among all the melodies suitable for the meter set by the host. Each melody having its own meter, it cannot be used with any lyrics; the meter of the improvised bertso has to adjust to the meter of the melody to sound correct. Ideally, a recording with a labeled melody would have the same number of utterances in each bertso as the music score and the same number of syllables in each utterance of each bertso as notes are in the music score. If this ideal condition would be filled the assignation of the notes would not be a problem, because the pairing would be straight forward. To test this first affirmation we compared the recording transcriptions with their respective melodies. From the number of recordings with the same amount of utterances per bertso, the utterances with the same number of syllables as notes in the score are 21.44 % of the total number of utterances. This low percentage shows that if we want to use the music scores of the standard melodies, more work has to be done.

New melody transcriptions and adaptations should be done by music experts. The standard melodies are traditional Basque melodies or melodies that bertsolaris created and sung in a improvisation session. The variants of each melody are a complicated issue, because new utterances or encores improvised by bertsolaris need to be identified and transcribed. For these reasons, we have focused our effort on utterances that match the exact number of syllables of the corresponding standard melody.

The theoretical interpretation of the notes of a music score is a staggered curve representing the duration values and pitch values derived from the musical score. This direct interpretation lacks naturalness in singing voice and the staggered curve suffers distortion in the actual interpretation. This distortion affects the duration of notes, the square form of the transitions between pitch values and the constant value of the pitch inside each note. Additionally, each singer has his or her own

vocal range, and needs to shift the melodies to her his own range. We named this shift as singer dependent conversion. Figure 4.4.1 shows how singer dependent conversion shifts the theoretical melody range to their own vocal range.

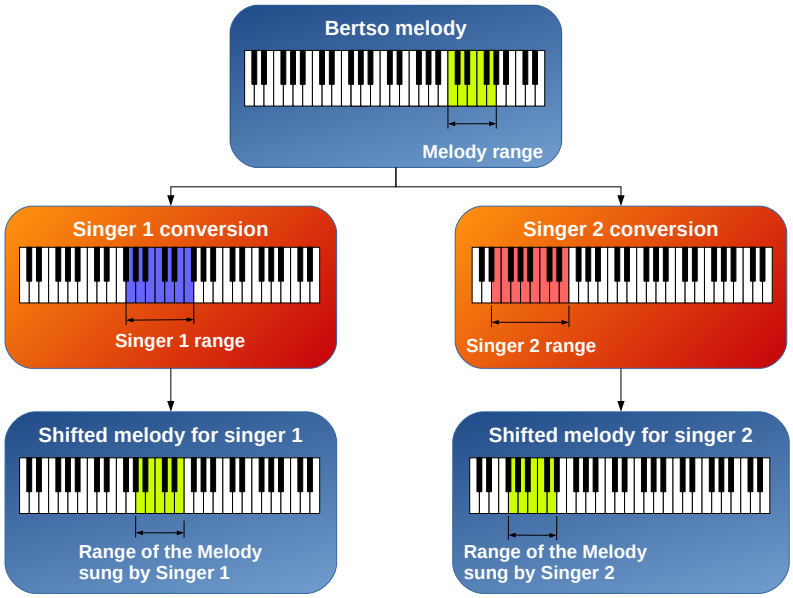


Figure 4.4.1: Singer dependent melody conversion

To compare the notes in the music score and the real interpretation, we can compare the f_0 of the actual singing voice and theoretical note assigned to each phoneme. This means that we have to create a frame level representation of the notes in the music scores. From the music score we know the note pitch value of each phoneme and from the phoneme alignments we can determine the frames that correspond to each phoneme. Combining these two elements, we can create a frame level representation of the musical score that is parallel to the real singing signal. We named this representation of the notes in the music score as ”full pitch projection”. We call it ”full” because this representation assigns pitch values to the unvoiced frames too. In the case of assigning pitch values only to voiced frames we named it ”pitch projection”. An example of singer dependent conversion can

be seen in Figure 4.4.2, where we can observe the full pitch projection of a music score and the f_0 of the singing voice. In the figure, we used the new term "phonetic boundaries" of the notes. We call phonetic boundaries of a note to the boundaries of the segment defined from the beginning of the first phoneme of the note to the end of the last phoneme in the same syllable of the lyrics. The segments without pitch signal are silences. In the figure we can observe all the phenomena that happens in a music score interpretation with singer dependent conversion.

- The boundaries between notes are not a step function.
- The value of f_0 inside a note is not a constant.
- The duration of each note cannot be exactly determined.
- The singer has adapted the range of the melody to his or her own vocal range.

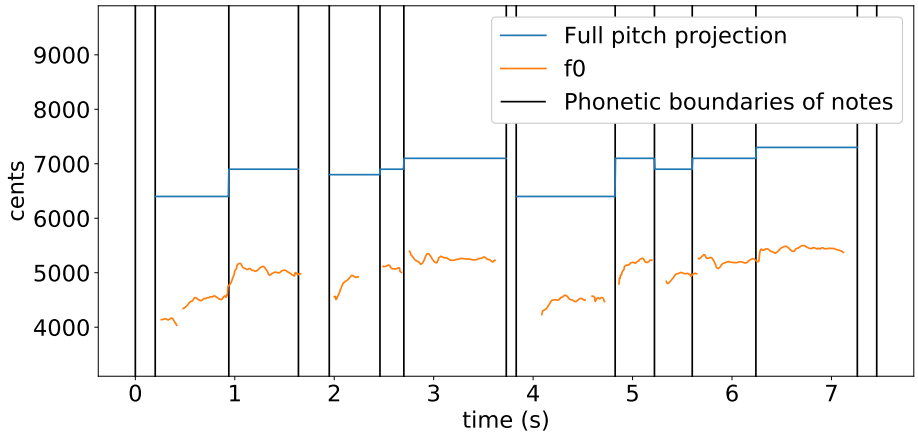


Figure 4.4.2: Comparison of the pitch projection and the f_0 of the real interpretation with singer dependent conversion

With these observations about the interpretation of theoretical music scores, we decided to use the melodies to label the note of each voiced frame of the f_0. This labeling is made by aligning the pitch projection with the f_0. We defined the

music score alignment as a problem of alignment in two axes: time axis and frequency axis. In the time axis, the locations of note transitions change considering phoneme boundaries and the transitions and stable areas of notes suffer modifications compared to a staggering sequence. In the frequency axis, the range of the melodies shift to the vocal range of the singer. To solve this problem we propose a novel algorithm for score alignment (Algorithm 2). The algorithm removes all the unvoiced frames from the f_0 and pitch projection, saving the position of the voiced frames in the original signal. After that, the voiced frames are aligned applying two iterations of mean conversion and DTW. We decided to apply only two iterations observing the value of the mean shift in each iteration. After the second iteration this value starts to be very small. With no mean shift the signals that are compared with DTW are very similar and therefore the same result would be obtained in every ulterior iteration. After the alignment, the new values in the voiced pitch projection are relocated to the original positions of the voiced frames. We named the new aligned values of the pitch projection as "aligned pitch projection".

Algorithm 2: Score alignment.

Data: f_0 and full pitch projection
Result: Alignment in time of the notes in the score with the f_0 curve

begin
- $f_0 \leftarrow$ Fundamental frequency of the singing voice;
- $PP \leftarrow$ Full pitch projection;
- $V \leftarrow$ Voiced values of f_0;
- $VPP \leftarrow$ PP values in voiced frames;
- $VF \leftarrow$ Save positions of all voiced frames;
- $VPP = VPP - mean(VPP) + mean(V)$;
- $VPP = DTW(VPP, V)$;
- $VPP = VPP - mean(VPP) + mean(V)$;
- $VPP = DTW(VPP, V)$;
- $APP \leftarrow$ Use VF to relocate the aligned VPP;

end

The visualization of the process can be seen in Figure 4.4.3. In the top graphic

the full pitch projection (orange) and the actual f_0 curve (blue) are represented. In the middle graphic, the unvoiced segments are removed from the staggered and f_0 curves and the aligned pitch projection for the voiced frames (green) is calculated applying Algorithm 2. In the bottom plot the final stage of the algorithm is applied and unvoiced frames are reinserted into place. We can see that in this example the note labeling of each frame (orange) is adjusted to the sung notes (blue). There is only a mismatch from second 0 to 0.2, but it is due to the microprosody generated by a consonant in the note that goes from 0 to 1 seconds.

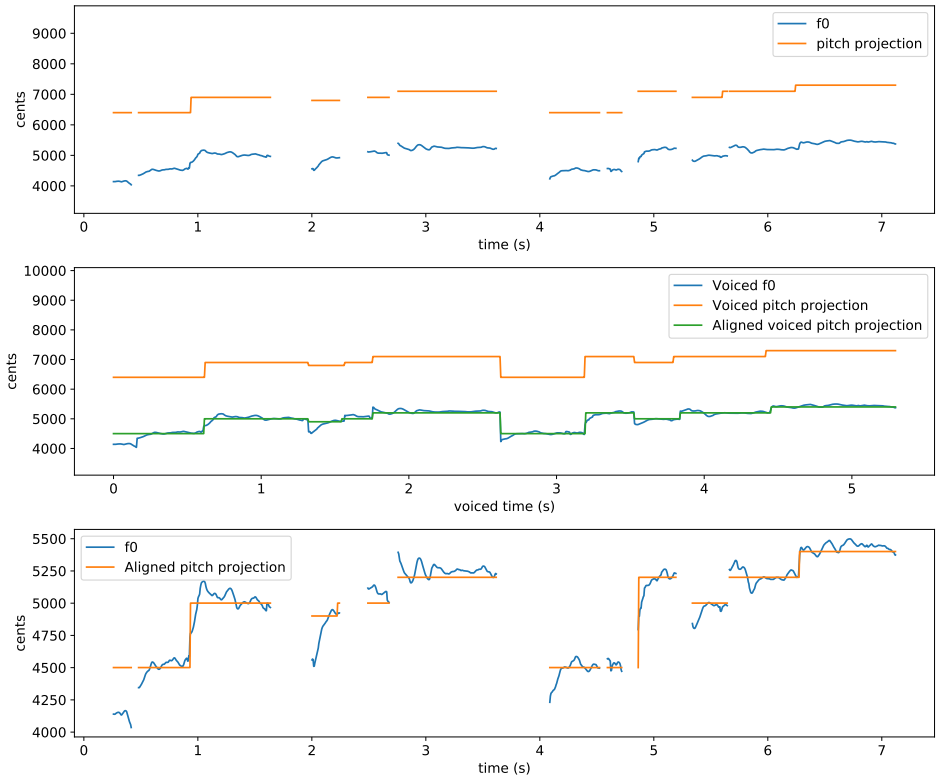


Figure 4.4.3: Representation of score alignment process

This method obtains good note labeling results when the interpretations are close to the music score. However, we know that the melodies are modified in improvisation sessions. Three scenarios can happen in the alignment depending on

the modifications of the melodies: correct labeling, out of tune labeling and wrong alignment. Figure 4.4.4 shows an example of a good labeling. We can observe that the initial form of the pitch projection and the sung f_0 have the same structure. Figure 4.4.5 presents an out of tune alignment. The structure of the melody and sung f_0 are similar and therefore there is no modification of the position of the notes after the alignment. But from second 0 to 0.5 the cent level of the f_0 and the aligned pitch projection are not the same. This happened either because the bertsolari has lost the tuning in that note or because a variation of the melody has been sung. In Figure 4.4.6 we can observe what happens when the sung melody and the score are completely different. The note pitch values of the pitch projection have been displaced because the alignment is impossible. This happens when the singer sings a completely different variation of the music score.

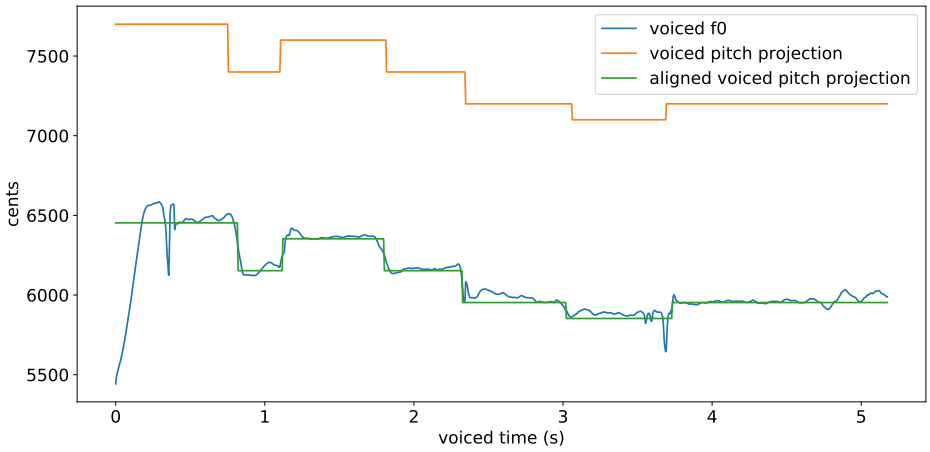


Figure 4.4.4: Correct labeling of notes using the proposed score alignment algorithm

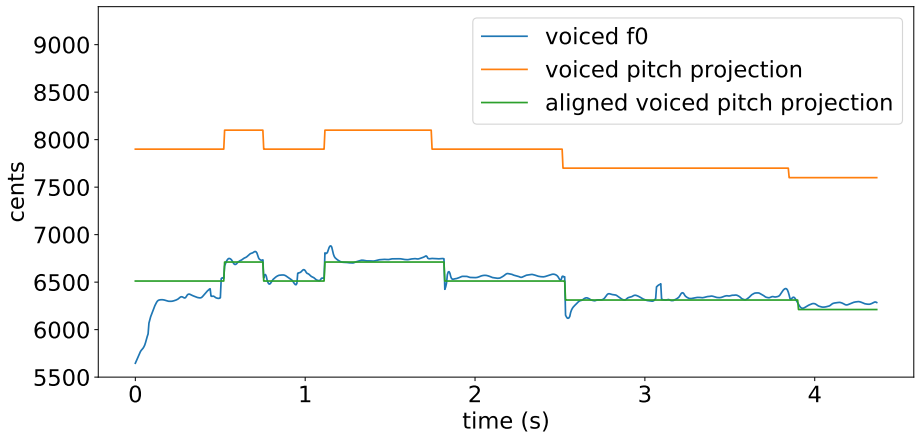


Figure 4.4.5: Alignment with correct note onset detection and note deviations

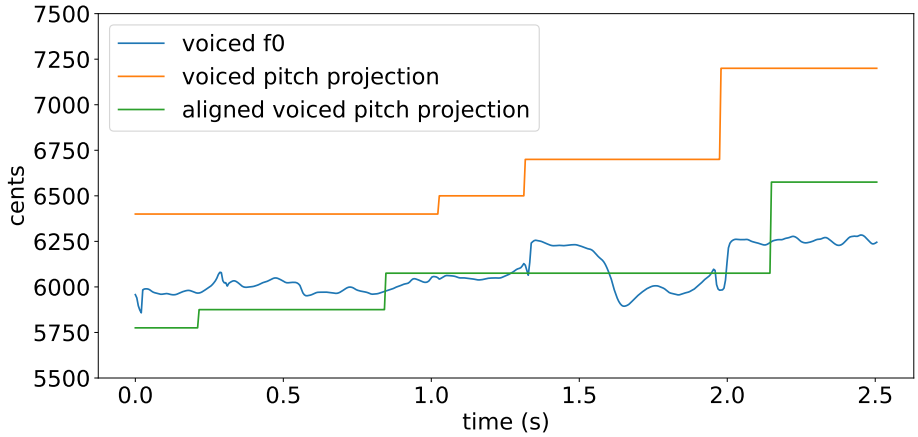


Figure 4.4.6: Alignment of different melodies

We observed in the previous figures the three main scenarios that are produced in the note alignment process. Even if the process produces an alignment as output, the quality of the alignment cannot be guaranteed. As we do not know beforehand what recording has strong deviations from the music score, we propose a method to evaluate these alignments a posteriori. The out of tune scenario is acceptable

because the positions of the notes are correct and the pitch label could be corrected with postprocessing. The melody difference is the scenario that we want to detect and avoid, because the resulting labeling is not easy to fix.

We propose a method to evaluate the similarity of the sung melody and the music score that compares the position of the pitch projection frames before and after the alignment. We want to measure if the pitch projection values maintained the original note position after the alignment. With this purpose, we create a new index projection of the notes, similar to the pitch projection. In the index projection the value given to each frame is a number corresponding to the position of the note in the utterance. If the same note is repeated in the music score, the index of the first note in the sequence is given to all the consecutive frames corresponding to the syllables assigned to these notes in the lyrics. We apply the DTW alignment procedure proposed in Algorithm 2 to the index projection, but without applying the mean shift. After the alignment, majority vote of aligned index projection is applied inside phonetic boundaries to obtain the new index value in each note. The majority vote assigns an aligned index value to each note.

The result of the method can be seen in Figure 4.4.7. In the upper part of the figure, using the melody alignment proposed in Algorithm 2 we obtain the aligned pitch projection. In this case the aligned pitch projection and f_0 are well adjusted, but the melody has not been properly sung, as the fourth note had to have the same value as the previous ones according to the music score. To compensate for this early change of note, the bertsolari has repeated the sixth note although it was not repeated in the music score. In the lower plot of Figure 4.4.7 we have visualized the result of the index projection alignment. The first three notes in the aligned index projection are assigned the same index (1) because they correspond to the same tone (6100 cents). The same happens with the four first notes in the index projection obtained from the music score. We can also see in the Figure that index 5 appears one note before in the aligned index projection than in the index projection calculated from the music score. Similarly, indexes 6 and 7 of the aligned index projection also occur one note before. This means that the melody and its interpretation used different pitch values in the same note positions. This misalign-

ment should not happen in an art where respecting the meter is very important. When a bertsolari creates a bertso with a specific meter he or she uses predefined melodies of that meter to define the pitch of each syllable in a simple way. With our proposed alignment procedure, we can measure if the pitch values in the melodies and the sung melodies coincide.

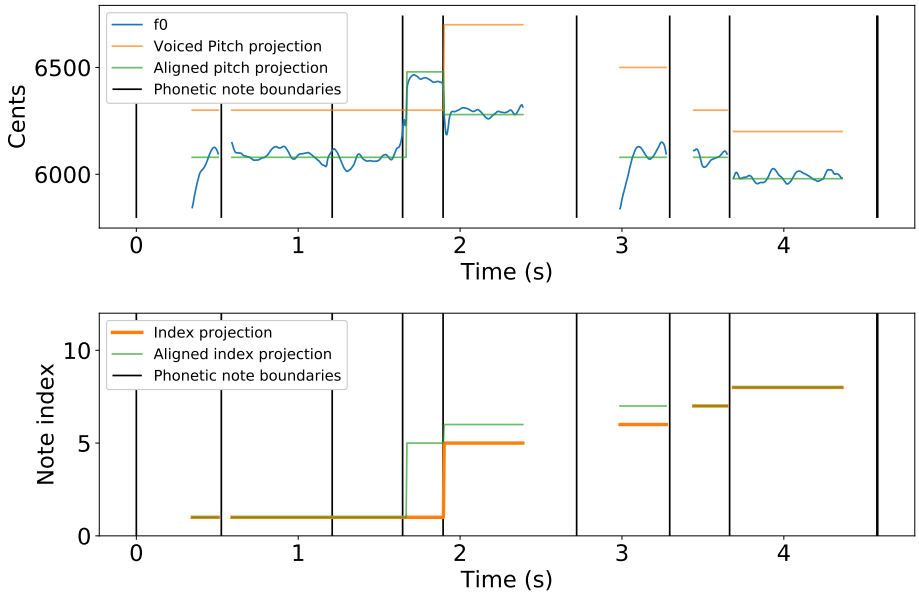


Figure 4.4.7: Alignment of the note index

Our proposal is to compare the theoretical indexes and the aligned indexes to evaluate if the sung f_0 adjusts to the structure of the music score and in which notes the differences have taken place. This measure can serve as a score of similarity between the music score and the improvised singing recording, i.e., the results obtained with the alignments provide information about the similarity of the score and the singing voice.

We have defined two evaluation measures for the melody similarity: percentage of notes with the same index before and after the alignment (Correct notes) and percentage of singing utterances where all the notes have the same index before

and after the alignment (Correct utterances).

To compare the results of our proposed algorithm in our database with results in other singing voice databases, we used the NITech and NUS databases, described in Sections 3.1.2 and 3.1.3 respectively. In the NITech database we used the note boundaries instead of the phonetic boundaries because phoneme alignments are not provided. We can observe the note alignment results in Table 4.4.1.

Database	Correct utterances %	Correct notes %
NUS database	68.66	91.32
NITech database	93.11	98.69
Bertso database	30.95	77.14

Table 4.4.1: Note alignment accuracy

The results in the NITech database, with a professional singer and manual labels are very good: the alignment adjusts well to the labeled score. We have to take into account that the labels of the NITech database are note based and not phoneme based, this makes the majority vote inside the evaluated region optimal for the alignment. The NUS database, with amateur and choir singers and external theoretical musical scores gets lower scores. Finally, the Bertso database has the lowest scores. This does not mean necessarily that bertso singers are bad singers, but that theoretical melodies and the real singing do not coincide. Taking into account the improvisation nature of bertsolaritza, two scenarios can be problematic in our database to consider the proposed similarity measure. The first one is the one observed in Figure 4.4.5: the alignment can be correct because the melody structure is similar but the pitch values are not the same. The second problem is the one observed in 4.4.7: shifting one note affects to the alignment of many subsequent pitch values even when the pitch values have been correctly sung. We are aware of these issues, but even so, this evaluation can provide information about the distortion of the melody interpretations and the potential use of these melodies to create the pitch labeling of the database.

To analyze the accuracy of the interpreters with respect to the duration of the

notes, we have to compare the length difference between the ideal and real notes in the recordings. As the music scores of Bertso database do not provide tempo information, we have a missing variable to calculate the ideal duration of the scores. Given that music durations are a referential system and there is no standard tempo to interpret a music score, we decided to assign the tempo that minimizes the error between ideal and real durations to every score. We used Equation 4.11

$$l = \frac{1}{n} \sum_{i=1}^{n} \frac{|d_i - \hat{d}_i|}{d_i} \tag{4.11}$$

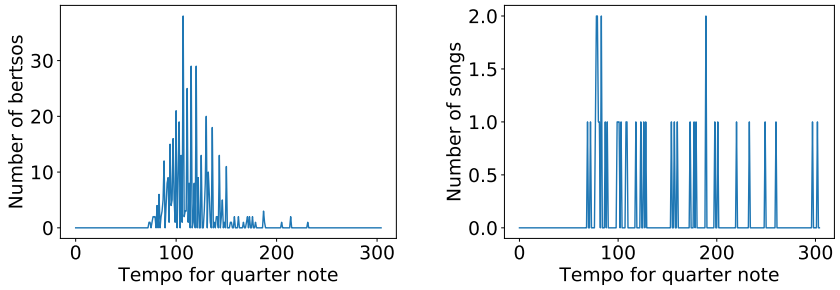
where l is the error, n is the number of notes in the recording, d_i is the duration of the voiced phonemes in the *ith* note and $\hat{d}_i$ is the ideal duration of *ith* note calculated with Equation 2.4. Optimizing the tempo to minimize this equation in each recording gives us the optimal tempo of the song and also the duration distortion in reference to this optimal tempo. The distortion results for the Bertso and NUS databases are shown in Table 4.4.2. We did not calculate the duration distortion in the NITech database due to the lack of phonetic alignment. The note alignments included in this database cannot be used to evaluate the difference between the actual duration of the syllable corresponding to the note in the lyrics and the duration of the note as indicated by the music score.

Database	**Distortion per recording (%)**
NUS	31.99 ± 11.13
Bertso	30.38 ± 7.17

Table 4.4.2: Time distortion in note durations

As the time distortion calculation requires the optimization of the tempo with a given symbol pairing, we first obtained the tempo distributions in both databases. Although the distortion in Table 4.4.2 shows us that these tempos combined with note symbols do not define perfectly the real durations, these optimal tempos can give us an idea of where the tempos of these databases are located. The distributions are shown in Figure 4.4.8. We can see that the tempos used in Bertso database

are concentrated between 75 and 180 while the NUS database exhibits a wider representation of different tempos.



(a) Bertso database **(b)** NUS database

Figure 4.4.8: Optimal tempo distribution

As the distortion results in Table 4.4.2 show, NUS database and Bertso database have similar duration coherence in reference to the musical scores. In [123], Saino et al. observed that the phoneme durations and positions are not coherent with the exact projection of the music symbols in time. Forcing the durations and positions of the phonemes to the duration of musical symbols resulted in unnatural results. They fixed these misalignments modeling the delays of the phoneme boundaries with Gaussians and decision trees. Measuring this distortion, if we use the NUS database as the unique reference, we can consider that the duration distortion of the Bertso database is better by a small margin. Although this distortion is relatively small, the "natural" distortion and not having a real gold standard create too many unknown variables that cannot justify the use of the durations in the scores and tempo alignments to calculate the delays defined in [123]. In addition, the strict use of the tempo is not a primary problem in bertsolaritza. Given that there are no instruments in the background there is no need to synchronize with them. Also the improvisation factor can create distortion of the tempo during the recording.

Given the results of music score coherence analysis in the Bertso database, we consider that it is complicated to use the music scores to obtain the music labels

of the recordings in the Bertso database. This is why we have to devise a labeling procedure without taking into account the music scores.

4.4.3 Musical labeling without musical score background

We have seen in Section 4.4.2 that the music scores of the melodies from our database cannot be used as reference to create the labels of the recordings. Without the music scores, there is no initial note structure to align and to define note boundaries; in addition, there is no note symbol reference to detect the tempo of the recording. The only information known when we have to create the music labeling on the database is the phoneme alignment, and therefore the initial information available to obtain the musical labeling without using the music score is the phonetic boundary of each note. With phonetic boundaries, the pitch of each note can be labeled analyzing the f_0. Nevertheless, the symbols, tempo, key and measures are challenging to label. Music knowledge is needed to label measures and key of singing voice recording, and even having this knowledge, the manual labeling of all the database would be highly time consuming. Our proposed automatic labeling procedure is to create music score structures but without key and measures.

The key information is needed in the reading of the music score for the interpretation of the positional information of the notes, but if we label the notes in cent level directly using the fundamental frequency of the voice we can simplify this step. Without the measure, our labels may miss the music structure information. Theoretically, the first note of a measure defines the beat and therefore has more power than the rest of the notes inside the measure. Loosing this information can create a more plain synthesis, but as we have no original scores and we are not confident about these prosody structures in the bertso improvisation style, we decided to avoid using it. In this way, we have defined the labeling of the pitch and duration of the notes like independent problems: the labeling of these two types of information are not related and we used independent methods to define each of

them.

The labeling of the note pitch without musical score can be defined as the process of assigning a musical note pitch value to the section of the recording inside a note phonetic boundary. Knowing that f_0 is closely related to the musical pitch values, this appears to be a simple problem. Measuring the closest note pitch to the f_0 values inside the phonetic boundaries of the note could be an option to label it. Saino et al. showed in [123] that note boundaries and phonetic boundaries in singing voice are not aligned. We consider that the phonemes in each syllable of lyrics constitute a note, but this does not mean that the calculation of the pitch of these notes can be done using f_0 inside these phoneme boundaries. In addition, there are many musical phenomena that create distortion in the stability of the notes, for example vibrato and portamento. Therefore we consider that it is important to create a note detection method that does not depend on the phoneme alignments.

4.4.3.1 Note detection algorithm

Our algorithm to detect note uses the premise that music is a sequence of pitch values that need a minimum stability and duration to be noticed as such.Under this premise, our main goal will be to find stable areas of f_0 where a note can be found. The devised method has two well defined steps:

- Finding stable areas in the f_0 curve
- Calculating the note value inside the area

Compared with a state-of-the-art algorithm like Tony [93], our method is much simpler, but, Tony has parameters that have been tuned with observations of manually annotated data. As we do not have any manually annotated data, we needed to devise a method that required the minimum supervision.

The first step in our proposed algorithm is to map the pitch curve to cent scale [12, 42], taking as origin the lowest note that we consider may appear in the recordings.

We use expression (4.12) to do the mapping:

$$f_{0c} = 1200 log_2(\frac{f_o}{f_{ref}}) + 6900, \tag{4.12}$$

where f_{ref} is 440 Hz, the frequency of A_4 note.

