Automatic Vocal Ensemble Intonation Analysis

Eugen Azcoaga

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 10.10.2016

Thesis supervisor:

Prof. Vesa Välimäki

Thesis advisor:

Prof. Jaakko Lehtinen

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Eugen Azcoaga

Title: Automatic Vocal Ensemble Intonation Analysis

Date: 10.10.2016 Language: English Number of pages: 7+61

Department of Signal Processing and Acoustics

Professorship: Audio Signal Processing

Supervisor: Prof. Vesa Välimäki

Advisor: Prof. Jaakko Lehtinen

The objective of this study is a specific music signal processing task, primarily intended to help vocal ensemble singers practice their intonation. In this case intonation is defined as deviations of pitch in relation to the note written in the score which are small, less than a semitone. These can be either intentional or unintentional. Practicing intonation is typically challenging without an external ear. The algorithm developed in this thesis combined with the presented application concept can act as the external ear, providing real-time information on intonation to support practicing. The method can be applied to the analysis of recorded material as well. The music signal generated by a vocal ensemble is polyphonic. It contains multiple simultaneous tones with partly or completely overlapping harmonic partials. We need to be able to estimate the fundamental frequency of each tone, which then indicates the pitch of each singer. Our experiments show, that the fundamental frequency estimation method based on the Fourier analysis developed in this thesis can be applied to the automatic analysis of vocal ensembles. A sufficient frequency resolution can be achieved without compromising the time resolution too much by using an adequately sized window. The accuracy and robustness can be further increased by taking advantage of solitary partials. The greatest challenge turned out to be the estimation of tones in octave and unison relationships. These intervals are fairly common in tonal music. This question requires further investigation or another type of approach.

Keywords: Acoustics, music, signal analysis, spectral analysis, frequency estimation

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Eugen Azcoaga

Työn nimi: Lauluyhtyeen intonaation automaattinen määritys

Päivämäärä: 10.10.2016 Kieli: Englanti Sivumäärä: 7+61

Signaalinkäsittelyn ja akustiikan laitos

Professuuri: Audiosignaalinkäsittely

Työn valvoja: Prof. Vesa Välimäki

Työn ohjaaja: Prof. Jaakko Lehtinen

Tässä työssä tutkitaan erityistä musiikkisignaalin analysointitehtävää, jonka tarkoituksena on auttaa lauluyhtyelaulajia intonaation harjoittelussa. Intonaatiolla tarkoitetaan tässä yhteydessä pieniä, alle puolen sävelaskeleen säveltasoeroja nuottiin kirjoitettuun sävelkorkeuteen nähden, jotka voivat olla joko tarkoituksenmukaisia tai tahattomia. Intonaation harjoittelu on tyypillisesti haastavaa ilman ulkopuolista korvaa. Työssä kehitetty algoritmi yhdessä esitellyn sovelluskonseptin kanssa voi toimia harjoittelutilanteessa ulkopuolisena korvana tarjoten reaaliaikaista tietoa intonaatiosta harjoittelun tueksi. Vaihtoehtoisesti menetelmää voidaan hyödyntää harjoitusäänitteiden analysointiin jälkikäteen. Lauluyhtyeen tuottama musiikkisignaali on polyfoninen. Se sisältää useita päällekkäisiä säveliä, joiden osasävelet menevät toistensa kanssa osittain tai kokonaan päällekkäin. Tästä signaalista on pystyttävä tunnistamaan kunkin sävelen perustaajuus, joka puolestaan kertoo laulajan laulaman sävelkorkeuden. Kokeellisten tulosten perusteella työssä kehitettyä Fourier-muunnokseen perustuvaa taajuusanalyysiä voidaan soveltaa lauluyhtyeen intonaation automaattiseen määritykseen, kun nuottiin kirjoitettua sointua hyödynnetään analyysin lähtötietona. Sopivankokoista näyteikkunaa käyttämällä päästiin riittävään taajuusresoluutioon aikaresoluution säilyessä kohtuullisena. Yksinäisiä osasäveliä hyödyntämällä voidaan edelleen parantaa tarkkuutta ja toimintavarmuutta. Suurimmaksi haasteeksi osoittautui oktaavi- ja priimisuhteissa olevien intervallien luotettava määritys. Näitä intervallisuhteita esiintyy tonaalisessa musiikissa erityisen paljon. Tämä kysymys vaatii vielä lisätutkimusta tai uudenlaista lähestymistapaa.

Avainsanat: Akustiikka, musiikki, signaalianalyysi, taajuusanalyysi, taajuusestimointi

Preface

Working with this thesis has been an exciting journey into the world of music signal analysis. I have only been able to scratch the surface of this fascinating research area, but it has nevertheless been a widening experience. Signal processing and mathematics have not always been on the top of my list of areas of interest, but during this research I have had the privilege of familiarising with these topics through an application area that is really dear to me – vocal ensemble singing. I want to thank my advisor Jaakko for giving the idea and inspiration for this topic, and for guiding and facilitating my day-to-day work. I also want to thank my supervisor Vesa for his encouragement and his audio signal processing related insights and practical tips. In addition, I want to thank my friends and family for being extremely supportive throughout the whole process. In particular I want to thank my friend Eero for his incredible peer support, not only during this thesis work, but throughout almost my entire bachelor and master degree studies. It goes without saying that this work would not have been this rewarding without the contribution of my fellow vocal ensemble singers, Antti, Pauli and Matias, who have been investing their valuable time in participating in my practical experiments. Finally I want to thank Suvi for believing in me and for giving her unconditional support, during this thesis work, and in life.

Espoo, 10.10.2016

Eugen Azcoaga

Contents

Abbreviations

ACF	Autocorrelation function
CBHPS	Cepstrum-Biased HPS
CQIFFT	QIFFT with correction for the bias of the window function
DFT	discrete Fourier transform
F0	Fundamental Frequency
FFT	Fast Fourier Transfrom
FT	Fourier Transform
HPS	Harmonic Product Spectrum
IDFT	Inverse Discrete Fourier Transform
IF	Instantaneous Frequency
IFT	Inverse Fourier Transform
IOI	Inter-Onset Interval
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
ML	Maximum Likelihood
PV	Phase Vocoder
QIFFT	Quadratic interpolated FFT
STFT	Short-Time Fourier Transform
UI	User Interface
WACF	Weighted Autocorrelation Function
YIN	A fundamental frequency estimator based on the autocorrelation method

1 Introduction

This work focuses on a specific music signal processing task, primarily intended to help vocal ensemble singers practice their intonation. It is closely related to the research areas of music information retrieval (MIR) [1], automatic music transcription [2], and signal processing for music analysis [3]. These fields have been rapidly growing during the past years due to the continuously increasing computational capabilities, and their increased demand in real-world applications, such as Spotify[1] and Shazam[2] to name a few.

Music information retrieval is, as described by J. S. Downie, "a multidisciplinary research endeavour that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world's vast store of music accessible to all" [4]. Among many subfields, it involves the extraction of meaningful *music features* from an audio signal [1], which is also the foundation for automatic music transcription [2] and the essence of signal processing for music analysis [3].

Our aim is to extract a music feature called *pitch* from a music signal containing singing of a *vocal ensemble*. The pitch information is then used to analyse the *intonation* of the vocal ensemble. We will cover these terms, as well as other needed terminology and concepts, more thoroughly in the upcoming chapter of this thesis.

With the term vocal ensemble we refer to a group of singers, with typically three to eight members, singing as a group, and each of the singers singing his or her own *part*. In other words, the singers do not share the same *tones*, but each of the individual parts sung together form a *harmony*.

Furthermore, we are interested in *a cappella* vocal ensemble singing in particular, meaning vocal ensemble singing without accompanying instruments, since this kind of music involves the challenges we are aiming to solve with our suggested application. A cappella vocal ensemble singing is a niche area of music, but it definitely plays an important role in the history of Western music, as well as affects the lives of many people around the world. For example, a significant part of early music from the Renaissance era is performed by a cappella vocal ensembles. In some countries, for example Finland, choir music played an important role in the era of romantic nationalism of the 19th century. A cappella music in general has been present in Western music since the early ages until today, including classical, popular and folk genres. In addition, it is popular around the world to to sing in either amateur, semi-professional or professional choirs performing a cappella.

For practical reasons, we have narrowed the scope of this thesis into ensembles where each part is sung by a single individual, but the same idea could be applied to choirs of any size singing a cappella. This would however require taking into consideration the dispersion of intonation within each part, since in this case there is more than one individual singing the same part.

Intonation, in our case, refers to the small pitch adjustments that vocal ensemble singers are constantly applying during their performance. Intonation can be either

[1] `https://www.spotify.com/`
[2] `http://www.shazam.com/`

intended or unintended. Intended intonation is part of the musical expression and related to the wider musical context of the musical *tones* produced by the singers. Unintended intonation errors, which are very common for vocal ensemble singers ranging from beginners and amateurs to top-level professional performers, are caused by precision errors and limitations both of the ear and the voice of the singers. Our work is especially motivated by the need of a means to instantly identify and correct unintended intonation errors.

There are a few factors which make the intonation of a vocal ensemble something worth analysing automatically, in other words computationally. A cappella vocal ensemble singing, like any other form of singing or music performing, has its own special characteristics. Two of these characteristics in particular are strongly affecting the nature of a cappella vocal ensemble singing.

Firstly, the singers of an a cappella vocal ensemble need to be harmonious in their singing. Creating an excellent harmony is the very foundation and essence of musical expression in a cappella vocal ensemble singing. In order to achieve this, the singers need to have a maximum level of precision and coordination between each other in their singing style. The singers must learn to phrase, articulate, and intonate the music and the lyrics together, basically, as they were one instrument. This is quite the opposite approach compared to solo singing, where musical expression comes from the freedom the artist is allowed and supposed to take on these musical aspects.

Secondly, singing a cappella leaves total freedom and responsibility for the singers to maintain their pitch, key, and intonation. As opposed to instruments like the piano or the guitar where the pitch of each tone is fixed and tuned beforehand, the pitch of the human voice can be freely tuned, and in a cappella singing there are no instruments with fixed tuning to lean on during the performance.

In addition, a vast majority of musicians need to hear a reference tone to be able to produce a given musical tone. Only a few people, 1 in 10 000 [5], possess a quality called absolute pitch, which makes them able to identify and produce musical tones without a reference tone. An average singer needs to train both the ear and the voice in order to gain the necessary precision in intonation that is required for a cappella vocal ensemble singing.

Due to these above mentioned characteristics, there are some typical challenges for a cappella vocal ensemble singers regarding intonation. With the lack of both absolute pitch and accompanying instruments, the singer has to develop some other capability to remember and maintain the key given in the beginning of the performance. Basically this is achieved by practicing, but it is very common for the key to shift either up or down, most commonly down, from the original key. To some extent it is tolerable and even a part of natural intonation in a cappella singing for the key to slightly shift up and down during a musical piece. Nevertheless, usually singers want to eliminate of this phenomenon in their singing, since a permanent shift of the key during a performance affects the timbre of the vocal ensemble and might cause technical challenges to produce high and low tones near the limits of the vocal range of each singer.

Another common challenge is singing single tones slightly out of tune. There are many reasons for singing out of tune, most often related to an untrained ear or an

untrained voice. Even professional singers are constantly struggling with this issue. Singing tones slightly out of tune in an a cappella vocal ensemble leads to issues with harmony, and it can also be the root cause for the shifting of the key. Singing very precisely in tune is important for vocal ensemble singers in particular, since it is the key to success in forming an excellent harmony.

Practicing to maintain the key and to sing in tune can be very demanding for a vocal ensemble. It is common that the issue is noticed but the cause cannot be identified accurately. Firstly, the singers must identify which of them is causing the issue, and secondly they must identify, if the tone was sung too high or too low. If the ensemble is not able to verify these factors, the singers often start correcting their intonation in the wrong direction and end up getting more issues.

There are some means that are commonly used to identify the causes of the issues. The ensemble can use a piano as reference to check their pitch along the singing. It is also possible to make recordings during exercise sessions and use them to analyse issues afterwards. A third means is to have someone else present at the exercise session as an external ear, who can more easily spot issues of the ensemble as a whole and identify the causes of the issues.

All of these currently used means have their limitations. Using a piano during the exercise session takes part of the focus of the singers away from singing as an ensemble, since they need to concentrate simultaneously on their singing, listening to others and playing and listening to the piano. In addition, the tuning of a typical rehearsal piano is not always accurate and reliable enough to be used as reference for vocal ensemble intonation fine tuning. At best, it can be used to spot rough unintended intonation errors.

Using recordings is a powerful means, as the singers are able to hear themselves from the perspective of an external ear, without having to concentrate on their own singing simultaneously. It would still be more efficient if the singers could receive instant feedback instead of listening to the recording afterwards. In addition, finding the cause of an intonation issue by listening can be difficult even from a recording.

Finding a person to act as an external ear may be challenging. Typically, this method is not used regularly by vocal ensembles in every single rehearsal. It is rather considered as a rare luxury. In addition, even a very skilled musician is not always able to directly hear and identify the root cause of every intonation issue.