Once we have the curve in cent scale, a smoothing is applied to the pitch curve to neutralize vibrato variation within the notes. For this smoothing we used the local maximum and minimum points. Interpolating all maximum points a curve of maximums is created and the same process is applied to the minimum values. The smoothed curve is created averaging maxima and minima curves. The process is visualized in Figure 4.4.9

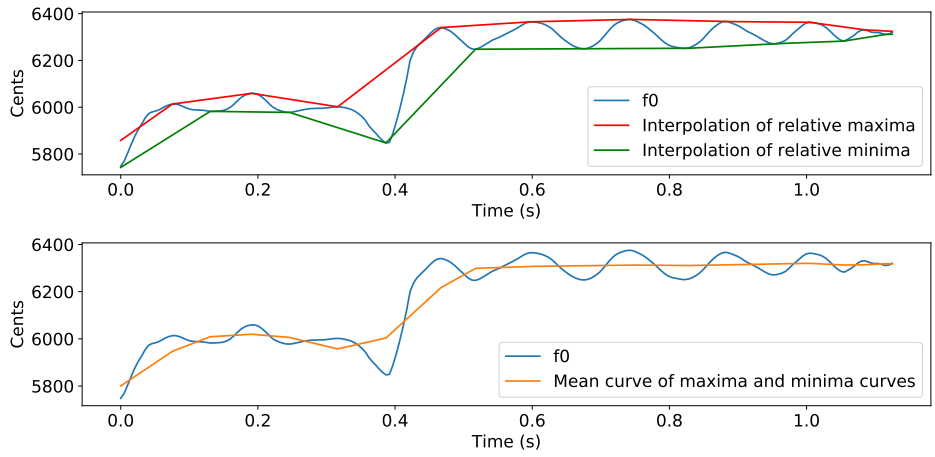


Figure 4.4.9: Vibrato smoothing process

With the smoothing of the f_0 we obtain a curve that is closer to the staggered curve that represents the melody. Our objective is to define the stable areas in this staggered signal to define the notes. When the singing is well tuned, these stable areas are located in the correct representation levels of the note pitch values that can appear in a music score. In our cent scale tuned with the 440 Hz A_4 note, the cent values that represent the note pitch values are all the multiples of the number 100. The natural f_0 signals are not limited to these pitch values: the possible gener-

ated frequency values are defined in a continuous space inside the singers natural range. We consider that if we represent the value of each frame of the f_0 as the closest note pitch in the A_4 scale we can obtain a good approximation of the staggered curve that the singer is trying to sing. We can observe this process in Figure 4.4.10. As a result of this process, the real smoothed f_0 curve (in blue) is substituted by the discretized curve (in orange). Looking at the figure we can observe that the discretization of the f_0 in the cent scale shows that sung frequency maintains the pitch around the cent levels of 6500, 6300 and 6200 in different moments of the signal. These values represent the notes with the MIDI number 65, 63 and 62 respectively. These MIDI numbers are the equivalent of the notes F_4, $D\#_5$ and D_5.

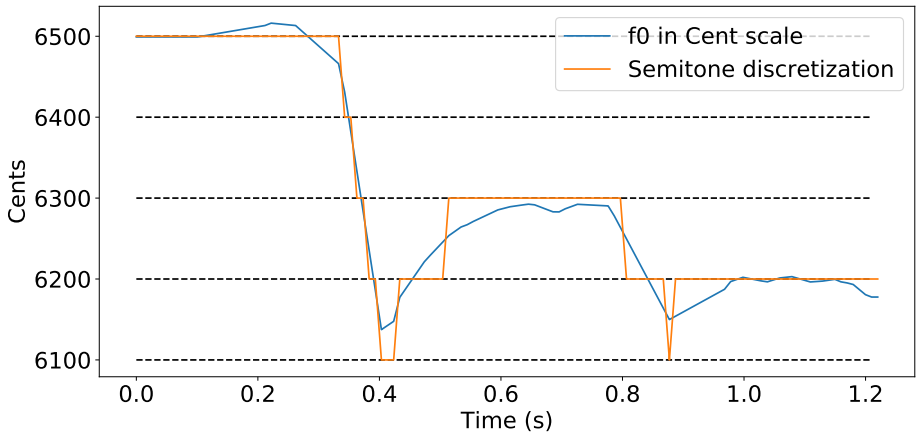


Figure 4.4.10: Discretization of a f_0 curve

Our objective is to detect these ”stable” areas around note pitch values. In a perfectly tuned voice, we could search for sequences of a unique pitch value, but the perfect tuning is complicated even for professional singers. Considering that the standard properties of the notes are a limited range of 100 cents and a minimum duration of 150 ms in Western music [21], we search for sequences of frames in the f_0 curve that meet these requirememts. To search sequences with minimal duration and maximum range we use an algorithm that uses subsequence search techniques [75, 100]. We search sequences in the discrete curve that fulfill the min-

imum conditions of length and stability defined in expressions (4.13) and (4.14):

$$Len(s) \geq L, \tag{4.13}$$

$$max(s) - min(s) \leq R, \tag{4.14}$$

where s is the note pitch subsequence, R is the maximum amplitude range and L is the minimum length allowed. The detailed steps of the proposed algorithm are defined in Algorithm 3.

Figure 4.4.11 shows the application of the algorithm with configuration parameters 100 cents and 150 ms to an example signal. We can see how a note detected by the algorithm would look (green line) and how the rest of the f_0 sequence is split into two smaller sequences. These two new sequences will be added to the sequence list to be analyzed recursively if they meet the minimum duration requirements.

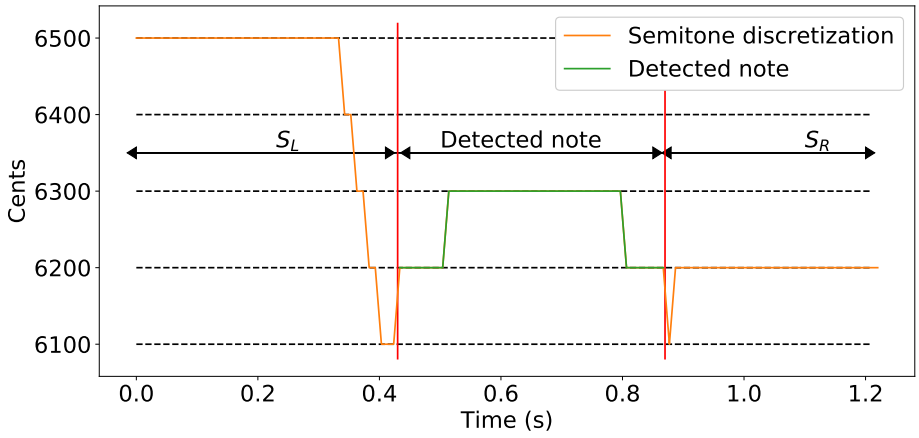


Figure 4.4.11: Detected musical note and new split sequences

With this method, the stable areas are correctly detected but the proximity of

Algorithm 3: Note detection.

Data: Sequence of K voiced sequences of discretized values
Result: Note list

```
1  begin
2      S ← Sequence of K voiced sequences of discretized values
         S = {S_1, S_2, ..., S_K};
3      L ← Minimum note length;
4      R ← Maximum tone variation;
5      N ← Empty Note list;
6      while length(S) > 0 do
7          S' ← empty New sequences list;
8          for S_i in S do
9              Find longest s that fits Len(s) ≥ L and
                 max(s) − min(s) ≤ R;
10             Save s in N;
11             S_Li ← Sequence left to s in S_i;
12             S_Ri ← Sequence right to s in S_i;
13             if Len(S_Li) ≥ L then
14                 Include S_Li in S'
15             end
16             if Len(S_Ri) ≥ L then
17                 Include S_Ri in S'
18             end
19         end
20         S ← S';
21     end
22 end
```

adjacent note pitch values can create problems obtaining note boundaries. In Figure 4.4.12 we can observe what can happen with contiguous notes that have one semitone of distance between them. In the top graphic of the figure we can observe the f_0 curve of two notes: the first one goes from second 0.1 to 0.4 and the second note goes from second 0.4 to second 1.2 approximately. In the middle plot of the figure a discretization of f_0 considering note pitch values has been applied. With note pitch discretization, the stability analysis cannot detect the note separation

in the second 0.4. In the bottom graphic a discretization of higher definition has been used, with a separation of half a semitone between different pitch values. In this case the stability analysis is correct and both note areas are detected separately.

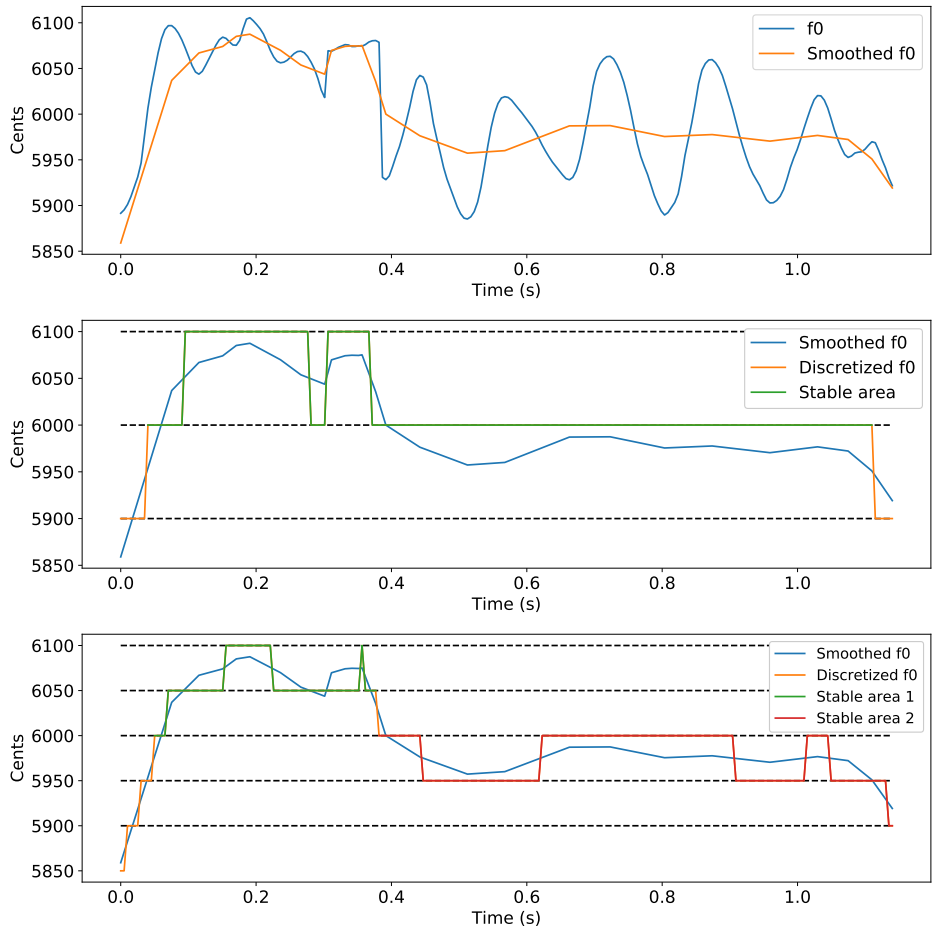


Figure 4.4.12: Stable sequence detection with different discretization definition

As the distance between note pitch values in a music scale is a semitone, we name the discretization definition level as "steps per semitone". The steps per semitone define how many equally spaced discretization levels are defined to shift a semitone. If we use one step per semitone, the discretization levels are the note

pitch values of A_4 tuned music scale. If we use two steps per semitone, a discretization level is created halfway between all note pitch values, and so on. The steps per semitone can be increased until all the possible values of the f_0 are considered as discretization levels. We named that particular case as infinite steps per semitone.

After the stability regions are defined, the new note pitch values have to be defined for each region. As in bertsolaritza the tuning is difficult to define, we decided to define each region with a continuous frequency value instead of a discrete note definition. The frequency to define each region is calculated with the median of the smoothed f_0 in the region in cents. In Figure 4.4.13 we can see an example of the detection of the pitch in stable regions using infinite discretization values.

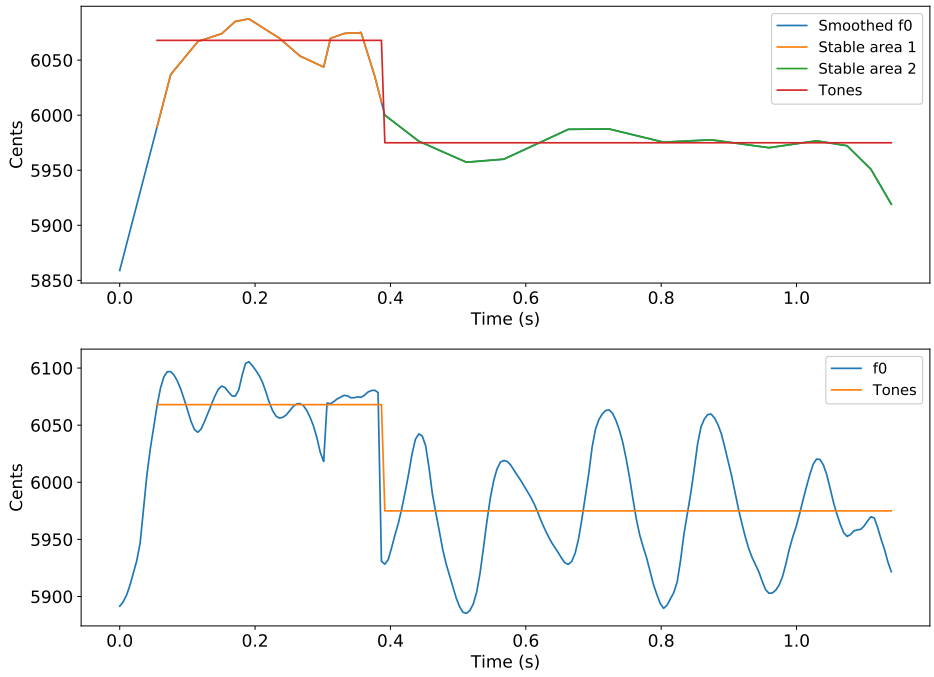


Figure 4.4.13: Tone definition in stable sequences

This system has no ability to define the note boundaries between two contiguous notes with the same pitch. This can be a problem if the result of this algorithm would be our only reference when musically analyzing the recordings. Our

intention is to consider also the phoneme boundaries to define the potential note boundaries and solve the problem of consecutive notes with the same pitch.

4.4.3.1.1 Analysis of the parameter sensitivity of the note detection algorithm

The note detection algorithm we have proposed in this Section offers some flexibility with respect to the characteristics of the notes it must detect, namely different length and range values can be taken into account. In addition, we defined another parameter that defines how many steps are considered between note pitch values in the discretization. The algorithm in Tony uses three steps between note pitch values in the detection of note levels [93]. In order to check if these configuration parameters have an important impact in the detected notes, we devised an experiment to test the sensitivity of the results to these parameters.

In Section 4.1 we used frame level note detection information to discriminate speech and singing voice. As good note detection is related to the performance of the speech and singing voice discrimination algorithm, we think that note labels obtained with optimal note detection parameters will obtain the best discrimination scores. In this experiment we applied our note detection algorithm with different minimum length, maximum range and steps per semitone to later discriminate speech and singing using the detected notes. We considered maximum ranges from 100 to 600 cents with intervals of 50 cents and minimum lengths from 50 to 450 ms with intervals of 50 ms. The data-set used is a Bertso database excerpt that was defined in Section 4.1.1.3.1. To evaluate the classification, we used a 10-fold cross-validation with a joint F-score test. The F-score results with different note detection parameters are shown in Figure 4.4.14. We also tested the method with infinite discretization values. The F-scores obtained with this method are shown in Figure 4.4.15.

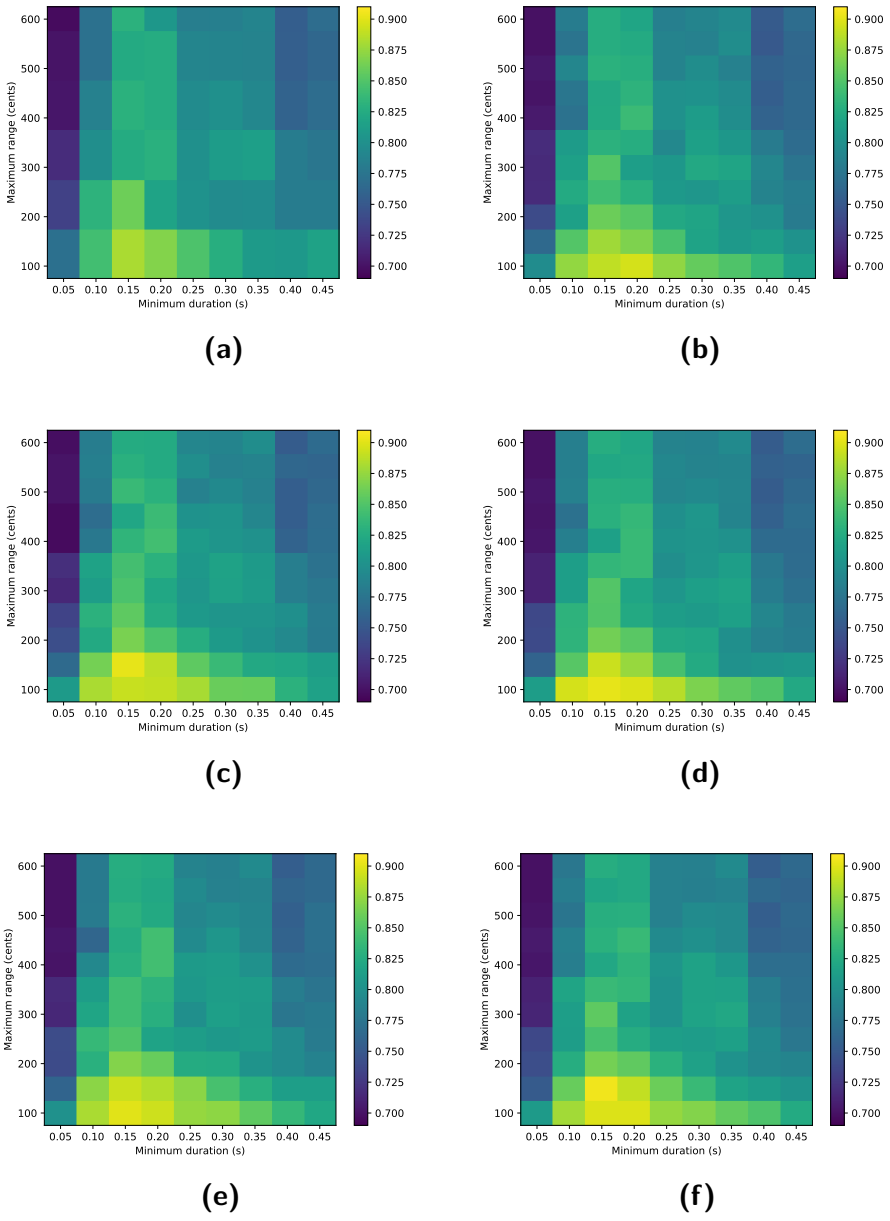


Figure 4.4.14: Speech/singing classification F-score for different number of steps per semitone in note detection algorithm. (**a**) one step per semitone; (**b**) two steps per semitone; (**c**) three steps per semitone; (**d**) four steps per semitone; (**e**) five steps per semitone; (**f**) six steps per semitone

Figure 4.4.14 shows that, when the note algorithm parameters get close to the values common in Western music (minimum duration in the range of 100–200 ms and maximum pitch range between 100 and 150 cents [21]), the discrimination between singing and speech improves. This means that the optimum value for the parameters has a strong relation with the style of singing that we want to discriminate from normal speech. The vertical resolution in Figure 4.4.14a is half of the resolution shown in the rest of the cases because one step per note pitch value does not allow for having different results between pitch values.

We also tested the parameter setting where the possible pitch steps are infinite. The results obtained with continuous pitch steps are shown in Figure 4.4.15, where a similar pattern to the one observed in the case of using the discretized f_0 curve can be seen: the best F-score is obtained for the common values of minimum duration and maximum pitch range in Western music.

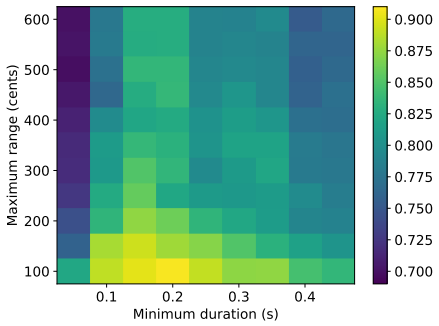


Figure 4.4.15: Speech/singing classification F-score for continuous f_0

The parameter combination that provides the best discrimination results for each of the different semitone step number is presented in Table 4.4.3. F-score improves as we use more semitone steps and the best result is obtained with continuous f_0, which corresponds to a situation where infinite steps are considered. Nevertheless, the differences in F-score are slight and, providing that the maximum pitch range and minimum note length have values suitable for Western music, the

Steps per Semitone	Minimum Duration (s)	Maximum Range (Cents)	F-Score
1	0.15	100	0.882
2	0.20	100	0.896
3	0.15	150	0.902
4	0.15	100	0.902
5	0.15	100	0.900
6	0.15	150	0.905
Continuous	0.20	100	**0.908**

Table 4.4.3: Best result of each semitone step division level

number of steps considered has a small influence in the results.

4.4.3.1.2 Comparison with a standard note detection algorithm

We have compared our note detection system with Tony, a state-of-the-art algorithm that uses HMMs with three states per note (onset, stable and offset) to detect notes in multiple pitch tracks calculated emphasizing different frequency ranges. As our speech/singing discrimination algorithm only uses the 'note/no note' decision, we have compared precisely this aspect between the two algorithms. With this purpose, we have used the Bertso database explained in Section 3.1.1. We have labeled 10 ms spaced frames with a binary label (note/no note) with both algorithms and calculated the agreement between systems using the kappa score [28] in each audio file. We used our note detection algorithm with one step per semitone, 150 ms minimum note duration and 100 cent maximum amplitude. The histogram of these scores is shown in Figure 4.4.16.

We can observe that the agreement between both note detection systems is strong for most files in the database, with a mean kappa of 0.73 ± 0.08.

4.4.3.1.3 Parameter selection for final labeling

Observing the results of the different experiments performed with our note detection algorithm, we decided to use a minimum duration of 100 ms and a range of

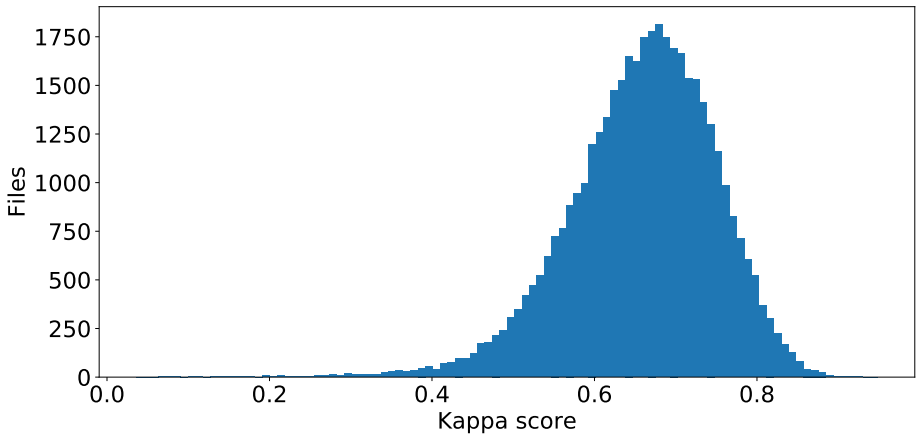


Figure 4.4.16: Histogram of kappa score between notes detected by Tony and our algorithm

100 cents to detect notes. We noticed that 100 cents is the optimal range to define note boundaries correctly. In respect to the minimum duration, we know that 150 ms is a theoretical definition of a note [21] and the results in Table 4.4.3 confirm that it is optimal for speech and singing discrimination. Nevertheless we realized that we are analyzing only the stable state of the notes and that the attack and the offset of the notes are included in the theoretical 150 ms measure. This makes it difficult to detect short notes and therefore we decided to reduce the minimum duration of the notes from 150 ms to 100 ms. We think that in our speech and singing voice discrimination experiment these small notes have not affected to the discrimination F-score because there are not many in the database. Now that we use our algorithm to annotate exclusively singing voice, we think that using a more conservative threshold is the best option. Changing this threshold does not change the detection of notes longer than 150 ms, but in addition to these notes more short notes are going to be detected.

4.4.3.1.4 Phoneme and note combination

We have devised an algorithm that identifies notes in a f_0 curve, but our final objective is to create the music scores of the bertso recordings. This means that we have to relate the phoneme alignments obtained in Section 4.3 with the notes detected with our algorithm. Our supposition is that as in bertsolaritza the metric is an important aspect, and therefore each syllable can be associated with as a note when bertso melodies are interpreted. As we use voiced phoneme f_0 to detect notes, we refer to the boundaries of the segment that starts in the beginning of the first voiced phoneme of the syllable and ends in the end of the last voiced phoneme of the syllable as voiced phonetic boundaries. Considering the voiced phonetic boundaries of each syllable, we can use the notes detected inside these boundaries to define the pitch of the syllable. This would be easy if the notes and voiced phoneme boundaries would fit perfectly, but the note and syllable onsets are not aligned in singing voice [123]. This is why multiple note regions corresponding to different pitch values may appear inside the voiced phoneme boundaries of each note. To define the note pitch in this situation, we decided to select the pitch with the highest number of frames inside the voiced phoneme boundaries of the syllable. In Figure 4.4.17 we can see that notes can be shared between contiguous syllables. In the /*n*/ phoneme between 1.0 and 1.3 seconds and /*m*/ phoneme between 2.8 and 3.0 seconds we can see that the tonal transition between notes can occur inside the phonetic boundaries of the syllable. We can also see that the note between 1.5 seconds and 2.8 seconds is unique although it corresponds to 4 notes with the same pitch value. The note boundaries are not detected by the note detection algorithm, but each syllable can be labeled with the correct pitch using phoneme boundaries.

In Figure 4.4.18 we can see the final labeling of the notes using phonetic boundaries of the syllables and the notes detected by our algorithm. We can clearly see that the the natural f_0 has strong variations around the sung melody.