All of these limitations of the means of identifying root causes of intonation issues in a cappella vocal ensemble singing could be relieved by using automatic intonation analysis. By using automatic real-time intonation analysis instead of a piano, the singers receive instant feedback without the distraction and inaccuracy of the piano. The real-time application could show the estimated pitch of each singer on a screen, and the singers could directly adjust their singing based on the real-time feedback they get from the application. The intonation analysis could also be performed on recordings. In addition to what they hear from the recording, the singers would receive precise feedback on their intonation in a graphical representation. Both of these use cases for an application would also minimise the need of a human external ear in rehearsing sessions.

The main emphasis of the thesis is in developing and technically evaluating an

algorithm that estimates the pitch of each singer from a polyphonic music audio signal, and calculates the deviation between the estimated pitch and a given reference pitch. The experiments will be carried out using the MATLAB[3] platform and programming language.

In order to find the right scope, we cover the necessary amount of background theory in music and acoustics, and develop a preliminary end-user application concept. We also cover the signal processing foundations required for this type of work, before examining this particular music analysis task more thoroughly, as well as perform a literature overview on related research and applications. The main contribution, however, is designing, developing, and evaluating the algorithm for this particular use case.

The aim is to develop an algorithm that can be used in real-time. A real-time application would allow the singers of a vocal ensemble to receive instant feedback on their intonation while practicing together. This real-time use case, however, sets quite hard requirements on the performance speed of the algorithm. Another use case for the algorithm would be to analyse recorded audio in non-real-time. This kind of application would also be beneficial for a cappella vocal ensemble singers, as described above.

[3] http://mathworks.com/products/matlab/

2 Background in music and acoustics

For the purpose of developing a technique for music signal analysis, it is advisable to take advantage of all the insight there is available about the nature of the signal and even music itself [2, 3]. The more closely the processing can reflect and exploit the particular properties of the signal, the more successful the application will be. This chapter covers the necessary amount of musical and acoustical background theory in order to be able to design and evaluate an algorithm for automatic vocal ensemble intonation analysis.

The minimum requirement is to understand the horizontal and vertical dimensions of music, as well as the concepts of polyphony, harmony, intonation, tuning and temperament. In order to understand these concepts, it is also required to define musical pitch and tone, as well as intervals, chords and scales. All these concepts are based on physical acoustical phenomena, such as the fundamental frequency and harmonic partials of complex tones.

The foundations of how musical sound events are produced and how humans and the human ear perceives them is extensively covered in scientific literature [6, 7, 8]. This thesis covers only the very essential information, which is needed to design and evaluate our algorithm.

Since the application is specifically targeted for vocal ensemble music, we also cover the particular characteristics related to this specific subset of music. The instruments contained in the music signal are known to be singing voices. In addition, we limit this study to Western music, in particular classical tonal vocal ensemble music, and cover the concepts related to intonation in this genre.

The emphasis on Western classical tonal a cappella vocal ensemble music, however, does not exclude the application of the contributions of this thesis to other musical genres, such as pop, folk or contemporary atonal music, or even to other instruments, such as string or wind instruments. Nevertheless, we keep this emphasis in order to obtain a reasonable scope for the research and to take advantage of the known characteristics of a music signal containing Western classical tonal a cappella vocal ensemble music.

2.1 Basic concepts and terminology

In terms of physics, music is acoustic information [6]. The carrier of information is the sound wave, which is elastic energy, practically oscillations of pressure. This energy is emitted from a source, transmitted by a medium, and received by a receptor. The source is the instrument, which in our case is the singing voice. The medium is air, and the receptor is the listener.

The audible frequency range of humans is determined to be 20 Hz to 20 kHz [7]. We can assume that outside this range there is no acoustic information. This is the maximum frequency range of the oscillation of audible sound waves, which we are analysing.

A pure tone, also called simple tone, is a sound wave whose instantaneous pressure variation as a function of time is a sinusoidal function [8]. This can be expressed

mathematically.

$$x(t) = A\sin(\omega t + \phi) = A\sin(2\pi f t + \phi) \tag{1}$$

A is the maximum amplitude, in other words the peak deviation from zero. f is the ordinary frequency, in other words the number of cycles or oscillations per each second of time. $\omega = 2\pi f$ is the angular frequency, in other words the rate of change of the function argument. ϕ is the phase, which specifies where the cycle is at zero.

Basically all naturally produced tones are complex tones [7]. A complex tone is composed of a number of sinusoids of different frequencies [8]. Complex tones can be either harmonic or inharmonic. When the complex tone is purely harmonic, it is composed by sinusoids, which are integer multiples of the lowest frequency component. In the most simplified case, this can be mathematically represented as a sum of sinusoids.

$$x(t) = \sum_{n=1}^{N} A_n \sin(n\omega t + \phi_n) = \sum_{n=1}^{N} A_n \sin(2\pi n f t + \phi_n) \tag{2}$$

The lowest frequency component in the sum of sinusoidal waves of a complex tone is called the fundamental frequency (F0). The rest of the frequency components are called harmonic partials, harmonics, or as music professionals often refer to them, overtone partials or overtones.

Natural sounds may contain all or only part of the harmonic components depending on the sound source [7]. For instance, string instruments, such as the violin or the guitar, contain all the harmonic components, while for example wind instruments, which are closed pipes, such as the clarinet, contain only odd harmonics.

One of the basic units in music is the musical tone. A musical tone is characterised by its pitch, loudness and timbre [6]. These are all subjective attributes, and thus difficult to define formally. They are not accessible to direct physical measurement. For the scope and purpose of this thesis we can agree, as it is frequently described, that the pitch is the sensation of altitude or height, loudness is the sensation of strength or intensity, and timbre is the sensation of colour of a musical tone.

Despite being subjective attributes, in other words psychophysical magnitudes, each of these three sensations, pitch, loudness and timbre, can be associated to a well-defined physical quantity of the original stimulus, which is the sound wave produced by an instrument or the singing voice [6]. These quantities can be measured and expressed numerically by physical methods. Pitch is primarily associated to the fundamental frequency, loudness to the intensity and timbre to the spectrum envelope of the sound wave.

This association is a simplified model of how we perceive pitch, loudness and timbre. For instance, pitch perception of a pure tone is also relative to the sound level, duration and envelope of the sound wave [6, 7]. In addition, we are able to hear the pitch of a complex tone based on the upper harmonic partials even when the fundamental frequency is missing. Pitch may also be perceived slightly different in one ear compared to the other, and other interfering sounds may affect pitch perception of a particular tone. Nevertheless, to keep a reasonable scope for this thesis we will stick with the simplified interpretation, and equate the perceived

pitch with the fundamental frequency. This is a common practice in music analysis applications [3].

It is left for further research to investigate the effect of these more refined aspects of pitch, loudness and timbre perception on the practical subjective accuracy of our intonation analysis algorithm. This would be a psychoacoustic experiment, where the relationships between an acoustic stimulus and the resulting subjective sensation are studied by carrying out listening tests to human test subjects [7].

One more relevant attribute of a musical tone, besides pitch, loudness and timbre, is duration [7]. The perceived duration has a more or less one-to-one mapping to the physical duration of the musical tone [2].

Pitch is the most important of the four attributes defining a musical tone in terms of our application. It is mainly pitch, which defines the intonation of a vocal ensemble. As previously stated pitch is the sensation of altitude of a musical tone. In other words, pitch is the characteristic of a sound that makes it sound high or low [7]. Musical tones can be ordered from low to high according to pitch.

Musical information is generally encoded into the relationships between tones and larger entities composed of these [2]. The relationship between two tones, whether played subsequently or simultaneously, is represented as the interval. When two tones have fundamental frequencies in a ratio of 2:1, humans perceive these tones as highly similar [8]. This ratio is called the octave, and the perceived similarity is also known as the octave equivalence [3]. Presumably, the basis of the perceived similarity is due to the fact that the harmonic partials of the lower tone in an octave are a proper superset of the harmonic partials of the higher tone [3].

The musical scale depends on how the octave is divided into smaller pieces, forming a set of discrete pitches that repeats every octave [3]. The musical scale in different forms is a common feature amongst different cultures. Contemporary Western music is based on the equal tempered scale. It allows the octave to be divided into twelve equal steps on a logarithmic axis. Equal temperament, however, is only one of many temperaments that have been used in Western music over time, as we will discuss later in this chapter.

An individual musical piece usually consists of only a subset of the twelve tones, also called twelve pitch classes [2]. The subset depends on the musical key of the piece. In Western music there are typically seven pitch classes, also called scale tones, corresponding to a given key. The most typical scales in contemporary Western music are the major and the minor key scales, which both consist of different seven pitch class sublets of the twelve step scale.

Each scale tone has a varying degree of importance or stability in the key context, as we will discuss later in this chapter [2]. The most important is the tonic note. A musical piece often starts and ends on the tonic.

Most of Western music belongs to the category of tonal music. Perception of pitch along musical scales and in relation to the musical key of the piece, as described above, is specifically characteristic to tonal music [9]. The basis of Western vocal ensemble repertoire belongs to the category of tonal music as well. In addition, it is common for vocal ensembles to sing contemporary atonal music, as well as early music. All of these subcategories of vocal ensemble repertoire need a slightly different

approach for intonation, as we will discuss in more detail later in this chapter.

Melodies are made up from pitch relationships [2]. Melody is the tune of a music piece, and it is constructed by a sequence of tones [3]. More precisely, in a melody the tones have musically meaningful pitch and inter-onset interval (IOI) relationships [2].

Another aspect of music is harmony, which studies the formation and relationships of chords. [2, 3]. Earlier in this study we defined that an interval represents the relationship between two tones. A chord is a combination of two or more tones played simultaneously. Our application for vocal ensemble analysis is basically a chord analyser, which makes harmony one of the most important aspects of music for us to understand.

Different chords result in different musical colours [3]. Some chords are consonant or harmonious, others are dissonant [2, 3]. For the listener, consonant chords are judged to sound more pleasant than dissonant chords. These are subjective attributes and related to the relationships between the pitch of the tones. Consonant chords are constructed from pitches with simple frequency ratios, and they have many shared overtone partials. A chord remains recognisable regardless of which instrument plays it.

Music can be seen as having two dimensions, the horizontal and the vertical. The horizontal dimension is the temporal aspect of music, how music is constructed over time. The horizontal dimension is expressed by the tempo, beat, rhythm and melody of the music. Earlier we described that melody is constructed by a sequence of tones and made up from pitch relationships. Tempo, beat and rhythm are related to the durations and timings of musical tones in a sequence.

The vertical dimension of music is expressed by harmony and timbre. As we previously described, harmony is constructed by playing or singing multiple tones together. More concretely, the vertical dimension is constructed by playing or singing chords. We have also already discussed, that timbre is the sensation of colour of a single musical tone, and depends on the spectral envelope of the sound. Different instrument and individual singing voices can be separated from each other based on their timbre.

These two dimensions are also present in the standard Western notation system. In written music, time flows from left to right, and the pitch of the notes is indicated by their vertical position on the staff lines [2]. Therefore, a melody is written as a horizontally flowing sequence of notes. Harmony and chords are written as vertically aligned notes. Figure 1 is a sample of a vocal ensemble or choir piece written in standard music notation. It represents the horizontal and vertical aspects of music well. The higher the note, the higher the pitch. Vertically aligned notes are sung simultaneously, and together they form chords and harmony. Time flows horizontally from left to right, and in this case the highest part, the top notes, are forming the melody of the piece.

The twelve-tone musical scale can be represented graphically using standard notation. In addition, notes can be labeled using the following letters and symbols: C, C#, D, D#, E, F, F#, G, G#, A, A#, H. This labelling repeats every octave, and the octave is indicated with a number, for example C3 or C4.

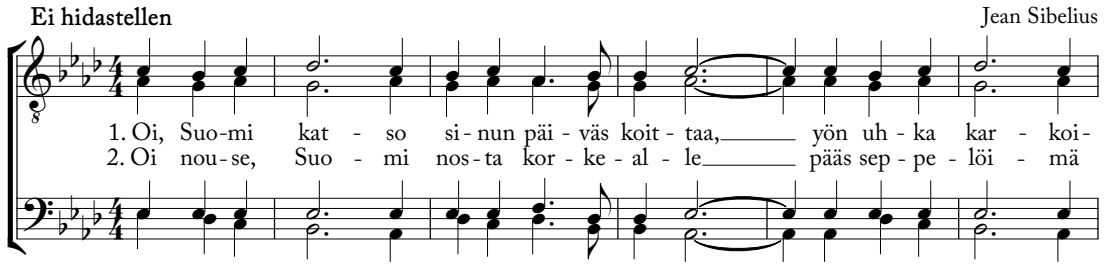

Figure 1: A sample of a typical male choir or vocal ensemble piece written in standard music notation. It consists of four parts, tenor 1, tenor 2, bass 1 and bass 2. Tenor parts are written on the higher staff and bass parts on the lower one. The four parts together form the harmony of the piece, which is the vertical aspect of music. Melody, which is sung by highest voice, tenor 1, flows horizontally from left to right. The remaining three parts could also be interpreted as separate melodies, although they are not intended to be sung independently.