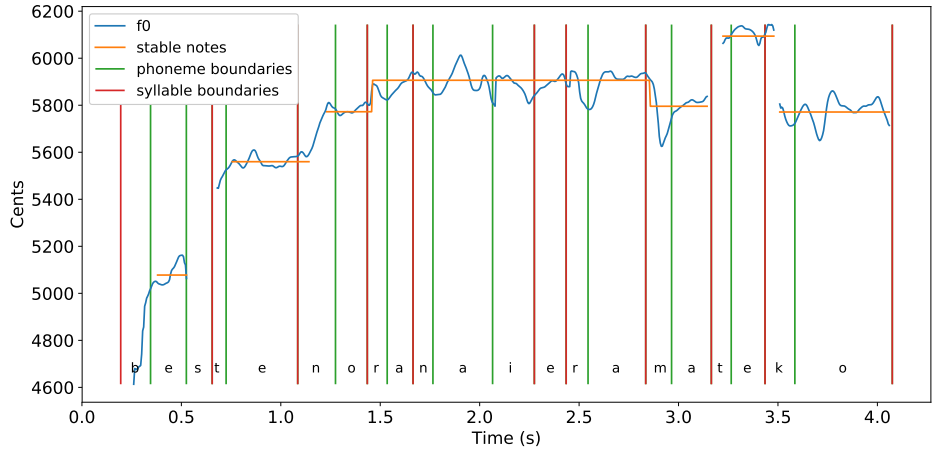


Figure 4.4.17: Detected notes in phonetic syllables

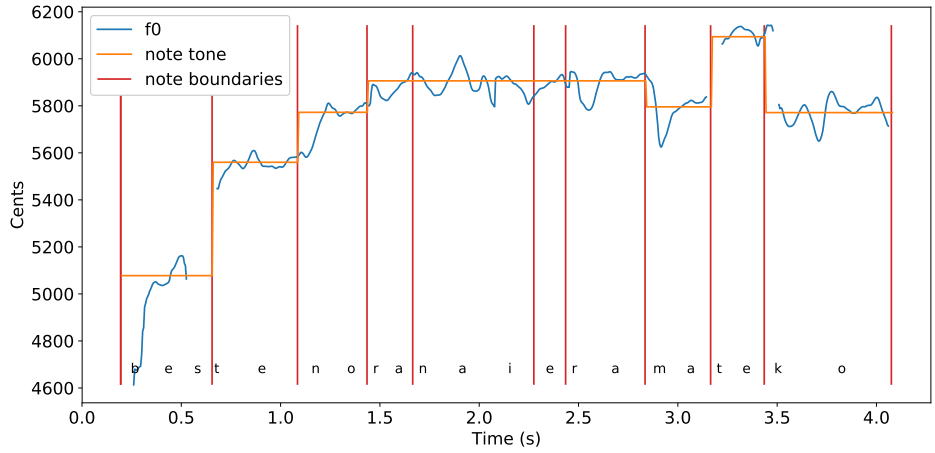


Figure 4.4.18: Defined note in phonetic syllables using majority vote

4.4.3.1.5 Note generation problematic

In the previous section we used the combination of the phoneme alignment and detected notes to label the pitch values in the recordings. We observed that some phoneme boundaries do not contain detected notes or have a small ratio of note frames in voiced f_0 frames. This means that these syllables do not include voiced f_0 areas with enough stability to be considered a note. The number of syllables with

a small note ratio is small in the database, but as we use f_0 normalization in the singing synthesis as explained in Section 5.2.1, this error can limit the database in a considerable way. In the f_0 normalization process, f_0 curves are normalized using the note pitch values to model only the difference between the ideal pitch and the natural f_0. If we cannot annotate a pitch in a note or if we obtain a low confidence annotation, a good normalization cannot be applied to the f_0. In these cases, the normalized f_0 signal would not be suitable for the training database. This is why having notes with no pitch label in a utterance makes the utterance being discarded from the training data. Considering this, each of these notes with no pitch reduces the database in a considerable way. This is why we decided to analyze these notes and propose a secondary pitch definition method for these notes.

In the database with the phoneme alignments defined in Section 4.3.4 there are 347068 notes. From these notes, 164 do not include any voiced frame inside the voiced phonemes which represents the 0.04 % of the total notes. We show the distribution of the ratio of note frames and voiced f_0 frames in Figure 4.4.19. We have not included the 164 notes with no voiced frames in the figure.

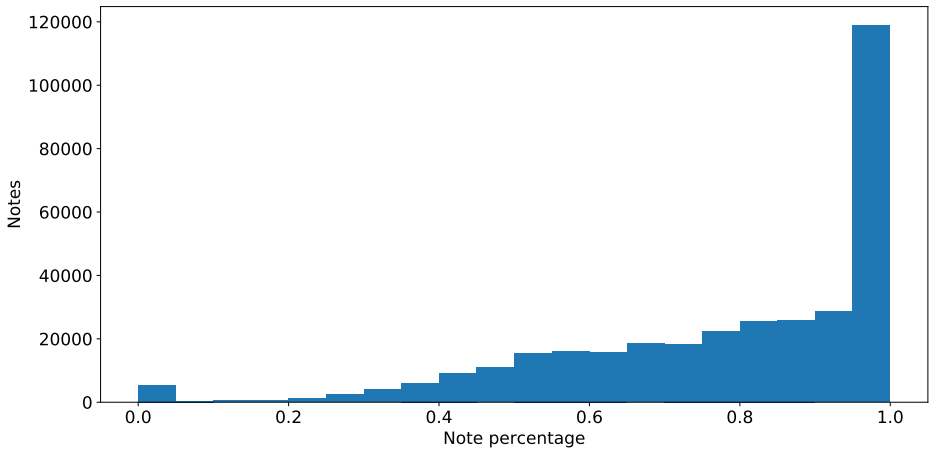


Figure 4.4.19: Histogram of note percentage in voiced phonemes

Observing the data we defined 0.2 as the minimum note ratio to accept the labeled notes as a reference to define the pitch. We analyzed the notes with a note

ratio lower than 0.2 and we observed two cases: general f_0 instability and portamento. With these considerations the unlabeled notes may be obtained due to three different reasons:

- **Missing voiced frames**: The f_0 of the voiced phonemes has no voiced frames.
- **Portamento**: We observed that in the initial note of the utterances and in some intra-utterance notes, the bertsolaris often sing notes that are almost entirely a transition either from a low pitch to the initial note or from one note to another. Neither of these cases present any stable area.
- **Instability**: Some notes contain unstable f_0 values that do not fit in the portamento structure.

The first phase of this secondary note annotation is to determine which of the three defined types of unlabeled notes corresponds to each note. The detection of notes with missing voiced frames can be made with a simple voiced frame count. The portamento and the unstable notes are notes with potential note analysis, therefore they are not distinguishable with a simple voiced frame count. We know that the portamento is a slow transition between two pitch values, therefore the pitch in portamento constantly increases or decreases. This is why it is difficult for our algorithm to detect the pitch in these notes and we propose to detect it using the derivative of the f_0. We observed that in bertsolaritza the portamento is mainly realized increasing the pitch, therefore we define an unlabeled note as portamento if the 70 % or more of the derivative of the f_0 is positive. The rest of unlabeled notes with voiced frames are classified as unstable.

We defined a solution to each of the causes of unlabeled notes to create the final pitch labeling.

- The notes with missing voiced frames are impossible to label and therefore we decided to leave them without pitch. As previously said, these notes represent only the 0.04 % of all notes and as the utterances containing these

notes cannot be used for synthesis, we will discard them. A total of 148 utterances with no voiced frames notes are thus excluded from the database.

- In the unstable and portamento cases we devised a secondary note detection method to define the note pitch that uses the raw f_0 in contrast with the smoothed f_0 used in the note detection algorithm.

  – In the unstable notes, we calculate the median f_0 value of the voiced frames and search for the longest sequence of voiced frames in the range of 200 cents around the median. The final pitch is the median value of this sequence. The process of detecting notes in unstable notes has been visualized in Figure 4.4.20. The note between 1.5 and 1.8 seconds has no clear stable area. Using the median of f_0 inside this note we set the 200 cent range denoted as secondary stable range and highlighted it in green in the Figure. Then, we calculated the median value of the sequence of pitch values in this range and we obtained 5700 cents, which is the assigned note in the secondary note detection procedure.

  – In the portamento case considered, the trend of f_0 is ascendant within the sung note. This is why we defined the labeling pitch as the maximum f_0 value of the voiced frames in the note. The secondary note detection in portamento is visualized in Figure 4.4.21. The first note that goes from 0 to 0.5 seconds has ascendant f_0 values. We set the pitch of the note to the maximum value of the f_0 inside the note, which in this case is 5800 cents.

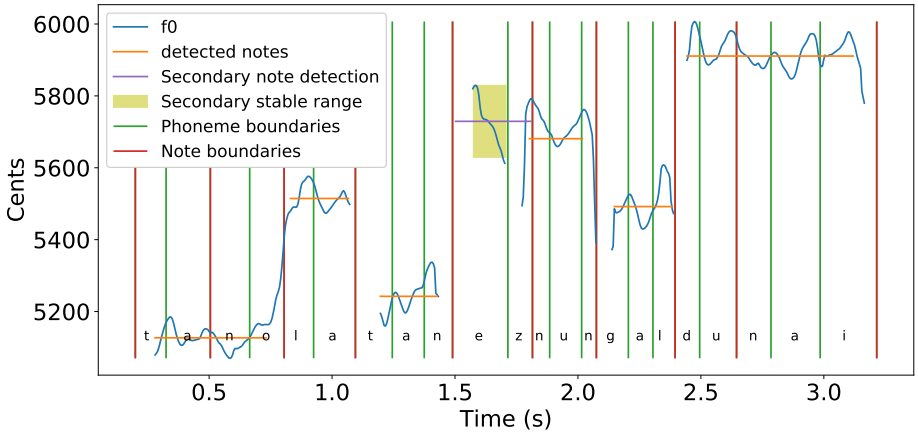


Figure 4.4.20: Note definition in unstable notes

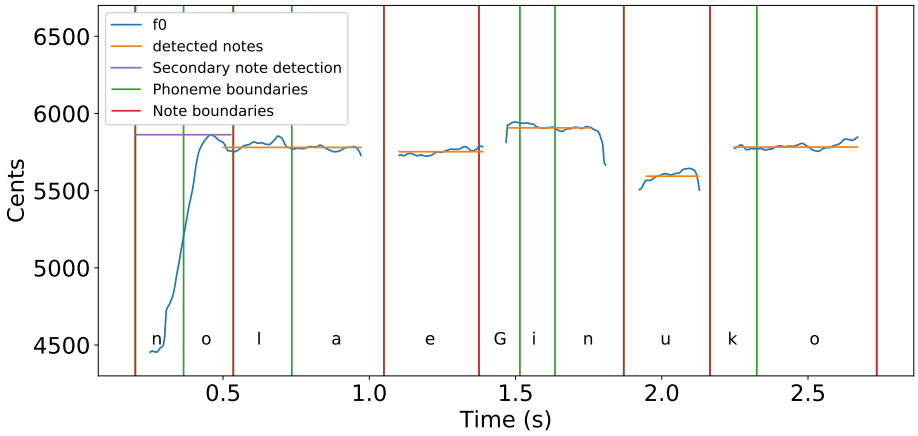


Figure 4.4.21: Note definition in notes with portamento

There are 6977 notes with lack of stability or portamento, representing the 2 % of all the notes. From this group, we detected that the 73 % corresponds to notes with portamento and the remaining 27 % to notes with no stability. We analyzed the position of the notes with portamento in the utterance in Figure 4.4.22. The majority of portamento appears in the first note of each utterance.

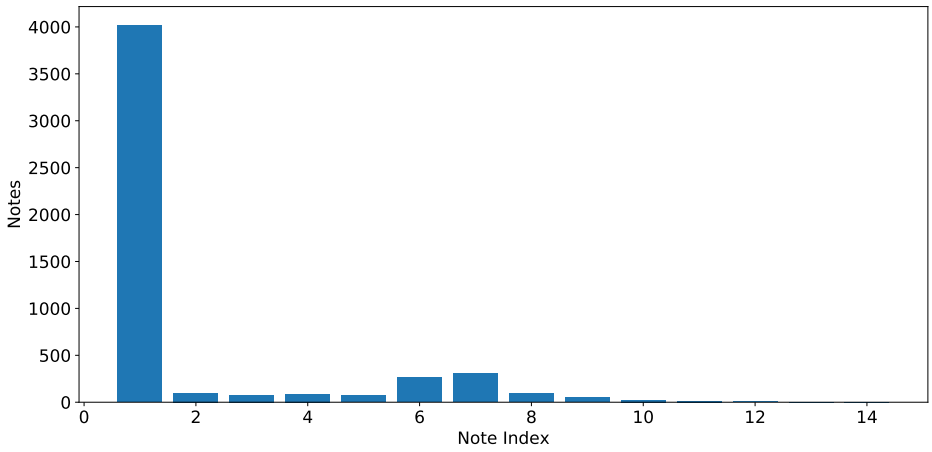


Figure 4.4.22: Distribution of the position in the utterance of the notes with portamento

4.4.3.1.6 Applied note annotation in Bertso database

Using the phoneme alignments, the note detection algorithm and the secondary note detection method, we assigned a pitch label to each note containing voiced f_0 frames in the Bertso database. The final pitch is not an integer number defining the MIDI number of the note, but a float number in cent units that can be located between two note pitch values in the A_4 scale.

4.4.3.2 Duration definition

Given the dynamic nature of the tempo parameter, the creation of the duration labels without the music score is a more complicated problem than that of the note pitch labeling. In the note/frequency relation, the A_4 reference is a common ground in music, although the singers do not use exact semitones of this scale. In Section 4.4.3.1 we defined the notes as continuous values in a frequency space, but the cent base that helps to define pitch changes is based on a static reference that corresponds to assigning 440 Hz to the A_4 note. Using a unique tempo to define a whole music system creates big flexibility issues considering the limitations of the duration symbols frequently used in music scores. As shown in Figure 4.4.8a,

our database needs a wide range of tempos to be defined with low temporal distortion. Although multiple dots and ties can be used to define the duration with more precision, these symbols are not commonly used in music scores and can produce unnatural music scores.

If we consider creating an almost continuous representation of musical durations, we should use a continuous tempo definition too. Using two continuous values to define the duration of a note may create too much complexity to be covered in statistical modeling. We can see in Figure 4.4.8a that the tempos in the edges of the distribution will pose difficulties to be modeled due to the lack of samples. Considering these conditions, we decided to create a discrete musical temporal system that can define a wide variety of durations and tries to minimize the complexity of information.

The simplest solution to solve the tempo complexity is to use the same tempo reference to label all note durations in the Bertso database, in the same way as we use the A_4 pitch as a tuning reference for note pitch labeling. The problem of time precision can be solved using a short note symbol unit and defining any duration as a multiple of this small unit. To fill these two conditions we need a tempo that represents in an optimal way the Bertso database and a short note symbol that can give us enough precision to define the note durations in time.

In 2006, Saino et al. addressed the problem of the unalignment between the note boundaries and music score tempo [123]. As we do not have the score tempo and we have seen the distortion between the music score durations and the recordings of our database, we decided to analyze the correlation between the duration of each syllable and the duration of the phonemes inside the syllable. Figure 4.4.23 plots the correlation and the slope of the linear regression curve between the phoneme durations and the duration of their corresponding syllable by phoneme. Vowels have strong correlation and high slope. The /n/ and /b/ phonemes present notable correlation too with medium slope values. The remaining consonants are grouped in the low correlation and low slope group. The /x/ phoneme also seems to be an outlier among unvoiced phonemes, but as we explained in Section 4.3.3 it is a phoneme with a too reduced representation in the database to be able to extract

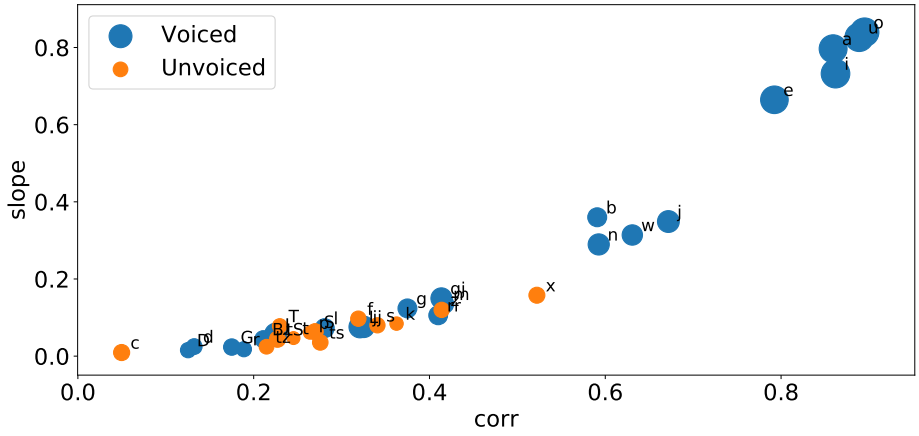
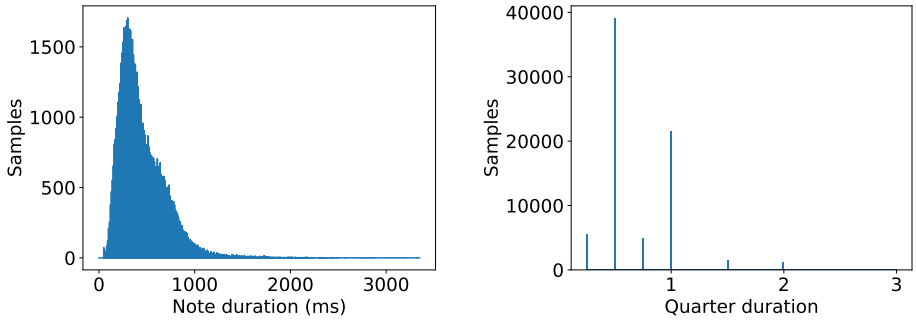


Figure 4.4.23: Correlation level and linear regression slope of phoneme duration with syllable duration, separated by phoneme

conclusions. Considering this chart and that music pitch is highly correlated with the presence of f_0 in the singing voice, we decided to use only the voiced phoneme durations. From now on, every time we refer to the duration of phonemes in a note, we will be considering the duration of voiced phonemes only.

To calculate the optimal tempo for the representation of the whole database we decided to align the distribution of the duration of voiced phonemes and the distribution of musical symbols of all notes in the music scores in the parallel data defined in Section 4.4.2. These two distributions are shown in Figure 4.4.24.

In Figure 4.4.24a, we can observe that the distribution of voiced phoneme durations has a clear peak around 300 ms, it is not gaussian and at the right of the 300 ms peak a change in the slope can be seen around 500 ms duration. The musical symbol distribution in Figure 4.4.24b shows that the most used note symbol is the eighth note with a duration of half when expressed in quarters. The second most used symbol is the quarter note. In a third level we can see the sixteenth and dotted eighth and finally the dotted quarter and the half note. The distribution of symbols is very common and shows that bertsolaritza music scores use symbols commonly



(a) Distribution of durations **(b)** Distribution of musical symbols

Figure 4.4.24: Distributions in Bertso database

used in the music scores in Western music.

For the alignment of these two distributions, we propose a distribution alignment algorithm. As this algorithm is going to be used again later in this work to align other distributions, we will explain it in a generic way. In the distribution alignment problem we define two sample sets, a static set A and a dynamic set B. If we define an integer value space I with a specific range for the set A in Expression 4.15

$$I \in [min, max] \cap \mathbb{Z} \tag{4.15}$$

we can define a function f with n independent variables that transforms the set B to the space I applying expression 4.16

$$B \xrightarrow{f(x_1,\ldots,x_n)} C \tag{4.16}$$

Any sample set from the value space I can be represented as a frequency distribution vector of length $L = max - min$. We can obtain the Probability Distribution Function (PDF) of the sample distribution vector of the static set A by normalizing it with the total area of the distribution. Using the PDF of set A we can define the probability per sample of the set C with Equation 4.17.

$$p_C = \frac{\sum_{i=1}^{N} P_A(s_i)}{N} \tag{4.17}$$

where P_A is the PDF of the set A, s_i is the sample number i of the set C and N is the total number of samples in set C. The solution of the distribution alignment problem can be obtained by maximizing the probability per sample of the transformed dynamic sample set C by optimizing the f transformation function applied in B to obtain C.

In the specific case of the quarter duration and real note duration alignment, f function corresponds to Expression 2.4 and the independent variable of the function is the tempo. The result of the alignment is shown in Figure 4.4.25.

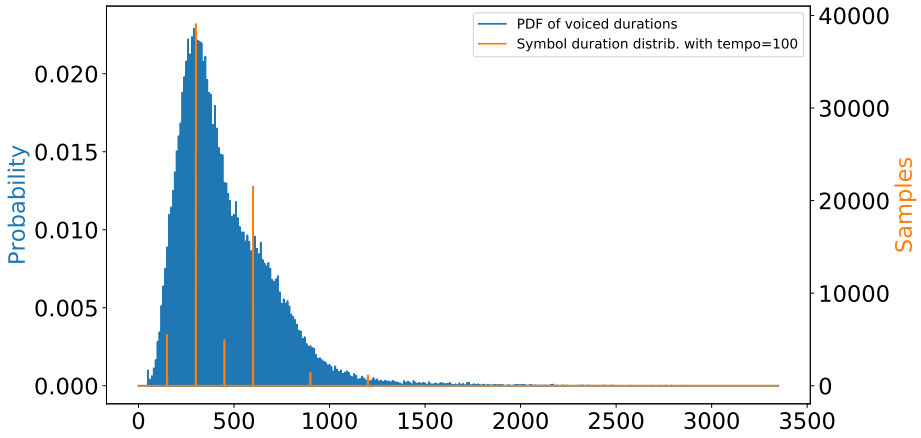


Figure 4.4.25: Alignment of voiced duration PDF of the database and time representation of notes in parallel music scores

The optimal tempo to represent the Bertso database taking into account the distribution of real duration values and note symbols is 100. In Table 4.4.4 we can see the temporal value of different symbols when the tempo value is 100.

The symbol with a duration closest to the standard frame analysis period in speech signal processing is the 512th. Speech analysis standard framerate is 5 ms and using the 512th duration as basic unit, the precision to define a duration of

Symbol	Duration with optimal tempo value (ms)
quarter	600.00
eighth	300.00
16th	150.00
32nd	75.00
64th	37.50
128th	18.75
256th	9.38
512th	4.69

Table 4.4.4: Duration values for each music symbol with optimal tempo

voiced phonemes in a note would be maximum. But using this precision would augment the complexity of the representation of the note duration. We decided to use the 64th symbol to reduce this complexity.

4.4.4 Vibrato labeling

The vibrato is a fundamental characteristic in singing voice. It consists in the modulation of the f_0 of the voice.

We propose to detect the vibrato using every local minimum and maximum in the stable areas detected by our note detection algorithm. First, a Savitzky-Golay filter is applied to every voiced segment of the f_0 using a Window length of 75 ms and a second degree polynomial. After this filtering, we detect all the local maxima and minima. We define any minimum or maximum as a pair of values made up by the value of f_0 and the instant where the minimum or maximum is located, as expressed in Equation 4.18.

$$m_i = (x_i, t_i) \tag{4.18}$$

where x_i is the value of f_0 in cents and t_i is the time in milliseconds. With this definition we can define the sequence M of maxima and minima as

$$M = (m_1, m_2, ..., m_L) \tag{4.19}$$

We can see in Figure 4.4.26 that the segment of f_0 between a maximum and the consecutive minimum can be considered as half a period of a sinusoid. We detect the positions of maxima and minima in the smoothed curve, however the frequency values are not directly taken from the smoothed curve. We consider that we may loose information about the modulation amplitude when we smooth the curve. Therefore, we take the frequency value of the original unsmoothed *f*0 in the detected maxima and minima positions. We define the half-period sinusoids using the distance in time and amplitude between contiguous maximum and minimum values. From the sequence in Equation 4.19 we can obtain the amplitudes A and frequencies F of all the intervals between all the maxima and minima.

$$A = (a'_1, a'_2, ..., a'_{L-1}) \tag{4.20}$$

where

$$a'_i = \frac{r_i}{2} \tag{4.21}$$

$$F = (f'_1, f'_2, ..., f'_{L-1}) \tag{4.22}$$

where

$$f_i' = \frac{1}{2t_i} \tag{4.23}$$

As our references are the positions of the maximum and minimum values, we define the frequency and amplitude of each of these positions by averaging the frequency and amplitude of the sinusoids at both sides of the position. As the first and last positions in the sequence only have one sinusoid no averaging is needed and the frequency and amplitude of this sinusoid is assigned directly. The frequency and amplitude of each maximum and minimum point is thus defined as

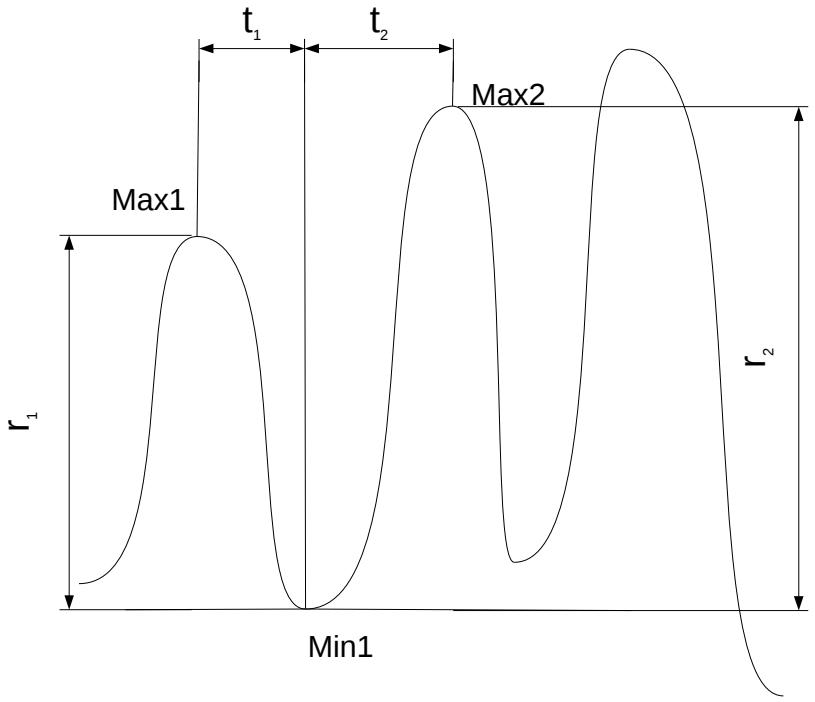


Figure 4.4.26: Vibrato detection

$$f_i = \begin{cases} f_1', & \text{if } i = 1. \\ \frac{f_{i-1}' + f_i'}{2}, & \text{if } 1 < i < L. \\ f_{L-1}', & \text{if } i = L. \end{cases} \tag{4.24}$$

$$a_i = \begin{cases} a_1', & \text{if } i = 1. \\ \frac{a_{i-1}' + a_i'}{2}, & \text{if } 1 < i < L. \\ a_{L-1}', & \text{if } i = L. \end{cases} \tag{4.25}$$

Having the frequency and amplitude values for each minimum and maximum point, we interpolate the values to obtain frequency and amplitude values for each

frame of the signal. All the process is visualized over a natural signal in Figure 4.4.27.

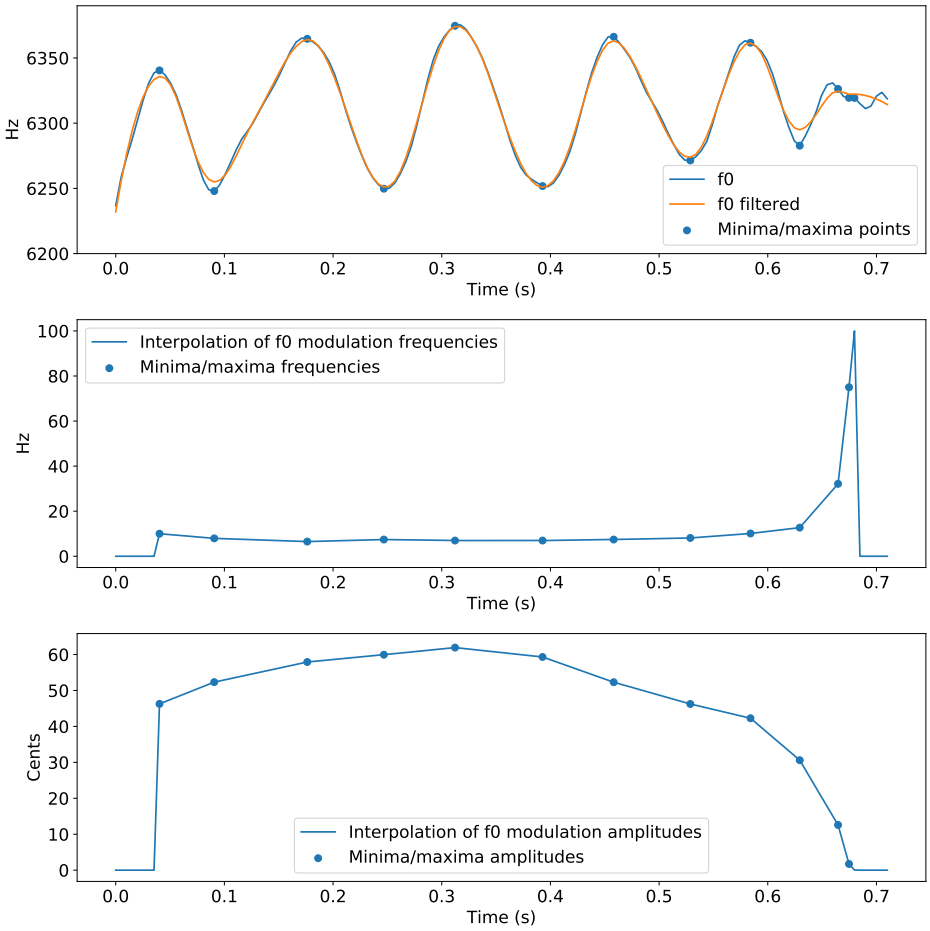


Figure 4.4.27: Frequency and amplitude signals in a natural segment of singing speech

Using the signal of the modulation amplitude and frequency of stable notes we detected the vibrato areas. The modulation is only analyzed if the length of the note is 500 ms or more. The vibrato areas are sequences in time inside detected notes that fill the condition of having a modulation frequency between 5 and 8 Hz and a modulation amplitude higher than 30 cents. We also set a minimum dura-

tion condition for the sequences that fill the amplitude and frequency condition: the length of the sequences has to be 400 ms or more. The detection of these sequences has been visualized in Figure 4.4.28. We can observe in the first plot that the modulation analysis is only made in the notes with duration greater than 500 ms. These notes are denoted as potential notes. In the second plot the modulation frequency condition is shown and in the third plot the condition about the modulation amplitude. We can observe that the note between seconds 2 and 3 contains a small segment that fills the modulation amplitude and frequency conditions. This segment cannot be classified as vibrato because it does not fill the minimum vibrato duration condition set at 400 ms.