The pitch of notes in Western music is calculated from a pre-determined reference tone. Most typically this reference tone is A4 set to 440 Hz. All the instruments are then tuned according to the reference tone. Another frequently used tuning is 442 Hz, for example in symphony orchestras. Early music performed with periodic instruments typically uses 415 Hz for A4 as the tuning reference. The fundamental frequency of any note in an equally tempered twelve-tone scale can be calculated with the following equation.

$$f = F_r * 2^{n/12} \tag{3}$$

Here F_r is the reference tone, which is most often set to A4 = 440 Hz. The variable n varies depending on how many semitones up or down is the tone from the reference tone, and f is the frequency of the calculated tone. Semitone is the term for one step in the twelve-tone scale, in other words one twelfth of an octave.

Standard notation is a symbolic form of writing music, which requires quantisation [2]. Most instruments are explicitly designed and constructed to allow performers to produce these quantised tones. These instruments are also tuned beforehand, and in some of them the performer cannot affect the tuning, temperament or intonation during performance. One example of this kind of instruments is the piano. Most of the instrument produce some level of arbitrary pitch values, and not just discrete notes. E.g. instruments based on vibrating air columns, such as flutes, clarinets and trumpets, produce discrete notes in principle, but in all of them the performer possesses some level of control on intonation. Non-fretted string instruments like the violin are even more sensitive to produce arbitrary pitches, and the singing voice has total freedom. Both intentional and unintentional deviations take place from the written discrete pitch values, when playing violin or singing. These deviations are the ones we are interested in analysing with our algorithm.

Lastly before moving to the next subchapter, we will cover the terms monophonic and polyphonic. In a monophonic signal at most one note is sounding at a time [2]. A melody without harmony is a monophonic music signal. A monophonic

instrument can reproduce only the melody without any harmony. The singing voice is a monophonic instrument, as well as all the wind instruments. Polyphonic instruments, like the piano, can reproduce chords in addition to melodies, because multiple tones can be played at the same time. A vocal ensemble is able to perform polyphonic music, because the multiple voices can reproduce chords and harmony when singing together simultaneously. This is the primary target material considered in this thesis.

2.2 The singing voice

In order to define and analyse a signal representing a vocal ensemble, we need to understand how the human singing voice is produced, as well as the behaviour of multiple simultaneous singing voices. In this subchapter we present an overview of the singing voice.

Singing sounds are produced by the human vocal organ [10]. It consists of three basic units: the respiratory system, the vocal folds, and the vocal tract [11]. The sound production process begins with the respiratory system creating an overpressure of air in the lungs. This is called the subglottic pressure, which creates an air flow through the vocal folds. Due to this air flow, the vocal folds start to vibrate, chopping the air flow into a sequence of quasi-periodic air pulses. These air pulses result in a sound, called the voice source, with a measurable fundamental frequency. Finally, the voice source passes through the vocal tract. The vocal tract modifies the spectral shape of the sound, determining the timbre of the singing voice. The generation of a voiced sound in the vocal folds is referred to as phonation, and the spectral shaping in the vocal tract is referred to as articulation.

The fundamental frequency of the singing voice is primarily controlled by the vocal folds [10]. The vibration frequency of the vocal folds is referred to as phonation frequency, which is the fundamental frequency of the generated singing tone. The vibration is controlled mainly with the musculature of the vocal folds. In addition, the amount of the subglottic pressure affects the to the phonation frequency. The greater the pressure, the higher the frequency. Phonation frequencies range from around 100 Hz for male bass singers to over 1000 Hz for female soprano singers.

The tone generated by the vocal folds includes overtone partials in addition to the fundamental frequency [10]. Some instruments, like the harpsichord, produce slightly inharmonic partials [12]. In the case of the vocal folds the partials are purely harmonic, meaning they are exact integer multiples of the fundamental frequency [10]. The amplitudes can be expected to decrease by 12 dB per octave in a simplified case [13].

The timbre of the singing voice is mostly controlled by the vocal tract [10]. The vocal tract acts as a resonating filter which emphasises certain frequencies called formant frequencies or formants. The articulators, including the jaw, tongue, and lips, control the shape of the vocal tract. In voiced sounds, the two lowest formants contribute most to the identification of a vowel. Higher formants contribute to the personal voice timbre of an individual singer.

In addition, the timbre of the singing voice is affected by the tension of the vocal

folds and the amount of subglottic pressure [10]. These result in different types of phonation, including pressed, normal, flow, breathy, and whisper, given in decreasing order of subglottic pressure. The vocal folds are not completely closed in normal phonation. When the subglottic pressure is high, the vocal folds produce a more pressed sound by closing more rapidly. In the breathy and whisper phonation, the amount of subglottic pressure is insufficient, and the vocal folds are not able to vibrate properly. This results in a breathy phonation. The flow phonation is ideal for singing, since the produced sound is neither pressed nor leaky.

The loudness of singing is mainly controlled with the amount of subglottic pressure and the type of phonation [10]. The sound pressure level of the voice source is maximal when using the flow phonation, although the subglottic pressure is smaller than in the normal phonation. In addition, trained singers can modify their vocal tract in order to match the phonation frequency with a formant frequency, creating a louder sound. This is known as the singer's formant.

The phonation frequency associates a singing sound with a musical note [10]. As we have already indicated in the introduction chapter, singing performances include both intentional and unintentional deviations from the nominal note pitches. Intended deviations in phonation frequency are commonly used to enhance the expressiveness of a singing performance. Unintended deviations are mostly due to the lack of voice training. Nevertheless, there is a perceptual association of phonation frequencies with notes even when the phonation frequency is not stable during the note. In order to understand this phenomenon, we consider issues such as vibrato, tremolo, glissando, legato, and singing out of tune.

Vibrato refers to the modulation of the phonation frequency during a performed note [10]. It can be characterised with the rate and depth of the modulation. Typically the rate varies between 4–7 Hz [14], and the depth between 0.3–1 semitones. The mean of the fundamental frequency during a vibrato note is close to the perceived pitch, according to experimental results with human listeners [15]. Tremolo refers to the fluctuation of loudness [10]. There is a high correlation between vibrato and tremolo.

Glissando refers to a slide in phonation frequency [10]. It is usually employed at the beginning of long notes. A typical singing manner is to start from a lower pitch, in other words flat, and then match the phonation frequency to the note pitch during the first 200 ms of the note [16]. Legato refers to tiding consecutive notes together [10]. In order to preserve legato when changing note pitch, the change is performed through a glissando.

The term singing out of tune can be defined as the situation in which a note pitch differs annoyingly from the tuning of the other notes within a performance [10]. According to listening test results, the mean phonation frequency may deviate ± 0.07 semitones from the nominal note pitch, and they will still be generally judged to be in tune [11]. In addition, even a deviation larger than ± 0.2 semitones was acceptable in certain situations, either when the phonation frequency was sharp rather than flat, at unstressed metrical positions or during tragic song mood. Another type of tuning problem is the drifting of the tuning in a cappella performances, as we already pointed in the introduction chapter.

In vocal ensemble singing it is less frequent to apply a significant amount of vibrato, tremolo or glissando. Intended expressiveness in the vocal performance is more sophisticated in comparison to solo singing. Instead, usually vocal ensembles strive for a perfect harmony, and the expressiveness of a cappella vocal ensemble music comes from clean and accurate intonation, which we discuss next.

2.3 Intonation in vocal ensemble singing

Before proceeding, let us revise the terms tuning, temperament and intonation. Tuning can be defined as the system which is used to tune instruments. It comes down to the question of how to divide the octave into a musical scale. Temperament can be defined as a variation to a tuning system. Intonation on the other hand can be defined as the practice of musicians to alter tuning during performance. Nevertheless, these three terms seem to be used in various ways, meaning more or less the same thing. The logarithmically equally divided twelve-tone scale can be called either equal tuning, equal temperament or even equal intonation. The intonation practice of a vocal ensemble can be referred as intonation or tuning in the same way. There is no major difference between these three terms, and the exact meaning usually becomes clear from the context.

Equal tuning or temperament has become a widely adapted standard in music. It allows an instrument to sound equally good in any key, because all semitones have an equal logarithmic ratio. Periodic instruments are usually tempered to sound better in a selected key. For example, a piano can be played in any key, whereas a harpsichord is separately tuned for a selected key using some tempered scale. Our ear has adapted to equal tuning as the right intonation, because it is so widely in use.

Musical consonance on the other hand is derived from the sharing of partial frequencies. When the fundamental frequencies of two notes are in a simple integer ratio, they are consonant. Nevertheless, building a scale in which all intervals are maximally consonant turns out to be impossible [17]. There must be made some compromise. The temperament or tuning system being used is defined by the way in which this compromise takes place.

Since the human voice is not locked into a single tuning system or temperament, intonation in singing is flexible and at the same time demanding. The exact tuning of each pitch may vary each time it is sounded [18]. It turns out that the tuning of a vocal ensemble cannot consistently be related to a single reference point like instruments with fixed tuning, for example the piano. The tuning of a vocal ensemble is rather a combination of horizontal and vertical musical factors from the reference tone [19, 20]. There is a strong probability that the weightings of the horizontal and vertical factors differ in different musical contexts [18]. This is why the relation between the these factors is complicated.

Music performers rely on very specific and practical rules what comes to intonation. According to informal interviews with three vocal ensemble music professionals, in principle, practical intonation in a cappella vocal ensemble music is done according to just intonation. Nevertheless, there are several exceptions. For Western tonal

classical a cappella music, there are some practical rules, which vocalists apply in their performance. One example is the seventh tone of a major or minor scale, which is a so-called leading tone. According to the convention, leading tones should always be intonated particularly sharp. On the other hand Barbershop quartets specifically leave the same seventh tone particularly flat, according to the harmonic seventh with the ratio of 7:4, which gives a distinguishable colour to this particular niche genre of a cappella vocal ensemble music. Atonal music is typically practiced and performed according to equal temperament. In addition, early music has its completely own rules of intonation.

The theory about tuning and intonation practices in Western music go up to the ancient Greece and Pythagoras, and comes all the way to these days. Each period in the music history has had its own musical needs, and the theory about tuning and intonation has evolved accordingly [21]. Another affecting factor has been the evolution of science and mathematics. It is no coincidence that the establishment of the equal tempered logarithmic scale and the discovery of the logarithmic function itself both take place in the seventeenth century. Contemporary work has shifted more from practical tuning and intonation practices towards finding universal theories and tendencies for explaining the phenomena of intonation in music.

According to Devaney and Ellis in a study on intonation tendencies in polyphonic vocal performances [18], the major issue in theorising intonation practices is the conflict between vertical and horizontal tendencies at any given point in time. The assumption is that vertical intonation tendencies will conform to the overtone series, and it is rooted in Helmholtz's theory of consonance and dissonance [22]. According to this theory, the coincidence of a significant number of partials between two pitches produce a consonance, whereas the absence of such coincidence produce a dissonance.

For the vertical aspects of intonation, Helmholtz's theory [22] is the foundation of relevant psychoacoustical theories that support the idea that maximal coincidence of partials in a vertical sonority produces maximal consonance [18]. Other research scientists have later contributed to this work. Plomp and Levelt [23] found an interval size called the critical band, which is also a significant factor int the perception of consonance. Beating is an acoustical phenomenon produced by the interference between tones of proximate frequency. Terhardt [24] presented a theory of consonance, which first states that a greater degree of consonance corresponds to a lesser amount of beating. If the beating is slower than 20 Hz, it becomes audible. If it is faster than 20 Hz, it is perceived as roughness. The second component of the theory of consonance states that the perception of harmonic consonance is dependent on the harmonic context, the mind's acquisition of an acoustical template. According to the theory, the majority of this learning comes from exposure to the complex tones found in speech, and the sensation of consonance in Western music is psycho-acoustical rather than cultural.

This work can be applied according to Devaney and Ellis [18] to the issue of intonation preferences in vertical sonorities. They consider that singers are more likely to intonate by matching lower partials in the overtone series than to aim for equally tempered tones. They also state that these intonation tendencies emerge more commonly in sustained notes, since in quickly moving note sequences the horizontal

aspect is likely to become more dominant.

For the horizontal aspects of intonation, Devaney and Ellis [18] have studied several theories. According to Lerdahl's [25, 26] model of melodic attraction, which is a component of his tonal pitch space theory, a dissonant pitch has a tendency to resolve to a consonant neighbour. There is an analogy with Newton's law of gravitation: The attraction of one pitch to another is the anchoring strength of the goal pitch. Lerdahl discusses the asymmetries in attraction when moving from stable pitches to unstable ones and from unstable pitches to stable ones. Because of this asymmetry, the same interval works differently in different musical contexts. Another work on melodic forces by Larson [27] quantifies how listener's expectations are met or confounded by particular musical patterns. There are three magnitudes describing different musical tendencies: gravity, magnetism, inertia. Gravity is the tendency of a musical line to go down. Magnetism is the tendency of unstable notes to move to stable ones. Inertia is the tendency of a musical line to continue rather than vary.

Devaney and Ellis [18] also introduce studies on the expressive aspects of intonation. Horizontal intonation practices are the ones in particular that often function as expressive phenomena. Emotion can be conveyed by music performers through deviations from a norm [16]. Accordingly, the denial of the listener's expectations is where emotional responses to music are rooted in [28].

Although the attraction models could serve as a more appropriate reference than simply using values from an equal tempered system, as stated by Devaney and Ellis [18], this kind of model might be too advanced for the purpose of this thesis. It is more convenient to split the research question into two separate questions:

1. What is the right intonation, in other words the correct pitch of each singer in a vocal ensemble, at a given point of time, taken the musical context into account?