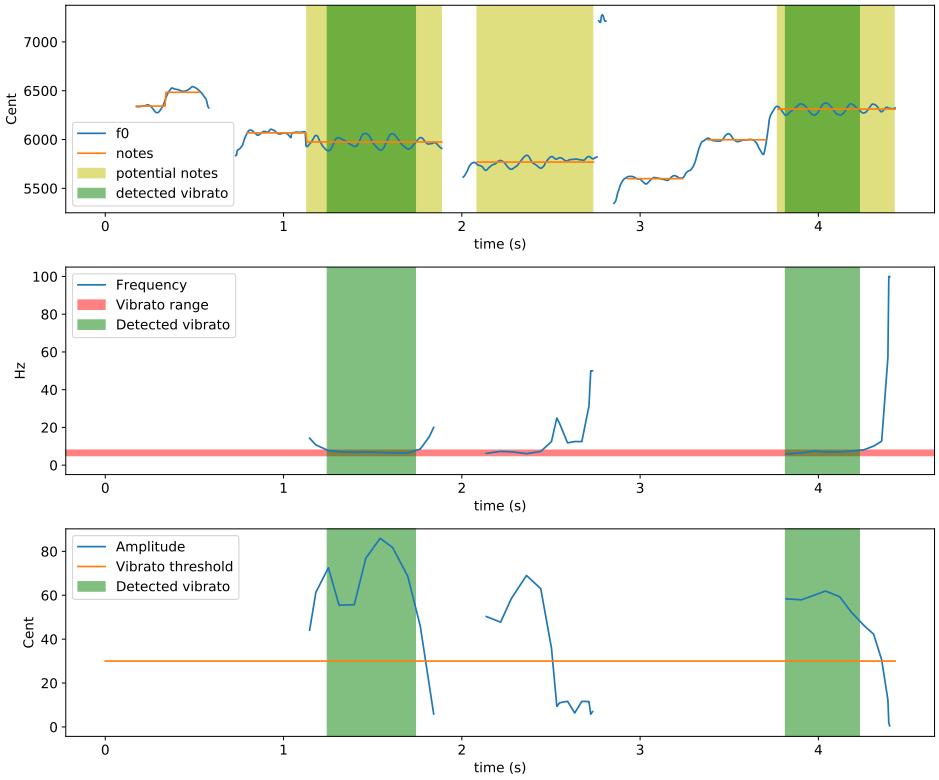


Figure 4.4.28: Vibrato detection using continuous amplitude and phase modulation parameters

The final amplitude and frequency modulation signal is created removing all the values that are outside the vibrato areas. Therefore, the existence of values in frequency and amplitude modulation signal indicates the existence of vibrato in the frame, in a similar way as happens with the voiced/unvoiced decision in f_0.

4.4.5 Melody detection

In Section 4.4.2 we have analyzed the differences between the bertso melodies and the actual performances of bertsos. In Section 4.4.3 a method to get the actual musical labeling of these bertsos without using music scores has been proposed. Although we observed differences between the performances and the melodies in the database, the melodies are recognizable in the recordings and therefore the melody label is a valuable metadata information in the structure of the database. In consequence, it would be interesting to evaluate the possibilities to detect the base melody of a recording taking into account that the interpretation introduces changes compared to the original music score. If we consider the limited possible melodies that a bertso may use as base, we can define a method to decide which of the potential melodies is the most similar to the sung bertso.

4.4.5.1 Dataset

As explained in Section 4.4.2, some recordings with melody label do not contain the same number of utterances in each bertso as the respective melodies, therefore we will only consider for this melody detection experiment the recordings with the same number of utterances as their melody. The data that fill the utterance condition is characterized in Table 4.4.5. To compare the three classification systems, we need fully labeled utterances in musical and phonetic level. Many utterances have been discarded because the impossibility of labeling and this creates a problem when classifying bertsos with unlabeled utterances. As the comparison of bertsos with music scores is made utterance by utterance, we decided to include the bertsos with missing utterances in the experiment. The classfification of these bertsos

is made comparing only labeled utterances. For this reason, the total number of bertsos and the total number of utterances are not coherent taking into account the utterance number the bertsos

Utterances in bertso	Total number of utterances	Total number of bertsos	Potential melodies
8	1399	245	6
10	451	49	8
16	1804	121	2
All	3654	415	16

Table 4.4.5: Melody prediction experiment data

4.4.5.2 Classification method

As we segmented the bertso utterances, our approach is to create a method to compare the recordings utterance by utterance with the music scores. We use DTW to align different sequence representations of the bertso utterances and utterances in the music score. Then we calculate the distance between these sequences and select as the predicted melody utterance the candidate with the lowest distance. We considered three different representations of an utterance to align them applying DTW:

- f_0 sequence: the voiced part of the f_0 of the utterance in cents is aligned with the representation of the pitch values of the music score. This DTW alignment needs to apply the mean shift alignment used in Algorithm 2. An example can be seen in Figure 4.4.29.
- Note sequence: the labeled notes in the utterance are aligned with notes on the music score utterance. This DTW alignment also requires to apply the mean shift alignment used in Algorithm 2. An example can be seen in Figure 4.4.30.
- Differential of notes sequence: the differential of the note sequence is used to align with the differential of the notes in score utterance. The differential

has been calculated using expression 4.26

$$diff[i] = n[i+1] - n[i] \tag{4.26}$$

where *diff* is the differential sequence and n is the note sequence. Differential does not require to apply mean shift alignment, because it uses a sequence of relative values. An example can be seen in Figure 4.4.31.

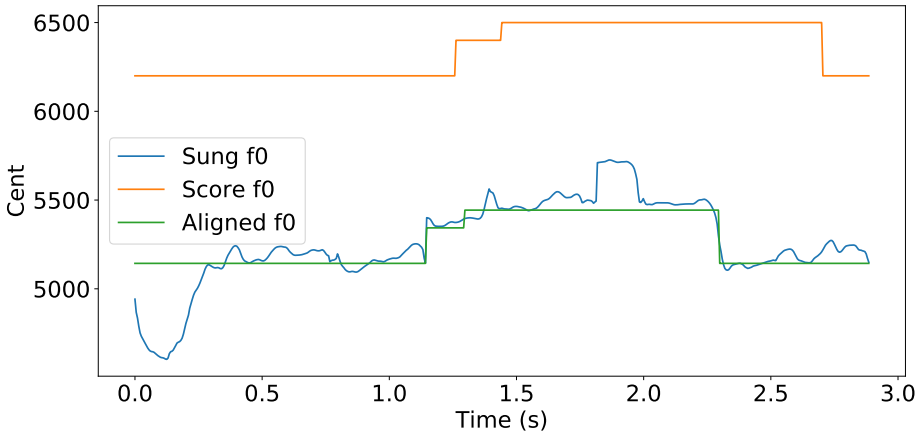


Figure 4.4.29: Alignment of sung f_0 and music score f_0

We observed that it is common to obtain ties between different melodies in aligned sequence distances when applying the last two methods, i.e., note and differential comparisons. This is why we included another parameter to select the most similar music score. We are considering the distance between the original sequence and the aligned one, but we do not take into account if the the original unaligned music score was similar to the sung melody. One way of estimating this similarity is to use the alignment path of the DTW. One example of alignment path can be seen in Figure 4.4.32.

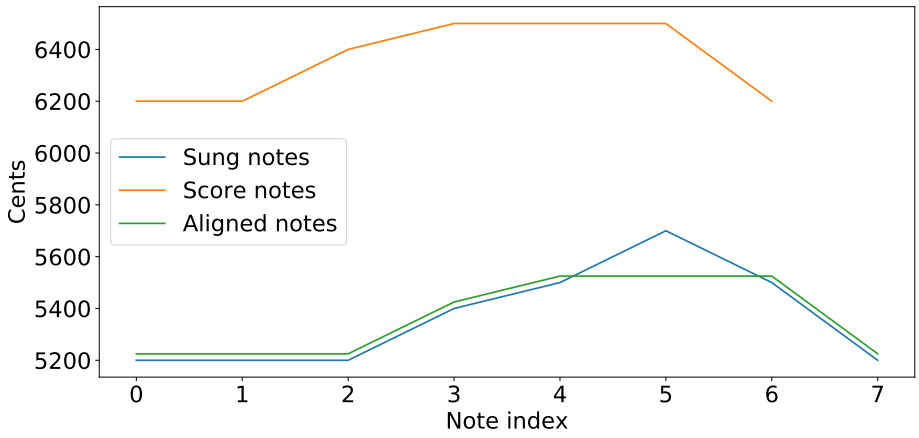


Figure 4.4.30: Alignment of sung notes and music score notes

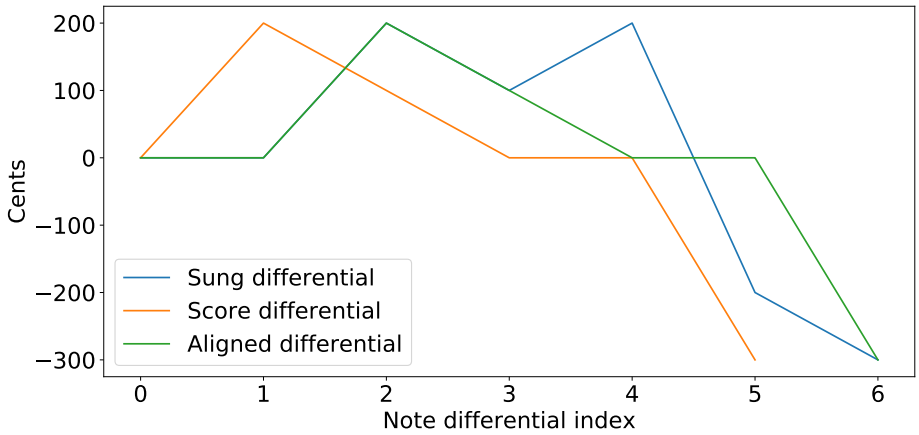


Figure 4.4.31: Alignment of sung note differential and music score note differential

In Figure 4.4.32 we can see that at the beginning of the sequences an adjustment had to be done to optimize the alignment. Our objective is to evaluate how diagonal the resulting alignment is; we consider that the more diagonal the alignment, the closer is the melody to the sung voice. In a DTW alignment there are only 3 possible transitions, so we count how many of these transition have been diagonal. We calculate a diagonality ratio of the alignment with the next equation

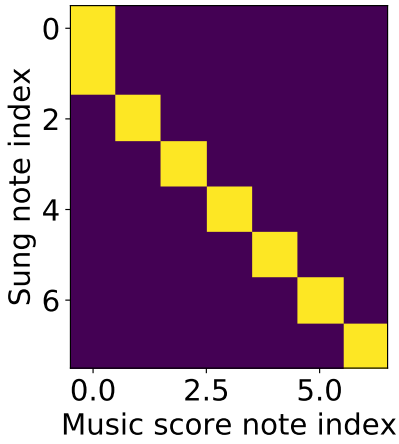


Figure 4.4.32: Alignment path of sung notes and music score notes

$$r_d = \frac{t_d}{t} \tag{4.27}$$

where r_d is the diagonality ratio, t_d is the total number of diagonal transitions and t is the total number of transitions. In case of ties, the melody with the higher diagonality ratio is the one selected as the prediction.

As the main objective of the system is to classify whole bertsos, we also created a method to combine the melodies assigned to all the utterances in a bertso to obtain the melody of the bertso. We propose two methods to make this combination:

- **Majority vote**: Each utterance is classified independently and a majority vote is applied to obtain the melody of the bertso.
- **Distance sum**: In this methods we calculate the alignment distance of each bertso to all the potential melodies and select the smallest distance melody as prediction. The alignment distance of a bertso to a melody is calculated summing up the alignment distances between all utterances in the bertso and the parallel utterances in the melody.

In the bertso classification methods we observed that no ties occur and there-

fore we have not applied the diagonality ratio.

4.4.5.3 Results

The accuracy obtained in each of the methods in the utterance and bertso classification is shown in Table 4.4.6 and Table 4.4.7 respectively.

Utterances in Music score	Number of utterances	Possible music scores	Accuracy (%)		
			f0	notes	Diff
8	1399	6	40.03	48.32	55.54
10	451	8	64.75	53.88	65.41
16	1804	2	71.56	74.33	85.42
All	3654	16	58.65	61.85	71.51

Table 4.4.6: Accuracy results in utterance classification

Utterances in music score	Number of bertsos	Possible music scores	Accuracy (%)					
			Majority vote			Distance sum		
			f0	Notes	Diff.	f0	Notes	Diff.
8	245	6	63.67	71.84	74.29	73.88	82.86	80.00
10	49	8	89.80	89.80	87.76	85.71	77.55	77.55
16	121	2	93.39	98.35	99.17	95.87	100.00	100.00
All	415	16	75.42	81.69	83.13	81.69	87.23	85.54

Table 4.4.7: Accuracy results in bertso classification

With the results we can draw these conclusions:

- Bertso classification obtains better results than utterance level classification.
- The optimal method to classify a single utterance is the differential sequence alignment distance.
- In bertso classification, the distance sum method obtains better results than the combination by majority vote. In bertsos of 10 utterances we can see that the majority vote is a better system but in bertso of length 8 and 16 the

distance sum is better. In the overall results the distance sum also obtains the best scores.

- The optimal method to classify a bertso is using the sum of note sequence distances.

4.5 Resulting databases

In this chapter we created different methods to label the Bertso database. After the singing voice detection, utterance segmentation, phoneme segmentation and musical labeling we obtained a properly labeled singing voice database. We discarded several recordings and bertso utterances in this labeling process to improve the uniformity and the quality of the final database. In the process we developed multiple systems for music labeling and we used the melody alignment system to align the melodies of the NUS database to the recordings. In this Section we analyze the characteristic of this final version of the Bertso database we created, as well as the singer range characteristics of the recordings aligned in the NUS database. We have also labeled the singing utterances of the hosts in the Bertso database, but we do not consider these recordings part of the final database. This is why this description of the database considers only the bertsolari recordings.

4.5.1 Bertso database

The Bertso database obtained is a multi-singer dataset, featuring 176 singers. The general properties of the database considering the gender of the Bertsolaris are shown in Table 4.5.1.

Gender	Number of Bertsolaris	Utterances	Total length (min)
Male	145	34880	2561.53
Female	31	5270	403.60
All	176	40150	2965.13

Table 4.5.1: Number of singers and utterances and total durations per gender in the labeled final Bertso database

There are 176 singers in the database and these are too many to characterize each of them in a detailed numeric way. This is why we decided to use anonymous general characteristics of the database for the analysis, as in Section 4.1.2. We have visualized the characteristics of the recordings of each bertsolari ordering the representation with the value of each parameter in increasing order. This type of visualization means the Bertsolaris in the horizontal axis are in different order in each plot, but this representation provides a clearer information about the general characteristics of the database. In Figure 4.5.1 we show the total duration of all the aligned phonemes for each bertsolari and in Figure 4.5.2 we can see the number of utterances for each bertsolari. In the figures we represent the number of utterances and recording duration of each bertsolari and singing host but we have not displayed any names in the horizontal axis. We avoid writing the singer names in the horizontal axis because they are too many and it would be impossible to see the value of each name.

In the figures we can observe that the majority of the bertsolaris in the database have a small amount of recordings, but there are multiple bertsolaris that have more than one hour of recordings. The female bertsolaris have less recordings and only a single female performer is above the one hour duration threshold.

We also analyzed the distribution of the year in which the recordings where realized. Figure 4.5.3 shows the distribution of the years of the recordings for each singer with a boxplot. We can observe that the database covers the span of 35 years of bertsolaritza and different singers have recordings in different time spans. The

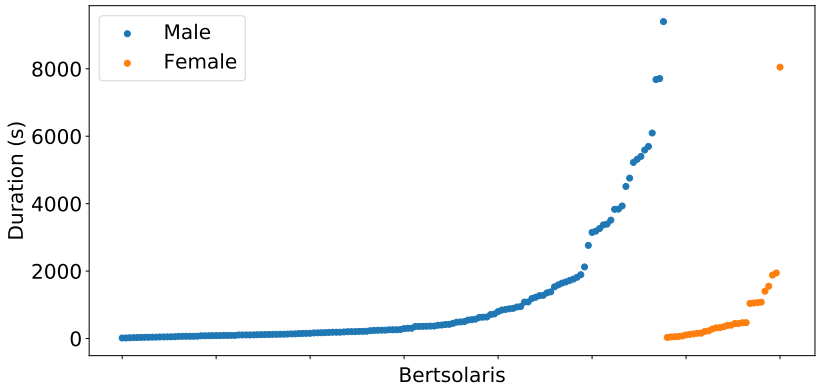


Figure 4.5.1: Distribution of recording time per bertsolari

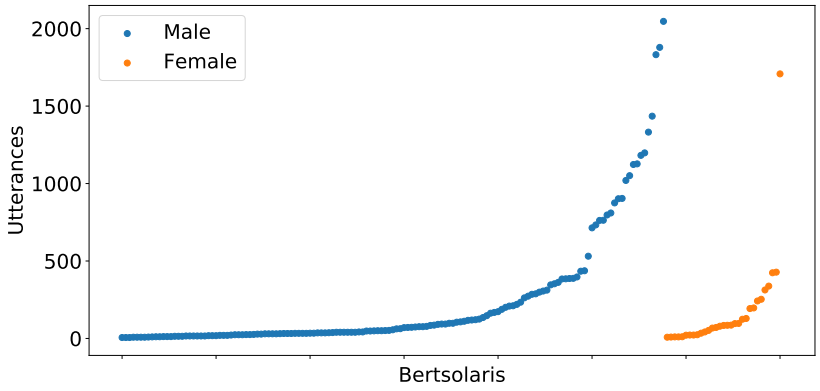


Figure 4.5.2: Distribution of number of utterances per bertsolari

moment when female bertsolaris were introduced in the competitions and public performances can be clearly seen in the figure.

The analysis of the note characteristics for each bertsolari is visualized in a box-plot with the note duration analysis in Figure 4.5.4 and the note pitch analysis in Figure 4.5.5. The majority of note duration values used by each bertsolari belong to the zero-one seconds duration interval. We can observe that the distribution in male and female bertsolaris is very similar. Regarding the pitch distributions, the first observation is that female bertsolaris use higher pitch values than male

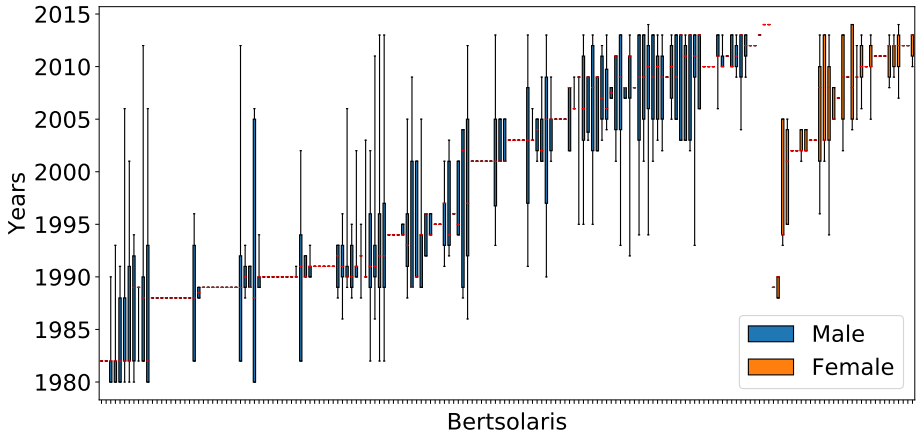


Figure 4.5.3: Distribution of recording years for each bertsolari in the Bertso database

bertsolaris. It can be also seen that the pitch values used inside each gender is not uniform: there are multiple male bertsolaris with pitch ranges comparable to female bertsolaris.

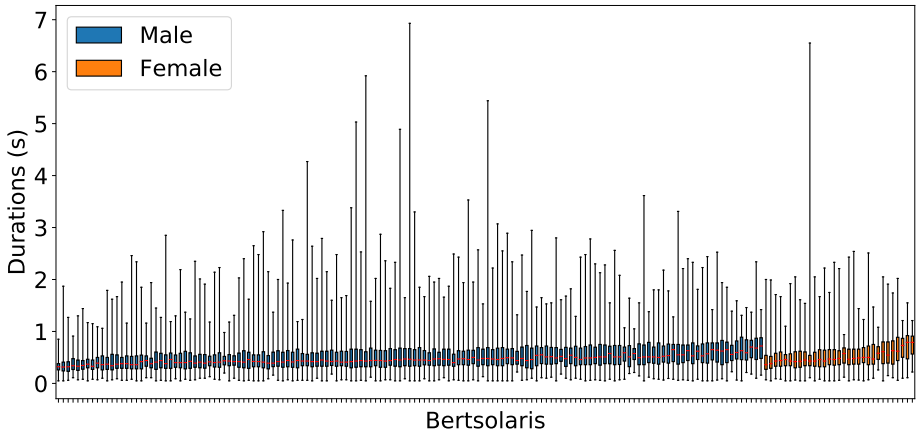


Figure 4.5.4: Distribution of note durations for each bertsolari in the Bertso database

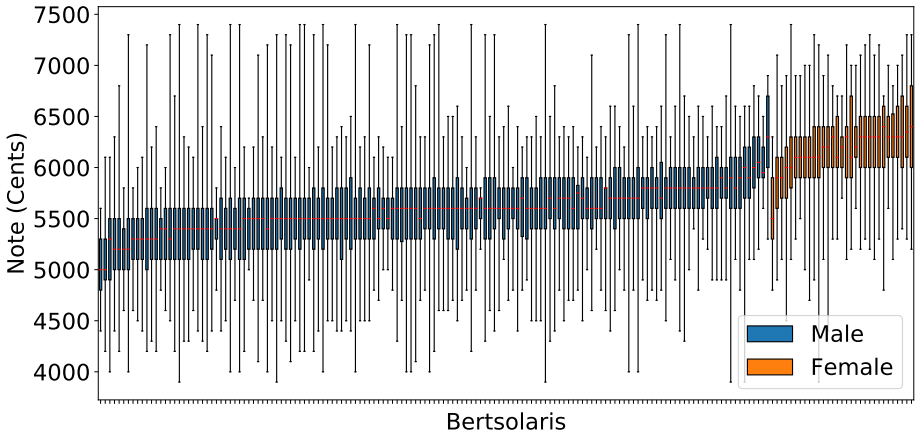


Figure 4.5.5: Distribution of note pitch values for each bertsolari in the Bertso database

In addition, we have analyzed the use of vibrato in the recordings. Figure 4.5.6 shows the percentage of notes where we have identified the use of vibrato in the recordings of each bertsolari. The majority of the bertsolaris make a small use of vibrato in their performances: multiple singers do not have even a single vibrato note detected. We can observe also that a small group of male singers use it in a more generalized way. There is no female bertsolari that reaches the maximum level of vibrato use observed in the male recordings. In Figure 4.5.7 we visualize in a boxplot the vibrato amplitude modulation in cents measured across all the vibrato segments realized by each bertsolari. The empty values represent the bertsolaris that have no vibrato segments detected. We can observe that some bertsolaris use higher modulation amplitudes than others and the total range of the value goes from 30 cents to 140 cents. The minimum value is easy to explain because 30 cents has been the minimum threshold used to detect it. We can also observe that female singers have been labeled with smaller modulation values. Observing these two figures we can say that vibrato is not a generalized singing technique applied in bertsolaritza and that it is harder to find it in female recordings.

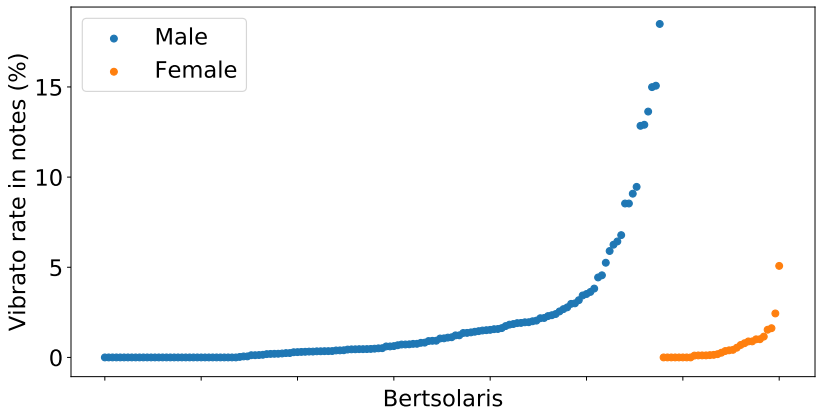


Figure 4.5.6: Percentage of notes with vibrato per bertsolari

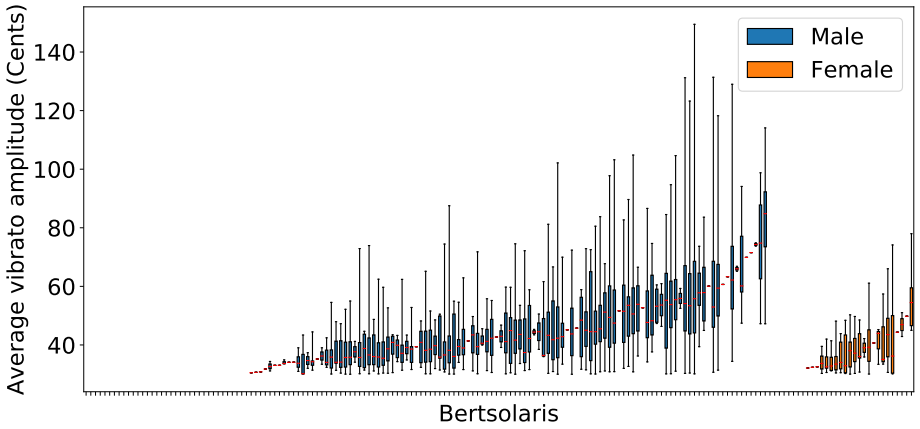


Figure 4.5.7: Average amplitude of vibrato per bertsolari

4.5.2 NUS database

We have applied the automatic melody alignment explained in Section 4.4.2 to the NUS database to obtain the pitch shift made by each singer in every music score utterance. We applied the pitch shift from each utterance to the original notes in the music scores to calculate an approximation of the sung notes in each utterance.

In this way, we obtained the approximation of note pitch values sung by the singers in the database. The note range used by each singer is shown in a boxplot in Figure 4.5.8.

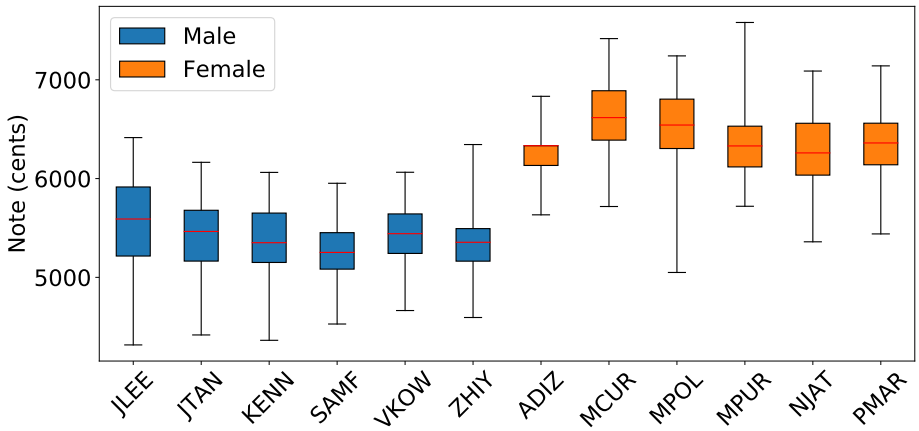


Figure 4.5.8: Range of note pitch values used by singers in NUS database

We can observe in the image that female singers made a higher pitch interpretation of the songs, as expected.