2. How do we estimate the intonation, in other words pitch, from a polyphonic music signal in order to measure the deviation between the expected intonation and the estimated intonation?

For the rest of this thesis we will focus on the latter research question: How to estimate the pitch from a polyphonic music signal and measure its deviation from a reference chord. To have some reference point for the measurement, we will consider using either equal temperament or just intonation. The advantage of using equal temperament is that the pitch of any tone can be calculated with on simple formula regardless of the musical context. The drawback of equal temperament is that it does not exactly produce simple integer ratio intervals with maximum consonance, which according to the studies are the closest approximation to real-life intonation practices. When investigating specific chords where the musical context is known, we will also consider using equal tuning as reference. The differences between equal tuning and just intonation are shown in table 1.

To conclude, intonation in a cappella vocal ensemble singing is a joint effect of both intended and unintended deviations from some theoretical reference tuning system. There is no straight-forward way to theorise practical intonation and form an

Table 1: Interval ratios in equal temperament and just intonation. The ratios are also represented in cents (c) from the tonic. The right column shows the deviations between the equally tempered and just intonated scale tones.

	Equal ratio	Equal [c]	Just ratio	Just [c]	Deviation [c]
Unison	$2^{0/12}$	0	1/1	0.00	0
Minor second	$2^{1/12}$	100	16/15	111.73	−11.73
Major second	$2^{2/12}$	200	9/8	203.91	−3.91
Minor third	$2^{3/12}$	300	6/5	315.64	−15.64
Major third	$2^{4/12}$	400	5/4	386.31	+13.69
Perfect fourth	$2^{5/12}$	500	4/3	498.04	+1.96
Tritone	$2^{6/12}$	600	7/5	582.51	+17.49
Perfect fifth	$2^{7/12}$	700	3/2	701.96	−1.96
Minor sixth	$2^{8/12}$	800	8/5	813.69	−13.69
Major sixth	$2^{9/12}$	900	5/3	884.36	+15.64
Minor seventh	$2^{10/12}$	1000	16/9	996.09	+3.91
Major seventh	$2^{11/12}$	1100	15/8	1088.27	+11.73
Octave	$2^{12/12}$	1200	2/1	1200.00	0

unambiguous set of rules for the performers, that would cover all possible situations and aspects of music. What we can do within the scope of this thesis is to design an auxiliary application that increases the vocal ensemble singers' awareness of their intonation and helps them monitor their pitch on an accurate and detailed level.

3 Signal processing foundations

This chapter covers the foundations of all audio signal processing applications, which are needed in order to implement the algorithm for automatic vocal ensemble intonation analysis. We will apply these methods to practice using the MATLAB programming language. MATLAB is a reasoned choice as a tool for experimenting with audio signal processing methods, since it offers a wide amount of pre-build audio processing functions and data visualisation tools. We will also take advantage of newly introduced real-time audio processing and graphical user interface building capabilities.

3.1 Digital sampling and quantisation

When real-world analog audio signal is captured with a microphone, before analysing, it is first converted from analog to a digital signal by performing an analog-to-digital conversion. In principle, we do not need to take care of this phase, since it is a built-in functionality in every modern laptop and mobile device. Most important is to understand the difference between real-life continuous analog signals and discrete digital signals. In digital-to-analog conversion, the continuous analog signal is sampled on the time axis and quantised on the amplitude axis to discrete-time sample values. A an example of a continuous and discrete sine wave is shown in figure 2.

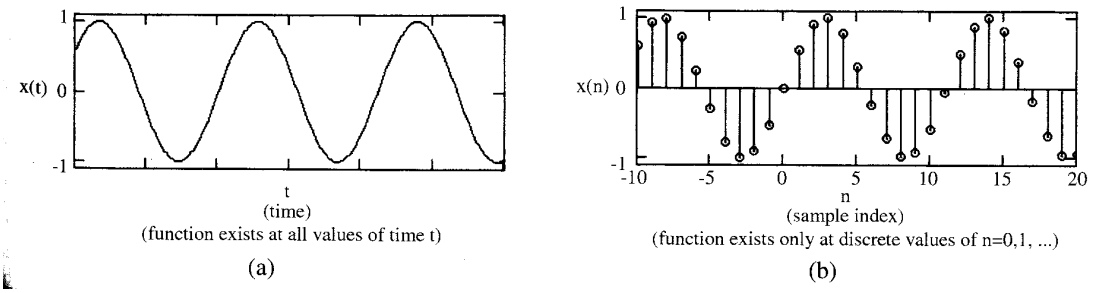

Figure 2: A continuous waveform (a) sampled to a discrete signal (b). To obtain the discrete signal, the continuous signal is sampled in the horizontal axis and quantised in the vertical axis. Adopted from [7].

Sampling frequency or sampling rate is the amount of samples taken from the continuos signal per second. The most typical sampling rates used in audio applications are 44100 Hz and 48000 Hz. When an audio signal is imported to the MATLAB environment, either from a file or directly from the microphone input, the sampling rate is also read and imported to MATLAB. The sampling rate must be taken into account in various audio signal analysis and processing tasks.

3.2 Time, frequency and time-frequency representations

Music signals can be represented, analysed, visualised and understood in two main domains, time and frequency [2]. In the time domain is the domain music signals

are recorded and played. Time flows on the horizontal axis, and the vertical axis represents air pressure, in other words amplitude. A digital signal is an ordered set of sample values representing the amplitude of a waveform at a sequence of instants in time [7].

The time domain shows temporal information, time data, how the signal changes over time, but for analysing musical information at a given point in time, we need to switch to the frequency domain. The frequency domain is where music signals can be represented an understood. It reveals information about pitch, harmony and timbre at a given time in a music signal.

Often we want to combine the temporal and the harmonic information to see how for example pitch, harmony or timbre changes over time. In this case we can use combined time-frequency representations [2]. A Western music score, like the one presented in figure 1, is actually a time-frequency representation, and notation is a specific encoding to represent combined time-frequency information.

In order to switch from the time domain to frequency domain, we need a mathematical representation for frequency. The frequency domain can be defined mathematically with the Fourier transform (FT) [2], which is represented in equation 4.

$$\mathrm{FT}_x(f) = X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \tag{4}$$

Here $x(t)$ is the signal in the time domain, and $X(f)$ the signal in the frequency domain.

The Fourier transform is defined for continuous signals. In digital signal processing signals are discrete instead of continuous. For discrete signals the equivalent representation of FT is the discrete Fourier transform (DFT), which is represented in equation 5.

$$\mathrm{DFT}_x(k) = X(k) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn}dt \tag{5}$$

Here x(n) is the discrete signal in the time domain, and X(k) the discrete signal in the frequency domain.