4.6 Chapter conclusion

In this chapter we have devised several methods to segment, align and annotate the recordings of the Bertso database. In the segmentation phase, we have proposed systems to separate singing voice from bertsolaris and speech from the hosts of the bertso session. The proposed segmentation system has obtained good results comparing to other speech/singing classification systems and has been published in [129] and [130]. In addition, an utterance segmentation system has been designed with good results in our database. For phoneme segmentation, we have applied a common triphone HMM system as the base system, but added three novel characteristics to the system to improve alignments. One addition is the use of

multiple bertsolaris in the training data to create more stable models. We also compared alignment methods using different phoneme groupings: single phonemes, notes and words. Finally, a postprocessing of the alignment has been designed using novelty features over MFCCs. The improvement of the alignment with the added methods has been proved and the final results are good considering that we have no prior information about note durations.

For the musical labeling, we designed different strategies to automatically label the whole database. We considered the problem of musical labeling knowing the original music score of the recordings and the problem of musical labeling without music score information. We also considered phenomena like portamento and vibrato in the note annotation systems. We have achieved good results and developed new tools for the automatic analysis of bertsolaritza.

Using the proposed automatic annotation systems, we annotated the whole Bertso database creating an annotated singing voice database that is bigger than many state of the art singing voice databases. This database can be used to train singing voice recognition and synthesis models and to perform general singing style analysis. The automatic segmentation, alignment and labeling systems are prepared to annotate more bertso recordings that are generated every year and compiled by Xenpelar documentazio zentroa.

5

Synthesis system

In this chapter we will use the database labeled in Chapter 4 to train different singing synthesis systems and evaluate the results. First, the general structure of the synthesis systems developed is presented in Section 5.1. Then, in Section 5.2 we explain the adaptation techniques we used in note pitch and durations to obtain more flexibility and better quality in our synthesis models. The process proposed for the reconstruction of the vibrato is explained in Section 5.3. Next, the difficulties of synthesizing any music score with a specific singer model is addressed and we propose a solution to the problem in Section 5.4. After that, the preprocessing and preparation of the linguistic labels, musical labels and acoustic parameters for statistical prediction training is explained in Section 5.5. Then, the particularities of the HMM and DNN-based synthesis systems proposed are defined in Sections 5.6 and 5.7 respectively. After explaining the systems we evaluate and compare them with objective and subjective tests in Section 5.8. We finalize the chapter

with the main conclusions in Section 5.9.

5.1 General architecture of the proposed singing synthesis system

We have created three singing voice synthesis systems to evaluate the suitability of different synthesis technologies for generating bertsolaritza signals. Two of them are HMM-based synthesis systems and the remaining one is a DNN-based synthesis system. Each system has unique characteristics but all of them share the main structure and multiple modules. In this section we explain the general architecture shared by all the developed systems. The particularities of each system will be explained with detail in their respective sections (Sections 5.6 and 5.7).

The general architecture is represented in Figure 5.1.1. It is is divided into two parts: the training phase and the synthesis phase. In the training phase labeled recordings of the databases are used to create models that can predict acoustic features from the labels. In the synthesis phase music scores that were not seen in the training phase are used to predict acoustic features and generate synthetic singing voice signals.

5.1.1 Training phase

In the training phase, the labels defined in Chapter 4 for the Bertso database are converted to phoneme-based context-dependent labels and adapted to accommodate them to the special needs of each synthesis system. The label adaptation module is different in HMM-based and DNN-based systems. Also the acoustic parameter extraction and preparation is different because it depends on what techniques are used by the synthesis system. The basic acoustic parameter extraction extracts three different acoustic parameters with Ahocoder [43]: MCEP, f_0 and MVF. The pitch curve is further processed to obtain a normalized version and to calculate parameters to encode vibrato when necessary. Delta and delta-delta parameters are

calculated for all the parameters. Finally, the phonemes with context-dependent information and the prepared parameters are used to train a statistical model. The three systems use separated models for the phoneme duration prediction and the frame level acoustic parameter prediction.

The labels defined in Chapter 4 are also used to obtain the distribution of notes corresponding to each singer, so that the new music scores that will be synthesized can be converted to the range of the singer before generating the synthesis labels.

5.1.2 Synthesis phase

In the synthesis phase, music scores that were not in the training material are converted to context-dependent phoneme labels and these labels are used to predict acoustic parameters.

The music scores to be synthesized are first adapted to the range of the modeled singer and the adjusted tempo. After that, the adapted music scores are separated into utterances. Then each utterance is converted to a phoneme sequence with context-dependent labels. The context-dependent labels are prepared in a different way for each developed synthesis system. These context-dependent labels are used to predict the duration of each phoneme applying the corresponding statistical duration model. The context-dependent labels completed with duration information are used as input for the acoustic parameter model to predict the frame level acoustic parameters. The predicted delta and delta-delta features are used in the Maximum Likelihood Parameter Generation (MLPG) algorithm to generate smoothed features. The pitch value is denormalized and vibrato is reconstructed from the predicted modulation features if needed. In all the systems, after the MLPG and pitch postprocessing modules we obtain the three acoustic features that will be used to build the synthetic waveform applying Ahodecoder: f_0, MVF and MCEP.

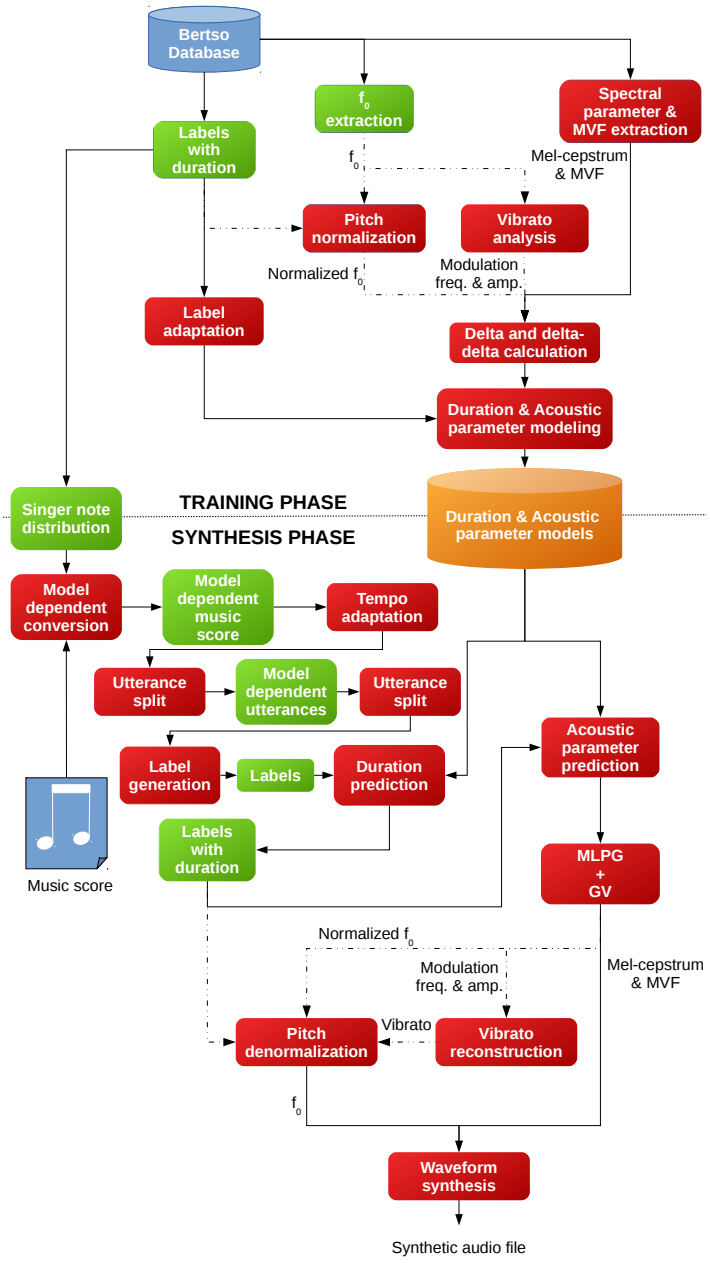


Figure 5.1.1: Generic statistical singing synthesis system

5.1.3 Summary of systems built

As we have already commented, we have developed three systems in total: two HMM-based systems and a DNN-based system. In Figure 5.1.1 we have represented the modules that are not shared by all the systems with a discontinuous line. These modules correspond to two techniques that we have introduced to improve the quality of the singing voice generation: pitch normalization (explained in Section 5.2.1) and vibrato reconstruction (explained in Section 5.3). We also proposed a technique to control the tempo of synthesized music score, explained in Section 5.2.2. To test the effect of these techniques we have used them differently in each of the three synthesis systems built. Table 5.1.1 shows what improvement technique is applied in each of the singing synthesis systems we have created.

Technique	**Synthesis systems**		
	HMM	HMM-pitch normalization	DNN
Pitch normalization	No	Yes	Yes
Vibrato reconstruction	No	No	Yes
Tempo adaptation	Yes	Yes	Yes

Table 5.1.1: Use of the proposed modeling improving techniques in each of the built singing synthesis systems

5.2 Techniques for gaining flexibility in the synthesis

A statistical singing voice synthesis system has to model the relationship between the information in a music score and the singing voice. In statistical synthesis systems, the quality of the synthetic voice depends on the data used to train the statistical models. The data have to be diverse enough to cover all contextual factors of the elements we want to synthesize afterwards. The number of contextual factors is very large in singing synthesis because of the complexity of the music scores, e.g., key, tempo, lyrics, duration... Nevertheless, we explained in Section 4.4.3 that as

we do not have the original scores of the recordings and it is highly time consuming to obtain these scores, we simplified the musical representation of these recordings to be able to obtain them automatically from the sung signal. For each note we only take into account the representation of the pitch in cents and the duration representation expressed in units of 64th. We obtained a precise representation of the notes but we have to train flexible models with these data. We have to be able to synthesize the lyrics in multiple levels of duration and pitch dimensions with limited training data. We have devised two techniques to obtain flexible and good quality synthesis models. The first is pitch normalization and adapts note pitch values. The second technique is called tempo adaptation and transforms the note duration to the reference tempo of our labeling. Both techniques are explained in the next sections.

5.2.1 Pitch normalization

As we explained in Section 4.4.3.1, the notes sung in our recordings are not adjusted to the standard A_4 tuning that we want to define for our synthesis system. A solution to this problem could be to label all the stable pitch segments as the nearest note in the A_4 tuned scale, but this would create high variance models for the pitch of each note. We propose a pitch normalization method to model only the variation of the pitch around the sung notes. This pitch normalization procedure contributes to the reduction of the pitch variance inside note values and to the improvement of the tuning of the excitation feature.

In the f_0 normalization module, a base melody is generated using the phonemes with musical labels obtained in Chapter 4. We subtract this base melody from the f_0 curve to obtain the deviation of the singer from the base melody and train the statistical models to learn the relation between the contextual labels and this pitch deviation. At synthesis time, the base melody is derived from the music score and the predicted phoneme duration values. Then, this base is added to the f_0 deviation predicted by the acoustic model.

We propose to obtain the base melody taking into account the boundaries of the

phonemes in the notes. A staggered curve is not suitable because the natural pitch curve is not stable when a transition between different notes is produced. Based on the observation of the dynamics of natural pitch curves, we have defined two kinds of pitch transitions between notes, depending on the kind of phoneme where the transition takes place (see Figure 5.2.1). As we can see in the figure, two types of transitions have been considered: transitions contiguous to unvoiced phonemes and transitions contiguous to voiced phonemes. In the unvoiced phoneme transitions, we define the transition area as the segment including all contiguous unvoiced phonemes. In this area we define a staggered pitch transition between the notes that is positioned in the middle of the transition area. We took this decision because we observed that unvoiced phonemes create a pitch boundary between notes and we do not want to distort the pitch in transitions. In the voiced phoneme transitions, the transition area is defined around the phoneme boundaries of the notes. A 25 ms space is taken at both sides of the phoneme boundary and spline interpolation is used to create a smooth transition between pitch levels corresponding to the two notes involved in the transition. As explained in Section 2.1.1, multiple studies have proposed to set the voiced note onset transition in the beginning of the first vowel in the note, but we preferred to use the first voiced phoneme. We have took this decision because of the high complexity and the context dependency of voiced transitions. We designed the voiced rule based transition to interfere with the smallest possible number of frames in the natural transition and to create a smooth reconstruction. A more complex rule-based voiced transition model would need a deeper analysis of bertsolaritza singing voice.

Expressions 5.1 and 5.2 define the value of the base melody in the transitions with voiced and unvoiced phonemes respectively.

$$M_b(x) = \begin{cases} N_a, & \text{if } x < T_a. \\ f(x), & \text{if } T_a \leq x \leq T_b. \\ N_b, & \text{if } T_b < x. \end{cases} \tag{5.1}$$

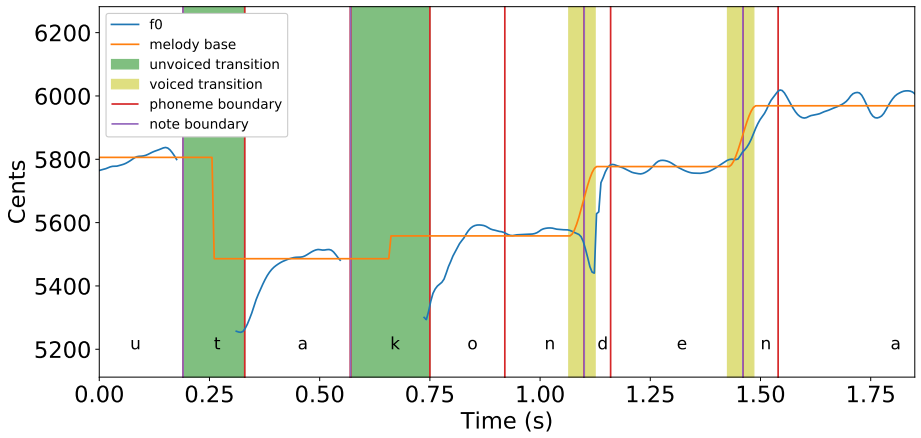


Figure 5.2.1: Transition model of the base melody

$$M_b(x) = \begin{cases} N_a, & \text{if } T_a < x \leq (T_b - T_a)/2 + T_a. \\ N_b, & \text{if } (T_b - T_a)/2 + T_a < x < T_b. \end{cases} \tag{5.2}$$

Where N_a and N_b are the pitch values of the previous and next notes in the transition, T_a and T_b are the time positions of the beginning and end of the transition area we have defined and $f(x)$ is a cubic spline interpolation between N_a and N_b values.

An example of the result of the base melody subtraction can be seen in Figure 5.2.2. We can see that before and after the voiced transition areas located at seconds 1.10 and 1.50 we obtain a f_0 with a mean value close to zero. This modulation around the zero value represents the natural modulation of the signal around the note pitch, known as fine fluctuations. In the unvoiced transitions located around 0.25 and 0.6 s, we can observe that the normalized f_0 before the transition is around the zero value and the onset of the note after the transition has a deviation until becoming stable. These are melody dynamics of note onset and offsets. The big negative spike observed in the transition at second 1.10 is the microprosody generated by the /d/ phoneme. Our statistical models will learn these natural variations of f_0 around a base melody and we will be able to reconstruct any melody afterwards

using a new base melody.

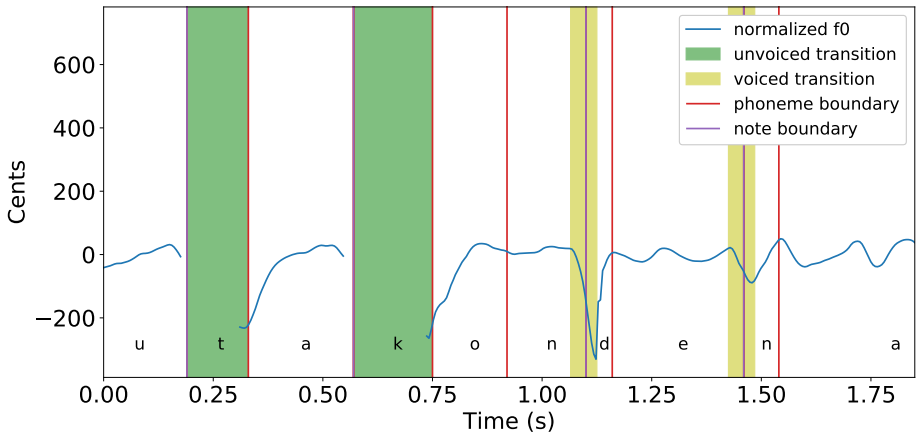


Figure 5.2.2: Transitions in the normalized f_0

5.2.2 Tempo adaptation

The duration labels we created for the Bertso database have a unique tempo reference. Using a unique tempo labels to train a singing voice model may create problems when trying to synthesize musical scores with different tempos, but we defined a conversion system to be applied before synthesis to convert the scores to the tempo of our model. Each note is converted to 64th units using tempo conversion according to Equation 5.3.

$$u = \lceil \frac{d}{D} \rceil \tag{5.3}$$

Where u is the number of 64th units of the optimal tempo (100) in the note, d is the duration of the note calculated with the tempo of the score to be synthesized and D is the duration of the 64th symbol in the optimal tempo which in our definitive system equals 37.5 ms.

As we used this technique in all the systems, we have not evaluated the result of using it.

5.3 Vibrato reconstruction

We explained in Section 4.4.4 how we created a frame level characterization of the vibrato modulation in the f_0. Considering that we have frame level information of the frequency and amplitude of the vibrato modulation, we can train any singing voice synthesis system to predict the vibrato in bertsolaritza. In this synthesis section we explain how we regenerate the vibrato once our statistical synthesis system have predicted the frequency and amplitude of the vibrato.

With the frame level information of amplitude and frequency modulation we create a sinusoid representing this modulation with the formula

$$v(t) = A(t)sin(2\pi f(t)t) \tag{5.4}$$

where $A(t)$ is the predicted amplitude modulation signal and $f(t)$ is the predicted frequency modulation signal. The resulting signal is added to the base melody defined in Section 5.2.1. We do not model the phase of the vibrato signal because we think the condition for the perception of good vibrato is phase continuity and not keeping the natural values of it. The result of using Equation 5.4 can create unnatural modulation signals if fast and big frequency changes occur in the $f(t)$ variable. But we observed that the predicted $f(t)$ create semi-continuous phase signals. Vibrato frequency modulation signal is limited between 5 and 8 Hz and has been calculated in previously smoothed f_0 curves. This creates slowly varying curves that produce no abrupt frequency changes in the reconstruction. Also, the value of the phase must be 0 at the edges to avoid abrupt changes when we add it to the synthesized f_0. The 0 value in the boundaries is a condition that is not naturally filled in the reconstruction and, therefore, we defined a method to force it.

We used a *sin* function in the reconstruction with no initial phase to force the first value in the signal to be 0. After the creation of the vibrato signal, we observe the position of the last zero-crossing point and delete the signal from this point till the end. The process can be seen in Figure 5.3.1.

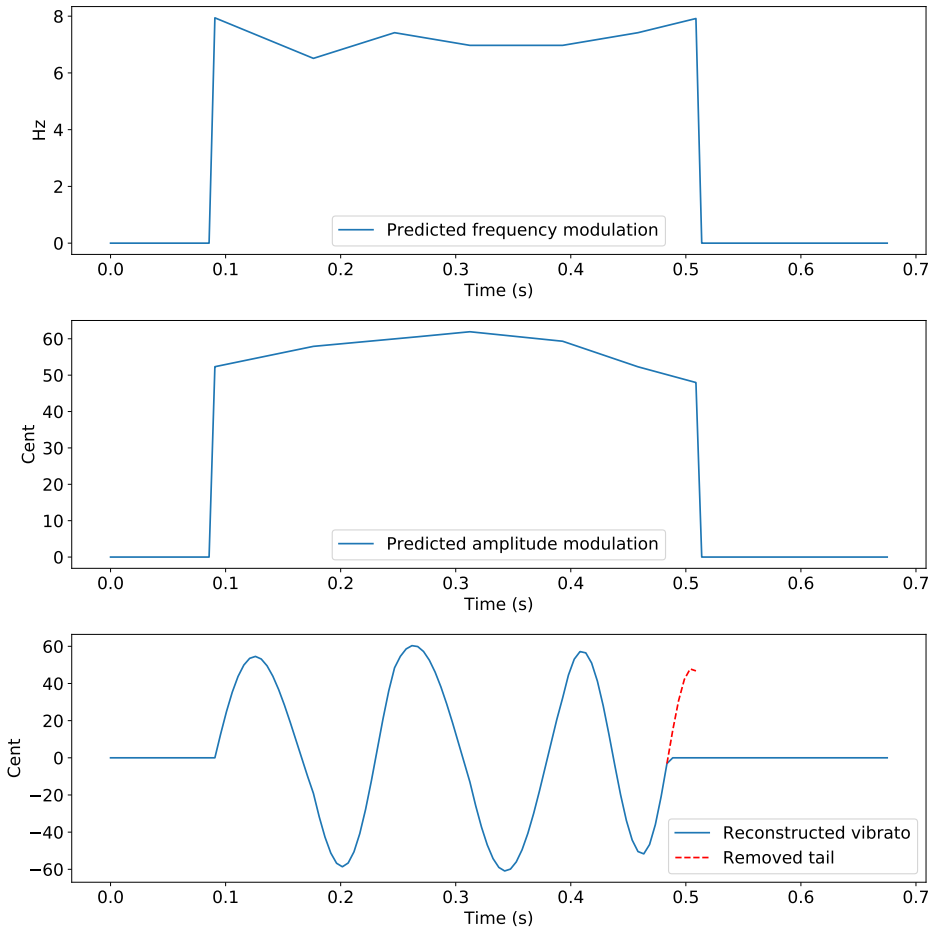


Figure 5.3.1: Vibrato reconstruction with phase constraints

5.4 Model dependent conversion of music scores

One of the main differences between singing voice and speech is the pitch range of each singer. Theoretically, any singer that knows how to read can read properly almost any existing text written in a certain language; conversely, this does not happen in singing voice. Not any music score can be sung by any singer because

of physical limitations. This happens because in speech the absolute values of the pitch values used in the intonation curve are not important to read the text correctly. The pitch conditions in speech are related with the prosody, and prosody conditions are related to relative differences of pitch values inside the sentence. In singing voice, a music score defines explicitly the pitch in specific phonemes in order to be sung correctly. The human range of pitch values is limited by physiological conditions as explained in Section 2.1 and therefore a singer can sing melodies within his or her pitch range. This is why the choruses have different type of singers to cover different 'Tessitura'. Nevertheless, a melody can be shifted to lower the higher pitch values and still be easily recognized because of the ability of humans to detect relative difference patterns between notes. When a music score for singing has been written to be part of a bigger part with multiple voices and instruments, it would be a bad idea to shift the pitch values, because it would break the harmonicity designed for the whole composition in the first place. But in bertsolaritza the singers sing mostly a cappella and the pitch level of the melody is closely related with the physiological limits of the bertsolaris. This is why we defined a method to adapt any music score to a singing voice model, artificially mimicking the process of adaptation that is naturally performed by any bertsolari.

In Section 4.4.3.2 we described the method proposed to align different distributions using a transformation function with n independent variables. We used the method to set a global tempo in the Bertso database. Here we will apply the same method, but in this case we will consider the pitch distribution of the recordings used to train the model and the pitch distribution of the music score. We use Expression 4.17, but in this case P_A is the PDF of the pitch values of all the recordings used to train the model, s_i is the ith note of the music score and N is the total number of notes in the music score. The solution of the distribution alignment problem comes by maximizing the shift in semitones that we have to apply to the music score to be sung in an optimal way by the singing model. The representation of this process can be seen in Figure 5.4.1.

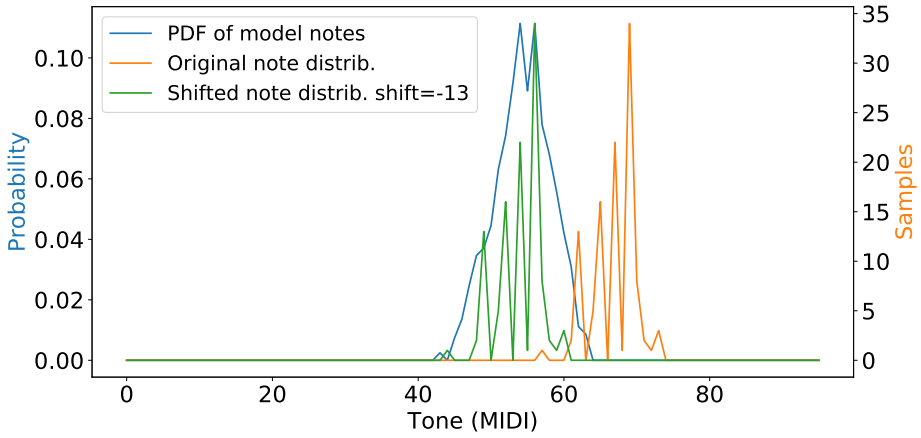


Figure 5.4.1: Model dependent score adaptation

We can observe in the figure that the original music score contains notes out of the range of the notes in this specific singer recordings. The problem is that this issue cannot be fixed recording songs with a wider range for the singer because each singer has a limited natural range. The interpretation of these notes from this singer does not ”exist” or the singer would interpret them in a poor quality. The music score has been shifted 13 semitones in order to obtain notes that are within the vocal range of the singer.

5.5 Data preparation

Bertso database has been segmented and annotated and the resulting dataset has been explained in Section 4.5.1. The database has a big size and irregular recording times and music scores per singer. This is why we have to define what recordings, contextual factors and features we are going to use to model the singing voice.

5.5.1 Selection and preparation of the recordings

To train the models we have used the database defined in Section 4.5. In the labeled database, we have recordings of 176 singers but all the singers have not the same amount of recordings. We decided to define a total recording time threshold to determine which singers to use to generate singing voice models. Considering previous works in statistical singing synthesis [123] [109][63][64], 70 minutes is a common threshold to create singing voice models. We have 12 bertsolaris that fill the condition of having a minimum of 70 minutes of recording time. The corresponding singing IDs and respective recording times are show in Table 5.5.1.

Singer ID	Recording time (min)	Genre
0030b	94.77	M
0045b	155.45	M
0051b	89.65	M
0054b	86.90	M
0087b	101.45	M
0108b	128.35	M
0111b	79.00	M
0113b	134.07	F
0115b	93.17	M
0125b	88.54	M
0126b	74.82	M
0151b	127.27	M

Table 5.5.1: Recording duration per singer with more than 70 minute recording

Having selected the singers, we also discarded some recordings from some singers, in order to eliminate recordings with labeling errors. As we know, the f_0 calculation algorithm as well as the note detection algorithm can produce errors. For examples, harmonics can be labeled instead of the fundamental. In order to minimize the number of recordings with labeling errors, we have defined a fixed interval of 20 semitones where the vocal range of the singer must be contained. The value of 20 semitones was chosen by simple observation. For each singer, the

PDF of the singer pitch values was obtained, and the 20 semitones interval was centered to maximize the area in that interval. Figure 5.5.1 shows how the optimal 20 semitone range is selected for bertsolari 0030b.

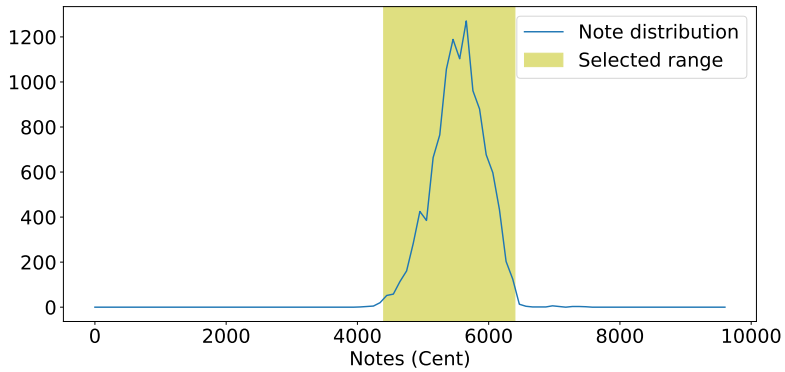


Figure 5.5.1: Note distribution and range selection in the recordings of bertsolari 0030b

Any recording with notes outside the selected range was left out from the final selection. After the range filtering, the available recording duration of the singers is downsized, as can be seen in Table 5.5.2.

Singer ID	Recording time (min)	Genre
0030b	90.98	M
0045b	147.32	M
0051b	85.78	M
0054b	83.14	M
0087b	98.54	M
0108b	125.68	M
0111b	74.27	M
0113b	134.00	F
0115b	88.75	M
0125b	87.43	M
0126b	73.19	M
0151b	123.21	M

Table 5.5.2: Recording duration per singer after range limitation

The distribution of recording years, note durations, note pitch values, percentage of notes with vibrato and average vibrato amplitude of each singer can be seen in boxplot format in Figures 5.5.2, 5.5.3, 5.5.4, 5.5.5 and 5.5.6 respectively.

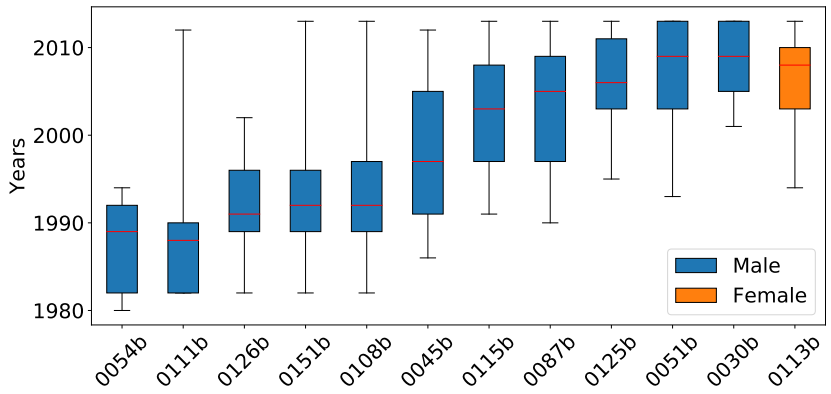


Figure 5.5.2: Recording year distribution per bertsolari in the selected data

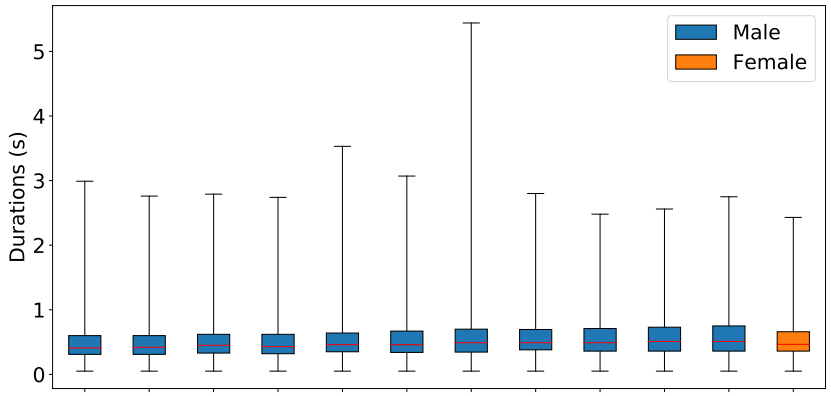


Figure 5.5.3: Note duration distribution per bertsolari in the selected data

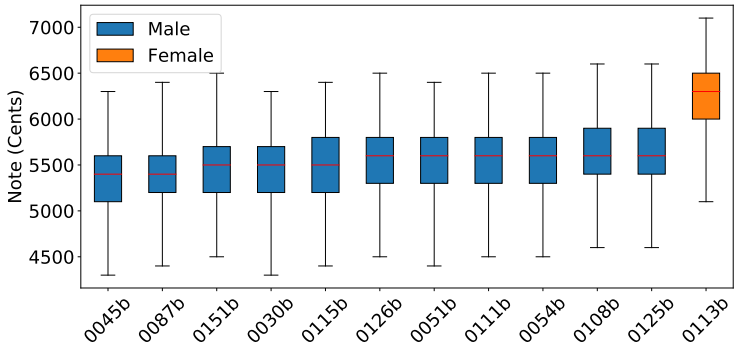


Figure 5.5.4: Note pitch distribution per bertsolari in the selected data

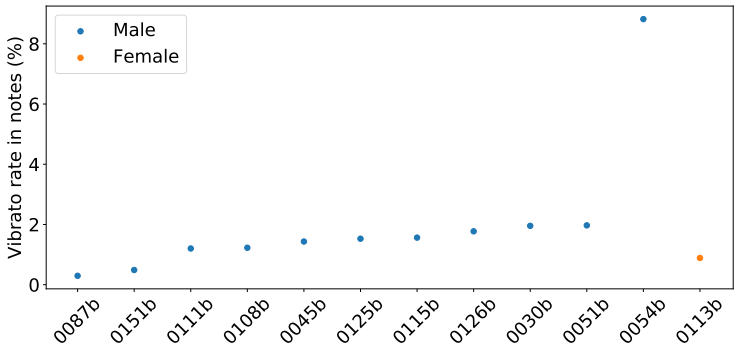


Figure 5.5.5: Percentage of notes with vibrato in the selected data

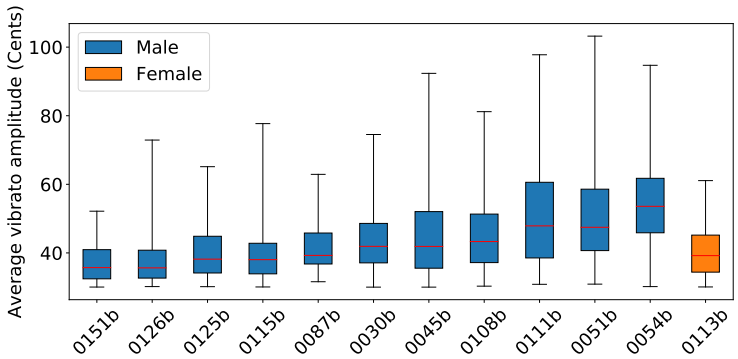


Figure 5.5.6: Average vibrato amplitude in the selected data

We split the utterances of each bertsolari into three different sets: train, validation and test. For the test block, we selected 10 utterances in a random way. We split 4% of the utterances for validation purposes and 96% as train set. The final duration of each block for each singer can be seen in Table 5.5.3.

Singer ID	Train recording time	Validation recording time	Test recording time
0030b	84.46	5.85	0.66
0045b	137.09	9.52	0.71
0051b	79.47	5.55	0.77
0054b	77.20	5.21	0.73
0087b	91.47	6.40	0.67
0108b	116.96	8.02	0.71
0111b	68.89	4.63	0.75
0113b	124.27	8.97	0.76
0115b	82.06	5.90	0.79
0125b	80.87	5.67	0.88
0126b	67.59	4.83	0.77
0151b	114.59	7.86	0.77

Table 5.5.3: Recording duration (min) per singer in train, validation and test sets

5.5.2 Preparation of the contextual labels

For each annotated bertso recording we created its music score. These music scores have no measure information because we do not have original music scores of the recordings and measures are hard to predict. The music scores we created are separated in utterances, have information about the phonemes corresponding to each note and each note has pitch and duration information. Using these music scores, contextual information added to each phoneme is listed in the following list.

- **Phonemes**: Including the current phoneme, the identity of the two previous and two next phonemes are used. The total number of possible phonemes is 37 including the silence.
- **Phoneme subgroup**: Additional information is added to each phoneme to indicate the phonetic subgroups to which it belongs (like plosives, fricatives

etc.). We defined the phoneme subgroups of the current phoneme, two previous phonemes and two next phonemes. The total number of possible subgroups is 51.

- **Note pitch**: The previous, current and next note pitch values. In HMM-based systems the pitch is represented as a MIDI number between 38 and 74 and in DNN-based system with an integer number in cents between 3800 and 7400.
- **Pitch distance**: The distance in semitones to the pitch of the previous note and to the next notes. The values of the distance is an integer between 0 and 30 MIDIs in the HMM-based systems and an integer between 0 and 3000 cents in DNN-based.
- **Duration**: The previous, current and next note durations are used expressed as multiples of 64ths with a tempo of 100 per quarter note.
- **Phonemes in note**: Number of phonemes in the current note.
- **Position inside note**: The position of the phoneme in the note starting from the beginning and also from the end of the note.
- **Notes in utterance**: Number of notes in the utterance.
- **Note position**: The position of the current note in the utterance starting from the beginning and also from the end of the utterance.
- **Closest silence distance**: The distance in notes from the current note to the closest previous and next silences.

The combination of all the labels sums up to 277 context-features added to each phoneme.

5.5.3 Acoustic features

The acoustic features we used in our synthesis systems are the ones provided by Ahocoder, defined in Section 3.2.5. As defined in Section 4.4.1, we had set the minimum and maximum values of f_0 in 75 and 580 Hz respectively. We used 40 MCEPs and all the analysis have been made to audio files with 16000 Hz sampling frequency.

To train the DNN-based system, apart from the mentioned spectral parameters, the two parameters representing the vibrato defined in Section 4.4.4 are also used in the training.

The delta and delta-delta coefficients have been calculated and modeled by the synthesis systems to apply MLPG algorithms after the parameter prediction. The delta features of any feature are calculated using Expression 5.5

$$\Delta[i] = -0.5x[i-1] + 0.5x[i+1] \tag{5.5}$$

where x is the feature. The delta-delta is calculated using Expression 5.6

$$\Delta^2[i] = \Delta[i-1] - 2\Delta[i] + \Delta[i+1] \tag{5.6}$$

5.6 HMM-based synthesis

5.6.1 System structure

The HMM-based system has the same structure explained in Section 5.2.1. As we explained in that section, the only difference between the developed systems are the label adaptation module and the parameter modeling and prediction blocks. In this system, the label preparation block rounds the continuous space of the musical notes to the closest ideal note using the A_4 tuning system. This discretization reduces the possible note values and therefore reduces the complexity of the model. At synthesis time, this model will only synthesize ideal notes in the A_4 tuned scale

and we think that creating a wider range labels would be a sub-optimal solution.

The acoustic parameters of phonemes with different context labels are modeled using the Equation 5.7

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} p(\hat{\boldsymbol{O}}|W, \boldsymbol{\lambda}) \tag{5.7}$$

where $\boldsymbol{\lambda}$ is the trained multistream HMM-GMM model, $\hat{\boldsymbol{O}}$ is the set of the acoustic parameters, W is the set of labels with contextual factors and λ is the initial HMM-GMM structure. In the model, Baum Welch equation is used to optimize the mixtures for each contextual factor. This optimization is combined with a tree-based context clustering to reduce redundant states and model uncovered states in the training data. The MDL criterion have been used to control the size of the decision trees in the clustering. The overall training is summarized in Figure 5.6.1.

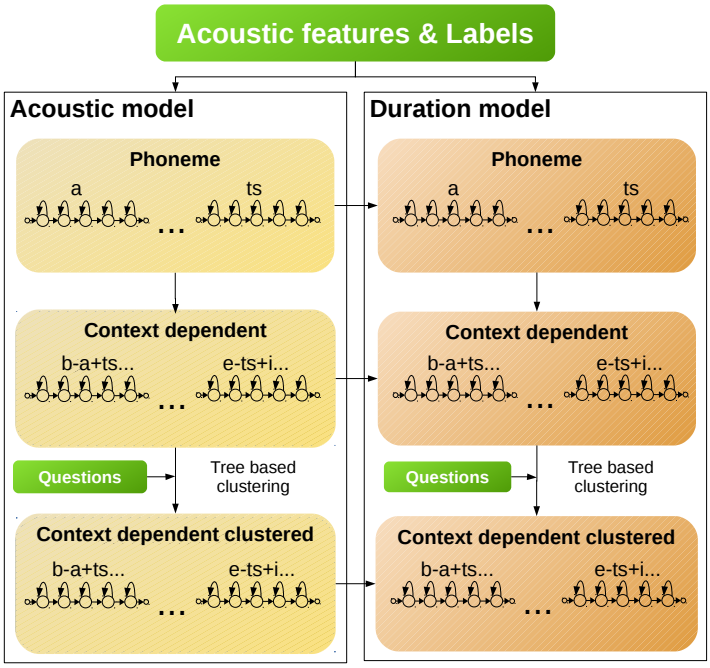


Figure 5.6.1: Training of the acoustic and duration models

The MCEP parameters, MVF and the respective delta features are continuous parameters and they are modeled by common HMM models. In contrast the f_0 parameter only exists for voiced frames. To model this discontinuity in f_0 and respective delta feature values, a Multi-Space Probability Distribution (MSD) HMM model is used. We have not used the vibrato parameter in this system. At synthesis time, before the f_0 denormalization process, we apply the MLPG algorithm to all acoustic parameters using the predicted variances and delta and delta-delta features [153]. Global Variance (GV) is also trained and applied in each utterances of synthesis to avoid oversmoothing [151].

5.6.2 Label preparation

Using the music scores created in Chapter 4 for the Bertso database, we have created the context labels defined in Section 5.5.2. The note pitch values determined by our automatic labeling algorithm define a continuous note space in the cent scale. Unfortunately, in HMM parameter modeling the contextual features are defined in a discrete mode to use decision trees in model clustering. This is why we discretized the annotated continuous pitch values to the nearest pure pitch when creating the synthesis labels. Duration information is also defined as a continuous space of multiples of 64th note durations. We discretized this parameter to the nearest integer. We also used this discrete logic to neutralize the pitch and duration information from unvoiced phonemes. The unvoiced phonemes have no f_0 value and therefore predicting pitch information for them is pointless. The note duration effect in unvoiced phoneme duration has also been analyzed in Section 4.4.3.2 and showed that a small correlation exists between the duration of unvoiced phonemes and the duration of the notes .

5.6.3 Trained models

The HMM models have been trained using the material defined in Section 5.5.1. Two models have been created per singer and the training material used has been

the combination of train and validation sets. The two different models trained for each singer have a different f_0 feature:

- **HMM**: This model has been trained with the labels defined in Section 5.6.2 and with the Ahocoder features defined in Section 5.5.3.
- **HMM-pitch normalization**: This model has been trained with the labels defined in Section 5.6.2 and with the Ahocoder features defined in Section 5.5.3, but we applied the note normalization procedure defined in Section 5.2.1. This normalization allows us to use continuous pitch values to normalize f_0 although contextual labels are still discrete.

In the models, we made 5 training iterations in each of context free monophone, full context phoneme and clustered full context phases. We modeled the MCEP coefficients, MVF parameter and duration values in the same way in both models and we did not include the vibrato features. In all the parameters and systems we used 1035 questions and a MDL penalty of 1.0 to apply the clustering. The trained models are evaluated in Section 5.8.

5.7 DNN-based synthesis

5.7.1 System structure

The Neural Network TTS system has the same structure explained in Section 5.2.1. As we explained in that section, we only change the label adaptation and parameter modeling and prediction blocks. In this system, the label preparation module converts every phoneme with its contextual factors into a numerical vector using questions about the values in these contextual factors. In duration modeling, each phoneme is vectorized and used as input feature to predict the duration of each phoneme. In a second acoustic phase, the vector of each phoneme is repeated for all the frames occupied by the phoneme and positional encoding inside the frame

is added to each frame vector to predict acoustic parameters. Because of the importance of the f_0 in singing voice, we decided to model pitch related parameters and spectral related parameters with different networks. In the case of neural networks, the discretization of the notes is not needed because we can represent a pitch in the continuous space. The system scheme can be seen in Figure 5.7.1.

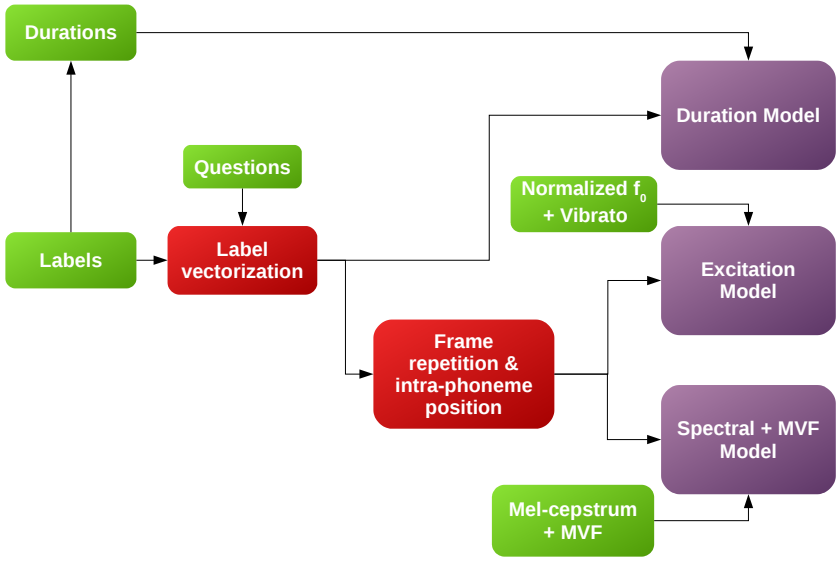


Figure 5.7.1: Training of the spectral features, excitation features and duration models in the DNN-based synthesis system

The neural networks used have 512 neurons in each layer and are visualized in Figure 5.7.2. They have two fully connected layers followed by two BLSTM layers, all of them using tanh as activation function.

In the duration prediction the silences from the utterances are included, because the duration of the silences must be predicted. In the case of acoustic feature prediction networks the frames annotated as silence are removed from the utterances. The MCEP parameters and the MVF are continuous parameters and are modeled without any preprocessing. The f_0 feature only exists in voiced frames and therefore we interpolated the unvoiced part of the signal and included a binary signal in the predicted features with the voiced/unvoiced information in each frame. In

this model we used pitch normalization and vibrato reconstruction techniques. As many frames have no modulation information, we used the same method used in the f_0 but with vibrato/no vibrato decision. The amplitude and frequency features are interpolated in the no vibrato frames and the vibrato presence information in encoded in an extra binary feature. All non binary features are normalized with zero-mean and unit-variance and the network is trained to predict normalized features.

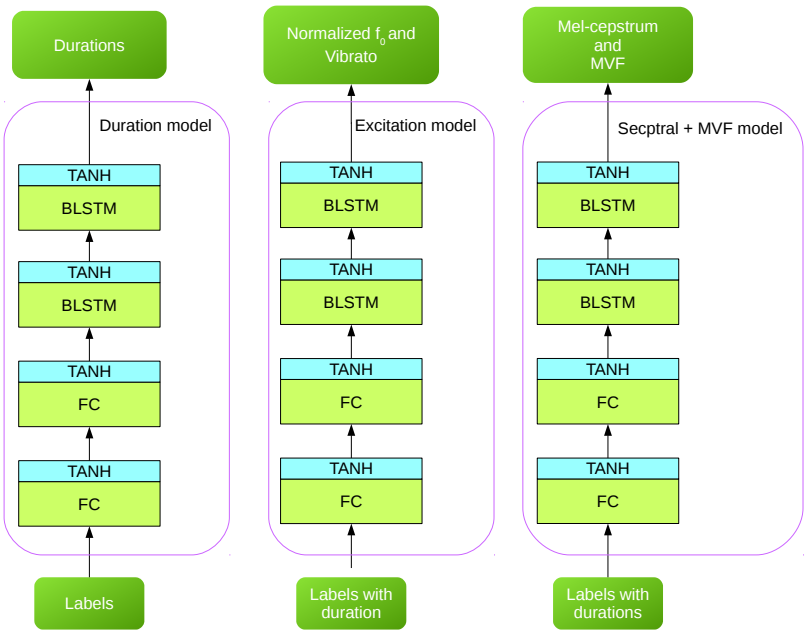


Figure 5.7.2: Neural Networks for the different models

In the feature prediction phase, the duration of each phoneme and silence is predicted with the duration network. The silences are removed from the predicted labels saving their position in the utterance. Parameters in non-silence phonemes are predicted and we apply the mean and variance denormalization and MLPG algorithm like in HMM-based synthesis systems. The MLPG algorithm in neural network systems is applied in a different way because no variances are predicted

for each feature. In Neural Networks the predicted output features from the DNN are set as mean vectors and pre-computed variances of the features from all training data as covariance matrices [103]. When all non-silence parameters are predicted, the silences are created in the saved positions repeating a silence frame that have been randomly selected from a recording from the Bertso database. When all frames in the utterance are generated the f_0 denormalization process is applied and the vibrato reconstruction is also added to the f_0 signal.

5.7.2 Label preparation

Using the music scores created in Chapter 4 for the Bertso database, we created the context labels defined in Section 5.5.2. In DNN-based synthesis systems the context labels are converted to vectors and used as input of the networks. We represented the phonemes with one-hot encoding and the phoneme subgroups with binary encoding because of their non-continuous nature, but the rest of the features are represented numerically with integers. DNN-based systems do not use the questions to cluster features like HMM-based systems. This allows to use big numerical ranges without increasing the number of questions. This is why we decided to use a higher precision in the definition of the note pitch and pitch distance in this system than the one used in the HMM-based systems developed. Using the continuous note pitch labels that we automatically created, we rounded the pitch value in cents to the nearest integer. This provides a higher precision compared to the MIDI number used in HMM-based systems. In the frame level prediction networks, 3 coarse-coded features and the duration of the phoneme have been added to each frame [166]. The coarse-coding feature adds information about the position of each frame inside the phoneme to the input vector. The positioning information is obtained by sampling three gaussian functions of 0.4 standard deviation that are stretched or compressed to adapt to the duration of each phoneme. This stretching or compressing process allows representing the same relative position values inside each phoneme with similar values of the coarse features. The equation of a gaussian with 0.4 standard deviation and 0 mean value is

$$g(x) = \frac{1}{0.4\sqrt{2\pi}} e^{-(x)^2/2(0.4)^2} \tag{5.8}$$

where x is the time axis. The three coarse features are obtained from this gaussian function applying different delays and stretching factors, as indicated in the next list:

- **First coarse feature**: The mean of the gaussian $(g(0))$ is positioned at the start of the phoneme and the gaussian is expanded so that $g(1)$ coincides with the end of the phoneme.
- **Second coarse feature**: The mean of the gaussian is positioned at the center of the phoneme and the gaussian is stretched so that $g(-0.5)$ corresponds to the beginning of the phoneme and $g(0.5)$ corresponds to the end of the phoneme.
- **Third coarse feature**: The mean of the gaussian is positioned at the end of the phoneme and the function is spread so that $g(-1)$ coincides with the start of the phoneme.

The representation of the gaussians can be seen in Figure 5.7.3.

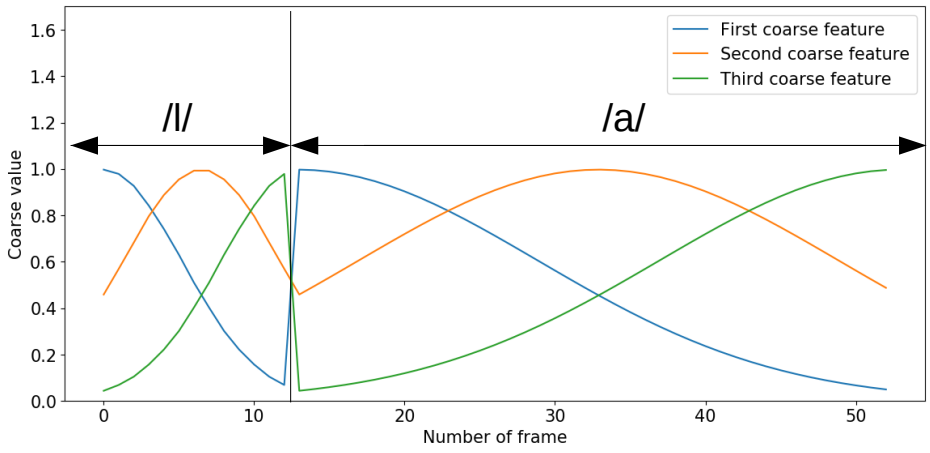


Figure 5.7.3: Coarse-coding in contiguous $/l/$ and $/a/$ phonemes

5.7.3 Trained models

The DNN models have been trained using the material defined in Section 5.5.1. One model has been created per singer. The training set has been used to train the networks and the validation set has been used to stop neural network iterations using the validation loss as reference. The numerical features of the input and the all the output features have been normalized with 0 value in mean and 1 in variance for the neural training. We used the Minimum Square Error (MSE) to optimize the networks with Adam optimizer and 0.001 learning rate. In the duration model, we have used a batch size of 512, input labels of size 456 and output labels of dimension 1. In the model of the MCEP and MVF the batch size have been 200, the input labels size 460 and the output features of dimension 123. In the model of the f_0 and the Vibrato the batch size have been 200, the input labels size 460 and the output features of dimension 11. In these models the f_0 normalization process explained in Section 5.2.1 has been used. The characteristics of the training of each model can be seen in Table 5.7.1.

	Durations		**MCEP + MVF**		**f_0 + vibrato**	
Singer ID	**Epochs**	**Validation MSE**	**Epochs**	**Validation MSE**	**Epochs**	**Validation MSE**
0030b	124	0.0560	38	0.5591	11	0.5964
0045b	82	0.0577	26	0.5591	20	0.4339
0051b	71	0.0747	42	0.6065	9	0.3499
0054b	139	0.0489	36	0.6219	13	0.5983
0087b	139	0.0527	29	0.5900	24	0.2944
0108b	103	0.0430	29	0.5912	23	0.5912
0111b	133	0.0375	34	0.5336	8	0.5915
0113b	86	0.0563	40	0.5902	8	0.5408
0115b	91	0.0900	42	0.5233	19	0.6628
0125b	26	0.3222	49	0.4239	25	0.4773
0126b	152	0.0379	28	0.6104	23	0.6104
0151b	94	0.0277	28	0.5121	11	0.6507

Table 5.7.1: Characteristics of DNN training with validation early stopping

The trained models are evaluated in Section 5.8.

5.8 Evaluation

5.8.1 Objective evaluation

For objective evaluation, we compared the parameters of the 10 utterances reserved for testing purposes synthesized with the values predicted by the different models built for the selected singers. The evaluation of phoneme durations and all acoustic parameters have been performed in different synthesis utterances in each bertsolari. The predicted phoneme durations have been evaluated using the original durations. However, the acoustic parameter evaluations have been calculated in utterances synthesized forcing the duration to that of the original utterances. In this way the need for alignment between the synthetic signal and the reference is eliminated. The error formulas used for each parameter are shown in equations 5.9 to 5.13:

- **Duration distortion**: The average duration distortion is calculated with the expression

$$D_e = \frac{1}{N}\sum_{t=1}^{N}\frac{|d_t - \hat{d}_t|}{d_t} \tag{5.9}$$

  where N is the total number of phonemes, d is the duration of the original phoneme and $\hat{d}$ is the predicted duration for the phoneme. The distortion calculated per singer can be seen in Figure 5.8.1. We visualized the mean and standard deviation of each system in all phonemes.

- **MCD**: The Mel Cepstral Distortion (MCD) is calculated using the expression

$$MCD = \frac{10}{\ln 10}\frac{1}{T}\sum_{t=1}^{T}\sqrt{\sum_{i=1}^{40}2(mc(t,i) - mc_{pred}(t,i))^2} \tag{5.10}$$

where T is the total number of frames, *mc* is the matrix with the original mel cesptral parameter frames and mc_{pred} is the matrix with the predicted mel cesptral parameter frames. The Mel Cepstral Distortion (MCD) calculated per singer can be seen in Figure 5.8.2. We visualized the mean and standard deviation per utterance in the test set.

- **MVF RMSE**: The MVF is a voice parameter with high range of values because it varies from the maximum periodic frequency in voiced frames to 0 in unvoiced frames. To reduce the distortion value of this feature we evaluated it only in the frames that are voiced in the reference and predicted features. The MVF error in these voiced frames is calculated using the expression of Root Minimum Square Error (RMSE).

$$MVF_e = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(mvf(t_{voiced}) - mvf_{pred}(t_{voiced}))^2} \tag{5.11}$$

where T is the total number of frames, *mvf* is the original MVF value for this frame and mvf_{pred} is the predicted value. The MVF error calculated per singer can be seen in Figure 5.8.3. We visualized the mean and standard deviation per utterance in the test set.

- **V/UV error**: The Voice/Unvoiced (V/UV) error is calculated using the expression

$$VUV_e = \frac{FP + FN}{T} \tag{5.12}$$

where *FP* is the total number of unvoiced frames classified as voiced, *FN* is the total number of voiced frames classified as unvoiced and *T* is the total number of frames. The V/UV error calculated per singer can be seen in Figure 5.8.4. We visualized the mean and standard deviation per utterance in the test set.

- f_0 **RMSE**: The f_0 error is calculated using the expression of RMSE

$$f_{0e} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(f_0(t_{voiced}) - f_{0pred}(t_{voiced}))^2} \tag{5.13}$$

where T is the total number of frames, f_0 is the original f_0 value and f_{0pred} is the predicted f_0. We only evaluated the frames that are voiced in both the original and predicted f_0. The f_0 error calculated per singer can be seen in Figure 5.8.5. We visualized the mean and standard deviation per utterance in the test set.

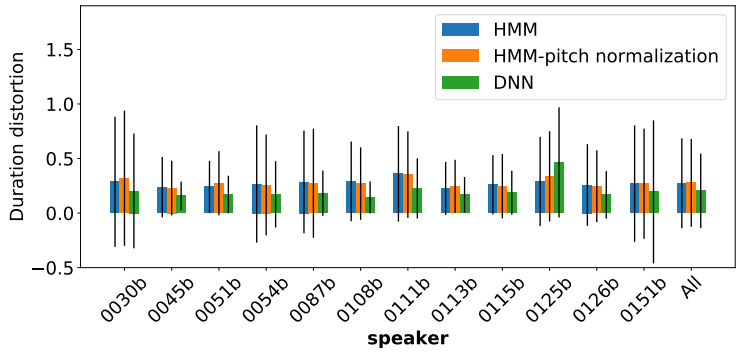


Figure 5.8.1: Duration distortion per singer

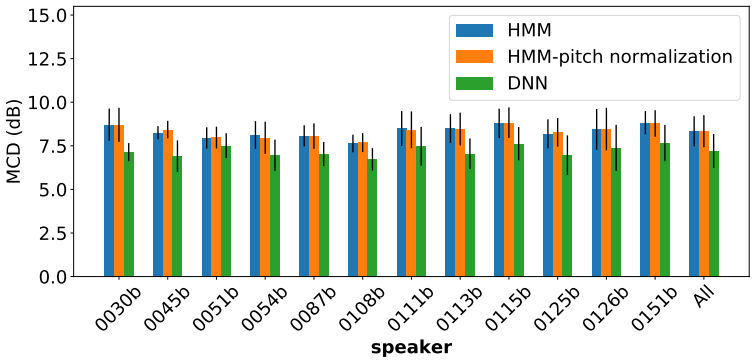


Figure 5.8.2: MCD per singer

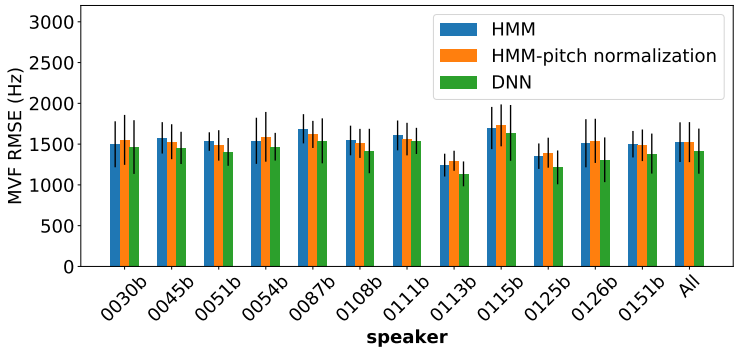


Figure 5.8.3: MVF RMSE per singer

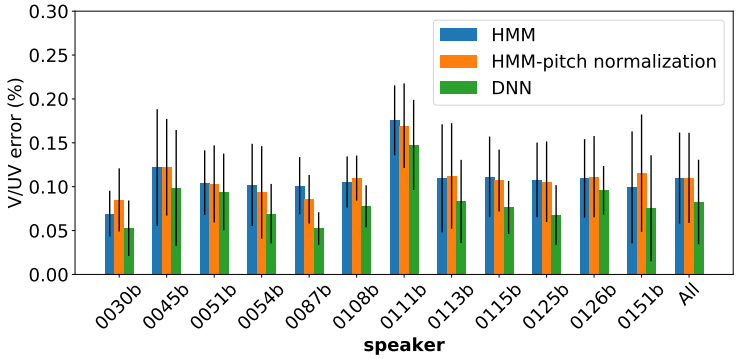


Figure 5.8.4: V/UV error ratio per singer

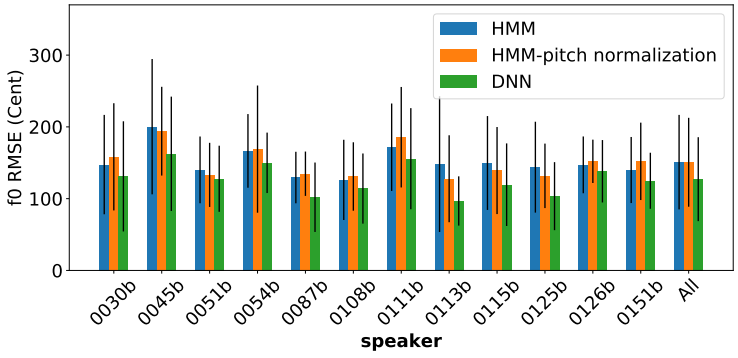


Figure 5.8.5: f_0 RMSE per singer

We can observe in the figures that the DNNs obtain the best result in every category and singer. With these results we can say that Neural Networks seem a better parameter prediction mechanism compared to the HMM-based systems. We are aware that the objective measures considered have not always correlation with the subjective evaluation. Nevertheless, these results are a clear proof of the advantage of Neural Networks in objective evaluation. If we consider the two HMM-based systems, we can observe that none of them offers a clear advantage over the other. To analyze this in a deeper way we wrote in Table 5.8.1 the number of times that each of the HMM-based systems obtains better mean results that the other HMM-based system.

Measures	**HMM**	**HMM-pitch normalization**
Duration distortion	4	8
MCD	6	6
f_0 RMSE	7	5
MVF RMSE	6	6
V/UV error	6	6

Table 5.8.1: Best position of each HMM based synthesis system for different measures

In the table we can observe there is not a clear advantage of a system in any of the objective measures considered. In the figures we also visualized the distortion of the parameter considering the utterances of all the bertsolaris together. In this evaluation the neural networks obtain the best result in all evaluation parameters and the HMM-based systems are very close. If we compare the HMM-based systems, the system with no f_0 normalization obtain better mean results in duration distortion, MCD, VUV error and MVF RMSE. The system with no f_0 normalization obtain better mean distortion in the majority of the parameters but the margin is very small as it can be seen in the figures.

5.8.2 Subjective evaluation

In the subjective evaluation, we selected multiple original and synthesized recordings and tested their quality and similarity using Mean Opinion Score (MOS) and Similarity Score method. This kind of test is completed by people and therefore the number of synthetic signals that can be evaluated must be limited. This is why we made a selection of singers for this test. We decided that taking into account the time limitation of the test, we could evaluate five singers. Looking at the figures in Section 5.5, we concluded that it was interesting to include singers from a diversity of year span, of different genres and level of vibrato. Regarding genre, we only have one possibility for female singers and therefore singer 0113b has been included in the test. About the year span, we consider that the best option is to test modern singers and singers with a high year span in recordings so that we can analyze session and year variability. Considering this, we included 0045b and 0108b who have a big year span and 0030b as a modern bertsolari. We consider the female bertsolari as a modern one too. For the case of vibrato, we selected bertsolari 0054b as he is a clear exception who often uses the vibrato reconstruction technique with a clear amplitude. About the note pitch and note duration variation, there are no clear different classes to analyze, only the higher pitch of the female bertsolari that it is already included in the selection. After this analysis, bertsolaris 0030b, 0045b, 0054b, 0108b and 0113b have been selected to be included in the subjective test.

The recordings of each of the singers have not been pre-designed to comprise all the musical spectrum that each of the singers can cover. We also have to take into account that each of the singers has very different songs in lyrics and melody among their recordings. Taking into account this variability, we decided to test sung utterances that have the highest probability considering the note distribution in the training set of each bertsolari. With this selection we tackle the problem explained in Section 5.4. The recordings of all the bertsolaris are not comparable and therefore we are interested in evaluating the ceiling quality of the systems trained with these recordings. If we synthesize songs with a higher note pitch range than

the one seen during training, the model will likely synthesize singing voice with errors in the notes with lack of data. The presence of these errors can be predicted before the synthesis by observing the range of the training data. The objective of this test is to imitate the results of models with pre-designed recordings and with a large musical range cover for each singer. Taking this into account, we used the method explained in Section 5.4 to adapt the most used 30 music scores in the melody database to the range of each bertsolari. After adapting the scores, we selected the 20 bertso utterances from these scores that have the best note probability in the note PDFs of the training data of each bertsolari.

30 people, most of them with experience in Speech and NLP research areas, evaluated the quality and similarity of 4 types of recordings in a 5 point MOS scale:

- **Original**: Original recordings that were included to have an upper reference to compare the results of our systems.
- **HMM**: HMM-based system modeling Mel-Generalized Cepstrum (MGC), MVF, f_0 in cents and respective delta and delta-delta features.
- **HMM-pitch normalization**: HMM-based system modeling MGC, MVF and the normalized f_0 in cents and respective delta and delta-delta features.
- **DNN**: DNN-based system modeling MGC, MVF and the normalized f_0 in cents and respective delta and delta-delta features.

The participants used the MOS evaluation system to evaluate the Quality and Similarity Score in two separate tests. The mean scores and the 95 % confidence interval values for the Quality and the Similarity scores are show in Tables 5.8.2 and 5.8.3 respectively.

Singer ID	Original	HMM	HMM-pitch normalization	DNN
0030b	4.85 ± 0.08	2.72 ± 0.22	2.89 ± 0.22	2.88 ± 0.22
0045b	4.75 ± 0.11	2.61 ± 0.22	2.69 ± 0.22	2.95 ± 0.18
0054b	4.77 ± 0.11	2.71 ± 0.22	2.68 ± 0.23	2.44 ± 0.19
0108b	4.76 ± 0.10	2.39 ± 0.22	2.41 ± 0.22	2.43 ± 0.19
0113b	4.87 ± 0.07	2.68 ± 0.23	3.02 ± 0.26	3.01 ± 0.20
Total	4.80 ± 0.04	2.62 ± 0.10	2.74 ± 0.10	2.74 ± 0.09

Table 5.8.2: Results of the subjective evaluation of quality

Singer ID	Original	HMM	HMM-pitch normalization	DNN
0030b	4.34 ± 0.26	2.64 ± 0.31	3.24 ± 0.29	2.72 ± 0.28
0045b	3.31 ± 0.37	2.79 ± 0.30	3.18 ± 0.26	2.70 ± 0.27
0054b	3.65 ± 0.35	2.43 ± 0.32	2.57 ± 0.30	2.20 ± 0.27
0108b	3.93 ± 0.35	2.81 ± 0.32	2.88 ± 0.28	2.34 ± 0.30
0113b	4.37 ± 0.29	3.33 ± 0.28	3.72 ± 0.30	3.10 ± 0.34
Total	3.92 ± 0.15	2.80 ± 0.14	3.12 ± 0.13	2.62 ± 0.13

Table 5.8.3: Results of the subjective evaluation of similarity with the original voice

5.8.2.1 Analysis of the subjective results

In the results of quality scores, we can observe that the original recordings obtain a score very close to the maximum score of 5, which indicates that evaluators did not score the stimuli randomly and that these recordings may indeed be an upper reference for the subjective measures. The results for the natural recordings in the similarity scores are not as close to the maximum as happened with quality MOS. The singers with more modern recordings, 0030b and 0113b, have higher scores of similarity for the original recordings than the rest. We interpret this as a clear sign that the bertsolaris with a bigger span of recording years have recordings with more diversity in quality, environment and bertsolari age. All these factors affected the perception of voice similarity in the test. We have empirically observed that

older recordings tend to have worse quality than the modern ones. With respect to the age, considering big time spans in the recordings creates non uniform data, because of the change of the voice with the age.

The methods that include f_0 normalization get the best results for quality MOS except for singer 0054b. There are two motives for these good results of the methods with f_0 normalization. First, the representation in musical labels of the notes is limited to the notes tuned with the 440 Hz A_4 and we know that the singers in the Bertso database do not sing all the notes with this tuning. This dispersion in the tuning creates problems of definition when we model the f_0 directly, creating high variance definition of each note. The variance between all the notes defined as the same semitone is too high to generate stable notes and therefore, unnatural notes are created. Second, created notes can be stable but not correctly tuned and the listeners can detect this lack of tuning. Another clear conclusion is that the singers with modern recordings, 0030b and 0113b, obtain better scores for almost all systems. Among the singers with modern recordings, the singer with the most recordings available to do the training obtains the best score (0113b). It can be said that obtaining better quality results with tuned notes, more uniform recordings, more quality in the recordings and more data to train are expected and intuitive results. If we compare the HMM-based and DNN-based systems that use pitch normalization, we obtain mixed results. In the modern bertsolaris, the scores are almost tied with a small advantage for HMMs; in the three remaining singers with older recordings, DNNs obtain a high advantage for 0045b, HMMs obtain a high advantage for 0054b and results for 0108b are very close in both systems, with a small advantage for the DNNs. The singer with vibrato, 0054b, obtains overall the worst results comparing with the rest of bertsolaris and there is no clear advantage in scores between the DNN-based system that uses the vibrato reconstruction and the rest of the systems. We observed that the vibrato is not present in many of the test recordings. This is because these utterances have not many long notes that are needed for the vibrato to be present. In addition, we have to consider that all the recordings of this bertsolari are from the 80s and we have seen that the bigger the amount of modern recordings, the higher is the subjective perceived quality.

In the similarity scores, the HMM-based systems clearly capture in a better way the identity of the singing voices. The HMM-pitch normalization system obtains the best result in every bertsolari and the DNN system obtains the worst result in every bertsolari except for 0030b. The scores are related to the number of modern recordings as in the quality evaluation and the note tuning does not seem to be an important aspect for singer recognition. This means that although the melody is not properly interpreted in the musical aspect, the listeners are able to detect the identity of the bertsolari.

5.9 Chapter conclusion

In this chapter we used GMM-HMM and Neural Networks to create three different statistical parametric singing voice synthesis systems using automatically annotated bertsolaritza recordings. We defined a method to synthesize tuned singing voices using f_0 normalization and defined a phoneme dependent f_0 normalization considering the characteristics of singing f_0. We included the vibrato reconstruction technique in neural network synthesis systems that automatically includes vibrato in the notes in the voices that uses it in the recordings. A musical score adaptation method have been defined to make it easier the data preparation for multiple bertsolaris with very different singing pitch ranges. The objective results shown a better parameter prediction have been obtained by Neural networks. The subjective evaluation of the quality ceiling showed preference for f_0 normalization and no clear difference between the HMM and Neural Networks that use it. We evaluated also the similarity of the synthesized voice with the original recordings obtaining better results in HMM systems. The similarity evaluation demonstrated also that bertsolaris with bigger time span in the recordings are harder recognize because the quality of sessions and the age of the bertsolaris vary more in their recordings.

6 Conclusions

In this chapter we summarize the contributions made in this work and comment on possible future developments and improvements. In Section 6.1 we explain all the contributions and publications of this work. Next, in Section 6.2 we define possible improvements and new research areas for the work made.

6.1 Contributions

In this thesis we have explored the possibility of using the bertso recordings compiled by Xenpelar Documentation Center for bertsolaritza singing voice synthesis. For this purpose, we have created diverse tools to label the bertso recordings at multiple levels of annotation. We have obtained a multi-singer singing voice synthesis database that is separated in utterances and is labeled at phonetic and musical level. We have used the database to obtain singing synthesis models of multiple bertso-

laris with different technologies. The automatic labeling systems and the singing voice synthesis systems have been evaluated with good results. In this section we briefly describe the contributions of the work and we list all the scientific publications derived from it.

6.1.1 Analysis of singing voice and bertsolaritza

We have studied the technologies used for analysis, synthesis, segmentation and labeling of singing voice, producing documentation and discussing the main advantages and drawbacks of each of them.

We also have made an in-depth description of the bertsolaritza art creating documentation defining its history, explaining its structure and addressing all the research made on this art. The documentation reveals the social relevance of this art in the Basque Country and the need of signal processing research to open new ways to analyze it.

6.1.2 Singing voice data collection

We have collected, documented and standardized all the available singing voice data. The bertso recordings of the Xenpelar Documentation Center have been characterized adapting data formats to make further analysis in audio files, music scores and transcriptions easier. The preparation of the Bertso database has been published in [128]. NUS and NITech databases have also been collected and described. Phonetic music scores have been added to the NUS database, obtaining the music scores of the songs and manually correcting differences in phonetic transcriptions and pronounced phonemes.

6.1.3 Automatic labeling of bertso recordings

We have created a pipeline with automatic labeling methods to prepare the bertso recordings from Xenpelar Documentation Center for singing voice synthesis. We

have devised several methods to obtain singing voice segments, and to perform utterance segmentation, phoneme segmentation and musical labeling.

The singing voice segmentation we have developed is a method that uses a novel note detection algorithm. The segmentation system has been published in [129] and [130]. The results of our methods have been compared with other systems obtaining better generalization for different databases and faster comnputation time.

We have proposed a multi-singer utterance segmentation system that detects silences that are not present in the transcriptions. The method uses different models for the phonemes depending on their position inside the word and applies singer adaptation to be able to segment recordings from new singers without retraining. This is an advantage as our method can be used in the future to segment new singers with no need of updating the model. For phoneme segmentation, we compared mono-singer systems with multi-singer systems that use singer adaptation. We also tested different phoneme modeling strategies considering the position of the phoneme in the word and syllable. The best results have been obtained in the multi-singer system with the syllable level phoneme models. We have also devised a novel phoneme boundary refinement method that uses audio novelty. We proved that the use of boundary refinement improves the alignment results both at a global level and also for every phoneme.

We have analyzed the coherence of the bertso recordings with the melodies and compared this coherence with other singing voice database to evaluate the results. Taking into account the results of this analysis, we have considered that bertso melodies are not suitable to be used to musically annotate the recordings. In consequence, we have devised methods to annotate scoreless singing voice recordings. We labeled the pitch of the notes using the obtained phoneme alignments and applying a novel method for note labeling. The proposed note labeling procedure includes a vibrato detection and characterization method that obtains frame-level information of the modulation. The duration of the notes has been labeled combining the note distributions of the bertso melodies and the real durations of the syllables. With this duration labeling system we have created a coherent labeling for the music scores of bertso melodies that allows flexibility in the tempo.

Multiple systems that predict the bertso melody of each bertso have been created. Although the correspondence between the labeled melody and the interpretation is not perfect, it have been proved that the classification of the recording in the most similar melody can obtain good results. The system with the best results uses our automatic musical labeling and the distance between note sequences to obtain the most similar melody.

6.1.4 Bertso database

After using all our labeling systems we obtained a singing voice database of bertsos with all the labels needed for singing synthesis. The multi-singer database has bigger size that any publicly available singing voice database with more than 49 hours of singing from more than 170 bertsolaris. We have characterized the distribution of pitch values, durations, number of utterances and vibrato use in it. We observed that there were more utterances and longer recording for males than females in the database. The use of vibrato is also higher in the male bertsolaris than in female bertsolaris.

6.1.5 Singing synthesis systems

We have built singing voice synthesis systems that use different techniques to improve the quality of the singing voice synthesis. We have included novel pitch normalization and vibrato reconstruction techniques. We have tested different uses of these techniques to evaluate the improvements obtained by them. In total two HMM-based synthesis systems (one applying f_0 normalization in the modeling and the other one without f_0 normalization) and a single DNN-based synthesis system have been created. We have developed an automatic method for singer adaptation of music scores to make the process of synthesizing any music score with synthesis models of different range singers easier.

6.1.6 Evaluation of singing synthesis systems

We have evaluated the synthesis systems with objective and subjective tests. In the objective test, DNN-based system obtained the best results in all the proposed measures and singers. The two HMM-based systems obtained worse results compared to DNN-based system, but there is no clear difference if we compare one against each other: they get similar results.

In the subjective evaluation, we evaluated the quality of the synthesis and the similarity of the synthetic voice with the original voice. In the quality test, the models that have been trained with more modern recordings and pitch normalization obtained the best results. The bertsolaris with bigger span in the recording years have more diversity in the characteristics and quality of the recordings and this has a damaging effect in the models. The pitch normalization procedure proposed helps to create relative pitch models that contribute to the synthesis of tuned melodies. Without the pitch normalization, the model of the pitch values has to deal with high variance values to model, resulting on unexpected melodies as result.

In the similarity evaluation results have shown that the HMM-based systems create synthetic singing voices that are easier to identify than those produced with DNN. The pitch normalization technique does not improve the singer recognition, this means that a good representation of the melody is not correlated with singer identification.

The vibrato reconstruction has not produced any clear improvement nor degradation in the results of any evaluation. The main reason for this is the automatic selection of the evaluation utterances. The vibrato is used exclusively in long notes and the utterances selected for evaluation have few long notes.

6.1.7 Publications

6.1.7.1 Journal publications

- **Sarasola, X.**, Navas, E., Tavarez, D., Serrano, L., Saratxaga, I. and Hernaez, I., 2019. Application of Pitch Derived Parameters to Speech and Monophonic Singing Classification. Applied Sciences, 9(15), p.3140.

6.1.7.2 Conference papers

- **Sarasola, X.**, Navas, E., Tavarez, D., Serrano, L. and Saratxaga, I., 2018. Speech and monophonic singing segmentation using pitch parameters. In IberSPEECH (pp. 147-151).

- Tavarez, D., **Sarasola, X.**, Alonso, A., Sanchez, J., Serrano, L., Navas, E. and Hernáez, I., 2017. Exploring Fusion Methods and Feature Space for the Classification of Paralinguistic Information. In INTERSPEECH (pp. 3517-3521).

- **Sarasola, X.**, Navas, E. eta Hernáez, I., 2017, Maiatza. Ahots kantatuaren sintesiaren tzapena bertsolaritzarako, Ikergazte 2017

- **Sarasola, X.**, Navas, E., Tavarez, D., Erro, D., Saratxaga, I. and Hernáez, I., 2016, May. A singing voice database in Basque for statistical singing synthesis of bertsolaritza. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016) (pp. 756-759).

- Erro, D., Alonso, A., Serrano, L., Tavarez, D., Odriozola, I., **Sarasola, X.**, del Blanco, E., Sánchez, J., Saratxaga, I., Navas, E. and Hernáez, I., 2016. ML Parameter Generation with a Reformulated MGE Training Criterion-Participation in the Voice Conversion Challenge 2016. In INTERSPEECH (pp. 1662-1666).

- del Blanco, E., Hernaez, I., Navas, E., **Sarasola, X.** and Erro, D., 2016. Bertsokantari: a TTS Based Singing Synthesis System. In INTERSPEECH (pp. 1240-1244).

6.1.7.3 Awards and distinctions

- Selected paper "Speech and monophonic singing segmentation using pitch parameters" in Iberspeech 2018 Conference to be part of "IberSPEECH 2018: Speech and Language Technologies for Iberian Languages" Special Issue in the Multidisciplinary Digital Publishing Institute Journal.

6.2 Future work

In this work we obtained a multi-singer singing voice database with musical labeling. In the speech generation area, multi-singer databases may be used for voice adaptation and voice conversion techniques. Voice adaptation methods use multi-singer data to create average voice models and then employ these average voice models to create synthesis models for one singer by applying adaptation algorithms. The amount of data required to create a synthesis model for a singer by adaptation is smaller than the one needed if we train it from zero. Voice adaptation systems with a good average model need less data for each singer to synthesize their voices, creating an advantage for the generation of new voices. In singing voice conversion systems two types of systems can be found: parallel and non parallel conversion. The Bertso database has no parallel data for any singer. Therefore it is impossible to work with parallel conversion methods. With this database, non-parallel conversion systems should be used.

Since 2016, the increasing availability of big speech databases has created new paradigms for statistical speech generation. Architectures based on neural networks like Tacotron and Wavenet have proven to create new ways to model speech signal and they have achieved better results than the previous systems. Similar DNN based architectures have been tested for singing voice but the resulting sub-

jective quality results are not in the level of speech. In this work we have created a relatively big singing database, but there are not enough recordings from each singer to train independent Tacotron or Wavenet systems. The only option to do experiments with these architectures is to adapt them or to use all the database to build multi-singer models using singer embeddings.

This work also asserted the importance of the f_0 for singing voice perception. The expressiveness and naturalness of the singing voice highly depends on f_0 and all the phenomena that happens on it. Creating new methods to model the f_0 curve after a deeper analysis of bertsolaritza would improve the final quality of the synthetic singing voice in a considerable way. Separating the f_0 in different elements (i.e. melody, transitions, vibrato and microprosody) and parametric reconstruction of these elements is the most used technique.

Acronyms

ABS Analysis-by-Synthesis. 32, 37

AST Automatic Singing Transcriptions. 43

BDB Bertsolaritzaren Datu-Basea. ix, xiii, 53–59, 61, 92

CGAN Conditional Generative Adversarial Network. 38

CMVN Cepstral Mean and Variance Normalization. 83

CNN Convolutional Neural Network. 38, 43

DFT Discrete Fourier Transform. 41, 82, 83, 87, 89

DNA Deoxyribonucleic acid. 67

DNN Deep Neural Network. iii, vi, xi, xvi, xx, 9, 10, 20, 21, 38, 65, 68, 101, 171, 172, 175, 189, 190, 193, 194, 196, 198, 203, 205–208, 212, 213, 215

DTW Dynamic Time Warping. 42, 118, 122, 157, 159

EpR Excitation plus Resonance. 32

FFE f_0 Frame Error. 114

FM Frequency Modulation. 30, 31

fMLLR Feature space Maximum Likelihood Linear Regression. 93, 95

FOF Fonction d'Onde Formantique. 32

FPE Fine Pitch Error. 114

GMM Gaussian Mixture Model. ix, x, xix, 41, 73–75, 82–84, 86, 87, 89, 101, 191, 208

GPE Gross Pitch Error. 114

GV Global Variance. 192

HMM Hidden Markov Model. iii, vi, ix–xi, xix, xxi, 3, 10, 20, 37, 38, 42–44, 65, 67, 68, 73–75, 84, 101, 105, 113, 138, 168, 171, 172, 175, 189–193, 195, 196, 203, 205–208, 212, 213

HSMM Hidden Semi-Markov Models. 43

HTK Hidden Markov Model Toolkit. 67, 101

IBM International Business Machines Corporation. 38

IDE Integrated Development Environment. 68

KTH Kungliga Tekniska Högskolan. 18, 30, 34

LDA Linear Discriminant Analysis. 93, 95

LPC Linear Predictive Coding. 31

LSTM Long Short-Term Memory. 20

MAP Maximum a Posteriori. 42

MCD Mel Cepstral Distortion. xvi, 199–201, 203

MCEP Mel-cepstral Coefficients. 68, 172, 173, 190, 192–194, 198

MDL Minimum Description Length. 191, 193

MFCC Mel Frequency Cepstral Coefficient. ix, 73–75, 82, 83, 86–89, 93, 101, 169

MGC Mel-Generalized Cepstrum. 205

MIDI Musical Instrument Digital Interface. 25, 55, 114, 130, 146, 189

MIR Music Information Retrieval. 43

MLLR Maximum Likelihood Linear Regression. 42

MLLT Maximum Likelihood Linear Transform. 93, 95

MLPG Maximum Likelihood Parameter Generation. 173, 190, 192, 195

MOS Mean Opinion Score. 204–207

MP3 MPEG-2 Audio Layer III. 55, 56

MSD Multi-Space Probability Distribution. 192

MSE Minimum Square Error. 198

MusicXML Music extensible Markup Language. 56

MUSSE Music and Singing Synthesis Equipment. 30

MVF Maximum Voiced Frequency. xvi, 68, 172, 173, 192–194, 198, 200, 202, 203, 205

NITech Nagoya Institute of Technology. xiii, 53, 65, 66, 68, 69, 124, 125, 210

NLP Natural Language Processing. 2, 205

NPSS Neural Parametric Singing Synthesizer. 38

NUS National University of Singapore. ix, xi, xiii, xiv, xvi, xix, 9, 53, 63–65, 68, 72, 76–79, 82, 85–88, 124–126, 162, 167, 168, 210

OLA Overlap-Add. 32, 37

PDF Probability Distribution Function. 149, 182, 205

RMSE Root Minimum Square Error. xvi, xvii, 200, 202, 203

SLM Sinusoidal Likeness Measure. 68

SPASM Singing Physical Articulatory Synthesis Model. 29

SPSS Statistical Parametric Synthesis Systems. 38

STFT Short-Time Fourier Transform. 32

STL Speech Transmission Laboratory. 34

SVM Support Vector Machine. 41, 42, 76

TIFF Tagged Image File Format. 55, 56

TTS Text-to-speech. 4, 193

VAD Voice Activity Detection. ix, x, xiii, xix, 40, 73–76, 79, 84, 96

VDE Voicing Decision Error. 114

WAV Waveform Audio File Format. 56

WGANSing Wasserstein Generative Adversarial Network Sing. 38, 39

References

[1] Manex Agirrezabal, Bertol Arrieta, Aitzol Astigarraga, and Mans Hulden. POS-tag based poetry generation with WordNet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 162–166, 2013.

[2] Ainhoa Aizpurua. Basque improvised sung poetry: the key to understand the linguistic rhythm. *Music, Language and Cognition*, 2017.

[3] Masato Akagi and Hironori Kitakaze. Perception of synthesized singing voices with fine fluctuations in their fundamental frequency contours. In *Proceedings of Sixth International Conference on Spoken Language Processing (ICSLP)*, pages 458–461, 2000.

[4] Masato Akagi, Mamoru Iwaki, and Tomoya Minakawa. Fundamental frequency fluctuation in continuous vowel utterance and its perception. In *Proceedings of Fifth International Conference on Spoken Language Processing (ICSLP)*, number 0027, 1998.

[5] Luc Ardaillon, Gilles Degottex, and Axel Roebel. A multi-layer f0 model for singing voice synthesis using a b-spline representation with intuitive controls. In *Proceedings of INTERSPEECH*, pages 3375–3379, 2015.

[6] Aitzol Astigarraga, Manex Agirrezabal, Elena Lazkano, Ekaitz Jauregi, and Basilio Sierra. Bertsobot: the first minstrel robot. In *Proceedings of 6th International Conference on Human System Interactions (HSI)*, pages 129–136, 2013.

[7] Aitzol Astigarraga, José María Martínez-Otzeta, Igor Rodriguez, Basilio Sierra, and Elena Lazkano. Emotional poetry generation. In *Proceedings of International Conference on Speech and Computer*, pages 332–342, 2017.

[8] Bishnu S. Atal and Suzanne L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50(2B):637–655, 1971.

[9] Guido Aversano, Anna Esposito, Antonietta M. Esposito, and Maria Marinaro. A new text-independent method for phoneme segmentation. In *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems (MWSCAS)*, volume 2, pages 516–519, 2001.

[10] Joxe Azurmendi. Bertsolaritzaren estudiorako. *Jakin*, 14(15):139–164, 1980.

[11] Onur Babacan, Thomas Drugman, Nicolas d'Alessandro, Nathalie Henrich, and Thierry Dutoit. A comparative study of pitch extraction algorithms on a large variety of singing sounds. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7815–7819, 2013.

[12] David J. Benson. *Music: A mathematical offering*. Cambridge University Press, 2008.

[13] Gunilla Berndtsson. The KTH rule system for singing synthesis. *Computer Music Journal*, 20(1):76–91, 1996.

[14] Hervé Bitteur. Audiveris, 2014. URL https://github.com/Audiveris/audiveris.

[15] Merlijn Blaauw and Jordi Bonada. A neural parametric singing synthesizer modeling timbre and expression from natural songs. *Applied Sciences*, 7 (12):1313, 2017.

[16] Merlijn Blaauw and Jordi Bonada. Sequence-to-sequence singing synthesis using the feed-forward transformer. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7229–7233. IEEE, 2020.

[17] Merlijn Blaauw, Jordi Bonada, and Ryunosuke Daido. Data efficient voice cloning for neural singing synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6840–6844. IEEE, 2019.

[18] Dawn A.A. Black, Ma Li, and Mi Tian. Automatic identification of emotional cues in Chinese opera singing. In *Proceedings of International Conference on Music Perception and Cognition and Conference for the Asian-Pacific Society for Cognitive Sciences of Music*, pages 250––255, 2014.

[19] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *Proceedings of the Institute of Phonetic Sciences*, volume 17, pages 97–110. Amsterdam, 1993.

[20] Jordi Bonada, Òscar Celma Herrada, Àlex Loscos, Jaume Ortolà, Xavier Serra, Yasuo Yoshioka, Hiraku Kayama, Yuji Hisaminato, and Hideki Kenmochi. Singing voice synthesis combining excitation plus resonance and sinusoidal plus residual models. In *Proceedings of the 2001 International Computer Music Conference*, 2001.

[21] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[22] Sandrine Brognaux and Thomas Drugman. HMM-based speech segmentation: Improvements of fully automatic approaches. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(1):5–15, 2016.

[23] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468, 2010.

[24] John P. Cater. *Electronically Speaking: Computer Speech Generation*. Sams Technical Publishing, 1983. ISBN 0672219476 (pbk.).

[25] Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez. WGANSing: A multi-voice singing voice synthesizer based on the Wasserstein-GAN. In *Proceedings of 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019.

[26] John M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the audio engineering society*, 21(7):526–534, 1973.

[27] John M. Chowning. Computer synthesis of the singing voice. In J. Sundberg, editor, *Sound generation in winds, strings, computers*, pages 4–13. Royal Swedish Academy of Music, Stockholm, Sweden, 1980.

[28] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[29] Perry R. Cook. Spasm, a real-time vocal tract physical model controller; and singer, the companion software synthesis system. *Computer Music Journal*, 17(1):30–44, 1993.

[30] Cristina de la Bandera, Ana M. Barbancho, Lorenzo J. Tardón, Simone Sammartino, and Isabel Barbancho. Humming method for content-based music information retrieval. In *Proceedings of 12th International conference on Music Information Retrieval (ISMIR)*, pages 49–54, 2011.

[31] Tom De Mulder, Jean-Pierre Martens, Micheline Lesaffre, Marc Leman, Bernard De Baets, and Hans De Meyer. Recent improvements of an auditory model based front-end for the transcription of vocal queries. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages iv–iv, 2004.

[32] Andrew Demetriou, Andreas Jansson, Aparna Kumar, and Rachel M Bittner. Vocals in music matter: the relevance of vocals in the minds of listeners. In *ISMIR*, pages 514–520, 2018.

[33] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[34] Diana Deutsch, Rachael Lapidis, and Trevor Henthorn. The speech-to-song illusion. *The Journal of the Acoustical Society of America*, 124(2471): 10–1121, 2008.

[35] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.

[36] Robert E. Donovan and Ellen M. Eide. The IBM trainable speech synthesis system. In *Proceedings of Fifth International Conference on Spoken Language Processing (ICSLP)*, number 0166, 1998.

[37] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang. The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and

speech. In *Proceedings of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–9, 2013.

[38] Homer Dudley and Thomas H. Tarnoczy. The speaking machine of Wolfgang von Kempelen. *The Journal of the Acoustical Society of America*, 22(2): 151–166, 1950.

[39] Homer Dudley, Richard R. Riesz, and Stanley S.A. Watkins. A synthetic speaker. *Journal of the Franklin Institute*, 227(6):739–764, 1939.

[40] Georgi Dzhambazov and Xavier Serra. Modeling of phoneme durations for alignment between polyphonic audio and lyrics. In *Proceedings of 12th Sound and Music Computing Conference*, pages 281–286, 2015.

[41] R EBU-Recommendation. Loudness normalisation and permitted maximum level of audio signals. 2011.

[42] Alexander John Ellis. On the musical scales of various nations. *Journal of the Society of arts*, 1885.

[43] Daniel Erro, Inaki Sainz, Eva Navas, and Inma Hernaez. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):184–194, 2013.

[44] Simone Falk and Tamara Rathcke. On the speech-to-song illusion: Evidence from german. In *Proceedings of Fifth International Conference on Speech Prosody*, 2010.

[45] James L. Flanagan. *Speech analysis synthesis and perception*, volume 3. Springer Science & Business Media, 2013.

[46] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 452–455, 2000.

[47] Gene A. Frantz, Kun-Shan Lin, and Kathleen M. Goudie. The application of a synthesis-by-rule system to singing. *IEEE Transactions on Consumer Electronics*, (3):257–262, 1982.

[48] Kotaro Fukui, Yuma Ishikawa, Eiji Shintaku, Masaaki Honda, and Atsuo Takanishi. Anthropomorphic talking robot based on human biomechanical structure. *Advances in Science and Technology*, 58:153–158, 2008.

[49] Joxerra Garzia Garmendia. *Gaur egungo bertsolarien baliabide poetiko-erretorikoak: marko teorikoa eta aplikazio didaktikoa*. PhD thesis, 1999.

[50] Joxerra Garzia Garmendia, Jon Sarasua, and Andoni Egaña. *The art of bertsolaritza: improvised Basque verse singing*. Bertsozale Elkartea, 2001.

[51] Joxerra Garzia. History of improvised bertsolaritza: A proposal. *Oral Tradition*, 22(2):77–115, 2007.

[52] Joxerra Garzia. Toward true diversity in frame of reference. *Oral Tradition*, 22(2):143–156, 2007.

[53] David Gerhard. Pitch-based acoustic feature analysis for the discrimination of speech and monophonic singing. *Canadian Acoustics*, 30(3):152–153, 2002.

[54] Izaro Goienetxea, José María Martínez-Otzeta, Basilio Sierra, and Inigo Mendialdua. Towards the use of similarity distances to music genre classification: a comparative study. *PloS one*, 13(2), 2018.

[55] Izaro Goienetxea, Iñigo Mendialdua, Igor Rodríguez, and Basilio Sierra. Statistics-based music generation approach considering both rhythm and melody coherence. *IEEE Access*, 7:183365–183382, 2019.

[56] Jean-Philippe Goldman. EasyAlign: an automatic phonetic alignment tool under Praat. In *Proceedings of INTERSPEECH*, pages 3233–3236, 2011.

[57] Emilia Gómez and Jordi Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2): 73–90, 2013.

[58] Rong Gong and Xavier Serra. Singing voice phoneme segmentation by hierarchically inferring syllable and phoneme onset positions. In *Proceedings of INTERSPEECH*, pages 716–720, 2018.

[59] Rong Gong, Philippe Cuvillier, Nicolas Obin, and Arshia Cont. Real-time audio-to-score alignment of singing voice based on melody and lyric information. In *Proceedings of INTERSPEECH*, pages 3312–3316, 2015.

[60] Juan Gorostiaga. *Antología de poesía popular vasca*. Biblioteca Vascongada de los Amigos del País, 1955.

[61] Thomas Hain and Philip C. Woodland. Segmentation and classification of broadcast news audio. In *Proceedings of Fifth International Conference on Spoken Language Processing (ICSLP)*, number 0851, 1998.

[62] Inma Hernaez, Eva Navas, Juan Luis Murugarren, and Borja Etxebarria. Description of the AhoTTS system for the Basque language. In *Proceedings of 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[63] Yukiya Hono, Shumma Murata, Kazuhiro Nakamura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Recent development of the DNN-based singing voice synthesis system—SinSy. In *Proceedings of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1003–1009, 2018.

[64] Yukiya Hono, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Singing voice synthesis based on generative adversarial networks. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6955–6959, 2019.

[65] John-Paul Hosom. Automatic phoneme alignment based on acoustic-phonetic modeling. In *Proceedings of Seventh International Conference on Spoken Language Processing (ICSLP)*, pages 357–360, 2002.

[66] David M. Howard, Graham F. Welch, Jude Brereton, Evangelos Himonides, Michael DeCosta, Jenevora Williams, and Andrew W. Howard. WinSingad: A real-time display for the singing studio. *Logopedics Phoniatrics Vocology*, 29(3):135–144, 2004.

[67] Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, 2009.

[68] Kanru Hua. Modeling singing f0 with neural network driven transition-sustain models. *arXiv preprint arXiv:1803.04030*, 2018.

[69] David Miles Huber. *The MIDI Manual*. Sams, Indianapolis, IN, USA, 1991.

[70] Satoshi Imai. Cepstral analysis synthesis on the mel frequency scale. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 8, pages 93–96, 1983.

[71] Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556, 1976.

[72] Hideki Kenmochi and Hayato Ohshita. Vocaloid-commercial singing synthesizer based on sample concatenation. In *Proceedings of INTERSPEECH*, pages 4010–4011, 2007.

[73] Anssi P. Klapuri. Automatic music transcription as we know it today. *Journal of New Music Research*, 33(3):269–282, 2004.

[74] Dennis H. Klatt. Review of text-to-speech conversion for English. volume 83, pages 737–793, 1987.

[75] Orestis Kostakis Kostakis and Aristides Gionis Gionis. Subsequence Search in Event-Interval Sequences. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 851–854, 2015.

[76] Nadine Kroher and Emilia Gómez. Automatic transcription of flamenco singing from polyphonic music recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):901–913, 2016.

[77] Nadine Kroher, Emilia Gómez, Amin Chaachoo, Mohamed Sordo, Jose-Miguel Díaz-Báñez, Francisco Gómez, and Joaquin Mora. Computational ethnomusicology: a study of flamenco and Arab-Andalusian vocal music. In *Springer Handbook of Systematic Musicology*, pages 885–897. Springer, 2018.

[78] Anna M Kruspe. Keyword spotting in singing with duration-modeled hmms. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1291–1295. IEEE, 2015.

[79] Jen-Wei Kuo, Hung-Yi Lo, and Hsin-Min Wang. Improved HMM/SVM methods for automatic phoneme segmentation. In *Proceedings of INTERSPEECH*, pages 2057–2060, 2007.

[80] Ki-Seung Lee. MLP-based phone boundary refining for a TTS database. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):981–989, 2006.

[81] Juan Mari Lekuona. *Ahozko euskal literatura*. Erein, 1982.

[82] S Lemmetty. "review of speech synthesis technology". Master's thesis, 1999.

[83] Micheline Lesaffre, Marc Leman, Bernard De Baets, and Jean-Pierre Martens. Methodological considerations concerning manual annotation of musical audio in function of algorithm development. In *Proceedings of 5th International conference on Music Information Retrieval (ISMIR)*, pages 64–71, 2004.

[84] Cheng-Yuan Lin and Jyh-Shing Roger Jang. Automatic phonetic segmentation by score predictive model for the corpora of mandarin singing voices. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2151–2159, 2007.

[85] Cheng-Yuan Lin, J.-S. Roger Jang, and Shaw-Hwa Hwang. An on-the-fly Mandarin singing voice synthesis system. In *Proceedings of Pacific-Rim Conference on Multimedia*, pages 631–638. Springer, 2002.

[86] Hung-Yi Lo and Hsin-Min Wang. Phonetic boundary refinement using support vector machine. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–933. IEEE, 2007.

[87] Carol C. Lochbaum and John L. Kelly. Speech synthesis. In *Proceedings of the Speech Communication Seminar*, pages 583–596, 1962.

[88] Alex Loscos, Pedro Cano, and Jordi Bonada. Low-delay singing voice alignment to text. In *ICMC*, volume 11, pages 27–61, 1999.

[89] Michael W. Macon, Leslie Jensen-Link, James Oliverio, Mark A. Clements, and E. Bryan George. A singing voice synthesis system based on sinusoidal modeling. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 435–438. IEEE, 1997.

[90] Fabrice Malfrère and Thierry Dutoit. High-quality speech synthesis for phonetic speech segmentation. In *Proceedings of EUROSPEECH*, pages 2631–2634, 1997.

[91] Ayami Mase, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. HMM-based singing voice synthesis system using pitch-shifted pseudo training data. In *Proceedings of INTERSPEECH*, pages 845–848, 2010.

[92] Matthias Mauch, Hiromasa Fujihara, and Masataka Goto. Integrating additional chord information into HMM-based lyrics-to-audio alignment. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):200–210, 2011.

[93] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, 2015.

[94] Robert McAulay and Thomas Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, 1986.

[95] Annamaria Mesaros and Tuomas Virtanen. Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010: 1–11, 2010.

[96] Luis Michelena. *Historia de la literatura vasca*, volume 7. Ediciones Minotauro, 1960.

[97] Emilio Molina, Isabel Barbancho, Emilia Gómez, Ana Maria Barbancho, and Lorenzo J Tardón. Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 744–748. IEEE, 2013.

[98] Emilio Molina, Ana Maria Barbancho-Perez, Lorenzo J. Tardón, and Isabel Barbancho-Perez. Evaluation framework for automatic singing transcription. In *Proceedings of 12th International conference on Music Information Retrieval (ISMIR)*, 2014.

[99] Emilio Molina, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Sipth: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):252–263, 2014.

[100] Yang Sae Moon and Jinho Kem. Fast normalization-transformed subsequence matching in time-series databases. *IEICE Transactions on Information and Systems*, E90-D(12):2007–2018, 2007.

[101] Hiroki Mori, Wakana Odagiri, Hideki Kasuya, and Kiyoshi Honda. Transitional characteristics of fundamental frequency in singing. In *Proceedings of 18th International Congress on Acoustics*, 2004.

[102] Kazuhiro Nakamura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Singing voice synthesis based on convolutional neural networks. *arXiv preprint arXiv:1904.06868*, 2019.

[103] Masanari Nishimura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Singing voice synthesis based on deep neural networks. In *Proceedings of INTERSPEECH*, pages 2478–2482, 2016.

[104] A. Michael Noll. Cepstrum pitch determination. *The Journal of the Acoustical Society of America*, 41(2):293–309, 1967.

[105] Yasunori Ohishi, Masataka Goto, Katunobu Itou, and Kazuya Takeda. Discrimination between singing and speaking voices. In *Proceedings of INTERSPEECH*, pages 1141–1144, 2005.

[106] Yasunori Ohishi, Hirokazu Kameoka, Daichi Mochihashi, and Kunio Kashino. A stochastic model of singing voice f0 contours for characterizing expressive dynamic components. In *Proceedings of INTERSPEECH*, pages 474–477, 2012.

[107] Santiago Oinandia. *Euskal Literatura*, volume 1 of *6*. Etor, 1972.

[108] Keiichiro Oura, Ayami Mase, Tomohiko Yamada, Satoru Muto, Yoshihiko Nankaku, and Keiichi Tokuda. Recent development of the HMM-based singing voice synthesis system—Sinsy. In *Proceedings of Seventh ISCA Workshop on Speech Synthesis*, pages 211–216, 2010.

[109] Keiichiro Oura, Ayami Mase, Yoshihiko Nankaku, and Keiichi Tokuda. Pitch adaptive training for HMM-based singing voice synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5377–5380, 2012.

[110] Serkan Özer. F0 Modeling For Singing Voice Synthesizers with LSTM Recurrent Neural Networks. Master's thesis, Univ. Pompeu Fabra, Barcelona, 2015.

[111] Edvin Pakoci, Branislav Popović, Nikša Jakovljević, Darko Pekar, and Fathy Yassa. A phonetic segmentation procedure based on hidden markov models. In *Proceedings of International Conference on Speech and Computer*, pages 67–74, 2016.

[112] Gordon E. Peterson, William S.-Y. Wang, and Eva Sivertsen. Segmentation techniques in speech synthesis. *The Journal of the Acoustical Society of America*, 30(8):739–742, 1958.

[113] Mike Plumpe, Alex Acero, Hsiao-Wuen Hon, and Xuedong Huang. HMM-based smoothing for concatenative speech synthesis. In *Proceedings of Fifth International Conference on Spoken Language Processing (ICSLP)*, number 0908, 1998.

[114] Emanuele Pollastri. A pitch tracking system dedicated to process singing voice for music retrieval. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 1, pages 341–344, 2002.

[115] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. The kaldi speech recognition toolkit. In *Proceedings of IEEE 2011 workshop on Automatic Speech Recognition and Understanding*, number EPFL-CONF-192584, 2011.

[116] Lawrence R Rabiner. Digital-formant synthesizer for speech-synthesis studies. *The Journal of the Acoustical Society of America*, 43(4):822–828, 1968.

[117] Alfredo Retortillo and Xabier Aierdi. A sociological study of sung, extempore verse-making in Basque. *Oral Tradition*, 22(2):13–31, 2007.

[118] Inmaculada Hernáez Rioja. *Sistema de conversión de texto a voz para la lengua vasca basado en un sintetizador por formantes*. PhD thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea, 1996.

[119] Xavier Rodet. Time—domain formant—wave-function synthesis. In *Spoken Language Generation and Understanding*, pages 429–441. Springer, 1980.

[120] Xavier Rodet. Musical sound signal analysis/synthesis: Sinusoidal+residual and elementary waveform models. 1998.

[121] Xavier Rodet, Yves Potard, and Jean-Baptiste Barriere. The CHANT project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 8(3):15–31, 1984.

[122] Matti P. Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3): 72–86, 2008.

[123] Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda. An HMM-based singing voice synthesis system. In *Proceedings of INTERSPEECH*, pages 2274–2277, 2006.

[124] Keijiro Saino, Makoto Tachibana, and Hideki Kenmochi. An HMM-based singing style modeling system for singing voice synthesizers. In *Proceedings of Seventh ISCA Workshop on Speech Synthesis*, pages 252–257, 2010.

[125] Keijiro Saino, Makoto Tachibana, and Hideki Kenmochi. A singing style modeling system for singing voice synthesizers. In *Proceedings of INTERSPEECH*, pages 2894–2897, 2010.

[126] Takeshi Saitou and Masataka Goto. Acoustic and perceptual effects of vocal training in amateur male singing. In *Proceedings of INTERSPEECH*, pages 832–835, 2009.

[127] Takeshi Saitou, Masashi Unoki, and Masato Akagi. Development of an f0 control model based on f0 dynamic characteristics for singing-voice synthesis. *Speech communication*, 46(3-4):405–417, 2005.

[128] Xabier Sarasola, Eva Navas, David Tavarez, Daniel Erro, Ibon Saratxaga, and Inma Hernaez. A singing voice database in Basque for statistical singing synthesis of Bertsolaritza. In *Proceedings of LREC*, pages 756–759, 2016.

[129] Xabier Sarasola, Eva Navas, David Tavarez, Luis Serrano, and Ibon Saratxaga. Speech and monophonic singing segmentation using pitch parameters. In *Proceedings IberSPEECH 2018*, pages 147–151, 2018.

[130] Xabier Sarasola, Eva Navas, David Tavarez, Luis Serrano, Ibon Saratxaga, and Inma Hernaez. Application of pitch derived parameters to speech and monophonic singing classification. *Applied Sciences*, 9(15):3140, 2019.

[131] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8): 1627–1639, 1964.

[132] Bernhard Scholkopf and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[133] Manfred Robert Schroeder. Vocoders: Analysis and synthesis of speech. *Proceedings of the IEEE*, 54(5):720–734, 1966.

[134] Emery Schubert and Joe Wolfe. The rise of fixed pitch systems and the slide of continuous pitch: A note for emotion in music research about portamento. *Journal of Interdisciplinary Music Studies*, 7(1-2):1–27, 2013.

[135] Björn Schuller, Bernardo José Brüning Schmitt, Dejan Arsić, Stephan Reiter, Manfred Lang, and Gerhard Rigoll. Feature selection and stacking for robust discrimination of speech, monophonic singing, and polyphonic music. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 2005, pages 840–843, 2005.

[136] Björn Schuller, Gerhard Rigoll, and Manfred Lang. Discrimination of speech and monophonic singing in continuous audio streams applying multi-layer support vector machines. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 1655–1658, 2004.

[137] Xavier Serra and Julius Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.

[138] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[139] Kanako Shirota, Kazuhiro Nakamura, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Integration of speaker and pitch adaptive training for HMM-based singing voice synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2559–2563, 2014.

[140] Peter Smit, Sami Virpioja, and Mikko Kurimo. Improved subword modeling for WFST-based speech recognition. In *Proceedings of INTERSPEECH*, pages 2551–2555, 2017.

[141] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45 (4):427–437, 2009.

[142] J. Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987. ISBN 9781565935839. URL https://books.google.es/books?id=VvnOAAAACAAJ.

[143] Johan Sundberg. The acoustics of the singing voice. *Scientific American*, 236 (3):82–91, 1977.

[144] Johan Sundberg. Synthesis of singing by rule. In Max V. Mathews and John R. Pierce, editors, *Current directions in computer music research*, pages 45–55. MIT Press, Cambridge, MA, United States, 1989.

[145] Johan Sundberg. Acoustic and psychoacoustic aspects of vocal vibrato. *Speech, Music and Hearing - Quarterly Progress and Status Report*, 35(2–3): 45–68, 1994.

[146] Johan Sundberg. Level and center frequency of the singer's formant. *Journal of voice*, 15(2):176–186, 2001.

[147] Johan Sundberg. The KTH synthesis of singing. *Advances in cognitive Psychology*, 2(2-3):131–143, 2006.

[148] Johan A.K. Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[149] Margareta Thalén and Johan Sundberg. Describing different styles of singing: A comparison of a female singer's voice source in "Classical", "Pop", "Jazz" and "Blues". *Logopedics Phoniatrics Vocology*, 26(2):82–93, 2001.

[150] Brian Thompson. Discrimination between singing and speech in real-world audio. In *Proceedings of IEEE Workshop on Spoken Language Technology (SLT)*, pages 407–412, 2014.

[151] Tomoki Toda and Keiichi Tokuda. A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Transactions on Information and Systems*, 90(5):816–824, 2007.

[152] Chee-Chuan Toh, Bingjun Zhang, and Ye Wang. Multiple-feature fusion based onset detection for solo singing voice. In *Proceedings of 9th International conference on Music Information Retrieval (ISMIR)*, pages 515–520, 2008.

[153] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1315–1318, 2000.

[154] Dave Tompkins. *How to wreck a nice beach: The vocoder from World War II to hip-hop: The machine speaks*. Melville House, 2011.

[155] Robert Lawrence Trask. *The history of Basque*. Psychology Press, 1997.

[156] Wei Ho Tsai and Cin Hao Ma. Speech and singing discrimination for audio data indexing. In *Proceedings of IEEE International Congress on Big Data*, pages 276–280, 2014.

[157] Martí Umbert, Jordi Bonada, and Merlijn Blaauw. Generating singing voice expression contours based on unit selection. In *Proceedings of Stockholm Music Acoustics Conference (SMAC)*, pages 315–320, 2013.

[158] Yusuke Wada, Ryo Nishikimi, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Sequential generation of singing f0 contours from musical note sequences based on wavenet. In *Proceedings of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 983–989, 2018.

[159] Chong-Kai Wang, Ren-Yuan Lyu, and Yuang-Chin Chiang. An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker. In *Proceedings of EUROSPEECH*, pages 1197–1200, 2003.

[160] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin. LyricAlly: automatic synchronization of acoustic musical signals and textual lyrics. In *Proceedings of the 12th annual ACM International Conference on Multimedia*, pages 212–219, 2004.

[161] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *Proceedings of INTERSPEECH*, pages 4006–4010, 2017.

[162] Colin W. Wightman and David T. Talkin. The aligner: Text-to-speech alignment using Markov models. In Sproat R.W. van Santen J.P.H., Olive J.P. and Hirschberg J., editors, *Progress in speech synthesis*, pages 313–323. Springer, New York, NY, United States, 1997.

[163] Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An open source neural network speech synthesis system. In *Proceedings of 9th ISCA Speech Synthesis Workshop (SSW)*, pages 202–207, 2016.

[164] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. The HTK book (v3. 4). *Cambridge University*, 2006.

[165] Steve J. Young, Julian J. Odell, and Philip C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology (HLT)*, pages 307–312, 1994.

[166] Heiga Zen, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7962–7966, 2013.

[167] Jan Zera, Jan Gauffin, and Johan Sundberg. Synthesis of selected VCV-syllables in singing. In *International Computer Music Conference Proceedings*, pages 83–86, IRCAM, Paris, 1984.

[168] Yong Zhao, Lijuan Wang, Min Chu, Frank K. Soong, and Zhigang Cao. Refining phoneme segmentations using speaker-adaptive context dependent boundary models. In *Proceedings of INTERSPEECH*, pages 2557–2560, 2005.