Both FT and DFT have inverse operations, the inverse Fourier transform (IFT) and the inverse discrete Fourier transform (IDFT), which can be used to switch from frequency to time domain. These are as well useful operations in music processing. For example, after modifying a music signal in the frequency domain it can be transformed to the time domain with the inverse Fourier transform for playback. In our research we only need to transform an input music signal from time to frequency domain for analysis, but there is no need to switch back to the time domain anymore after that.

When computing a DFT or IDFT, it is more efficient to implement a fast Fourier transform (FFT) algorithm rather than to use the DFT definition directly [29]. This reduces the algorithm complexity from $O(N^2)$ to $O(N \log N)$. There are numerous different FFT algorithms, from which the Cooley–Tukey algorithm is the most commonly used [30].

To achieve the highest efficiency of FFT algorithms, all N values of the DFT must be computed [29]. In some cases there can be a reasoning to compute only a portion the full frequency range, as we will later observe. There are other algorithms that can be used in these cases. They may be less efficient in computing the full frequency range as FFT algorithms, but more efficient and flexible for computing a partial range.

There are a couple of important properties regarding the discrete FT [2]. The first one is related to the Shannon theorem, which states that the range of frequencies where the discrete FT is meaninful has an upper limit given by $k_s/2$. This frequency is called the Nyquist frequency, and k_s is the sampling frequency. The other property is related to frequency resolution, which is even more relevant in terms of our application. The frequency resolution of a DFT is highly dependent on the number of samples used for the transform, in other words the time window length. The longer the time window, the higher is the frequency resolution. The shorter the time window, the lower is the frequency resolution.

When analysing a music signal, we are usually interested in what is happening at a specific moment in time. This is why frequency analysis is typically done in pieces for one short time frame at a time. A common duration for one frame is from 20 ms to 100 ms [2].

Frames are typically computed with a window, which are generally positive and symmetric [2]. Standard window shapes include the Gaussian, Hamming, Hanning, and rectangular windows. By using one of the first three windows, we eliminate the discontinuities at the beginning and end of the frame, which typically have a negative effect on the result of the FT. A rectangular window simply cuts the frame without modifying the signal in any way. Rectangular windows should not be used in principle, unless there is a specific reason for it.

A frame localised at time t_0 and computed with window w, is represented in equation 6.

$$s^w_{t_0}(t) = x(t)w(t_0 - t) \tag{6}$$

This equation can be applied to a longer signal as a sliding window by multiplying [2]. This way we get one windowed frame from a given point in time. When analysing a signal in frames, the signal at beginning and end of each frame becomes quite weak when windowing is applied. In order to prevent losing information, it is typical to apply some window overlapping.

We have covered the basic methods to represent both time and frequency domains and convert the signal from one domain to another. Often we want to combine both domains in order to create a time-frequency representation. The most popular time-frequency representation is the spectrogram [2]. It can be constructed from short-time Fourier transforms (STFT), which are Fourier transforms of successive windowed frames. The STFT is represented in equation 7.

$$\mathrm{STFT}^w_x(t, f) = \mathrm{FT}_{s^w_t}(f) = \int_{-\infty}^{\infty} x(\tau)w(t - \tau)e^{-j2\pi ft}d\tau \tag{7}$$

The spectrogram is an energy representation of the signal. It is defined as the squared

modulus of the STFT and represented in equation 8.

$$\mathrm{SP}_x^w(t,f) = |\mathrm{STFT}_x^w(t,f)|^2 \tag{8}$$

Each different window w defines a different spectrogram. By using different window lengths, we can represent either spectrograms with high frequency resolution and low time resolution, or spectrograms with low frequency resolution and high time resolution.

The following figures, 3, 4, 5, 6, and 7, show examples of the different representations covered in this subchapter. They all represent the same one-second sound sample, which is taken from a recording containing a vocal ensemble singing in a rehearsal. The sample contains only one relatively stable chord from a piece sung by the ensemble. The chord is a D flat minor sung by four male voices. The fundamental frequencies of the notes written in the score using equal temperament are 155,6 Hz, 233,1 Hz, 311,1 Hz, and 392,0 Hz from lowest to highest.

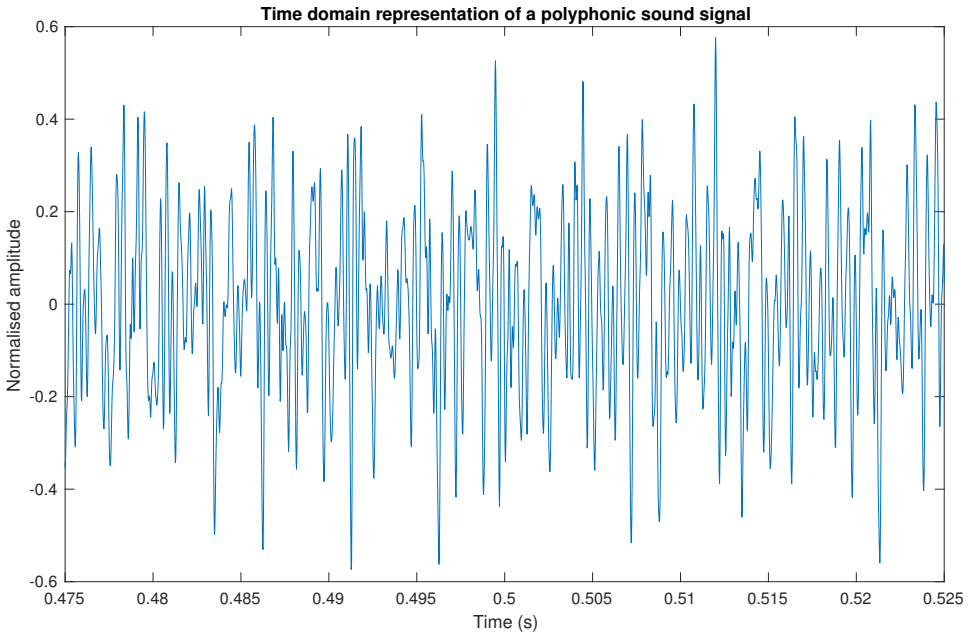

Figure 3: Time domain representation of a vocal ensemble singing one chord. The figure shows a 0.1 second clip of the whole one second sound sample.

Figure 3 represents the time domain. which does not give us much information regarding the harmonic content of the signal. Figure 4 represents a spectrogram with relatively high time resolution but low frequency resolution. This figure shows us the harmonic content of the signal over time. There vertical yellow lines are frequency peaks produced by four male singing voices. The lowest ones are the fundamental frequencies, and the higher ones are harmonic partials. These frequency peaks together represent the harmony created by the vocal ensemble, and correspond to the chord written in the music score.Figure 4 reveals that there is slight variation

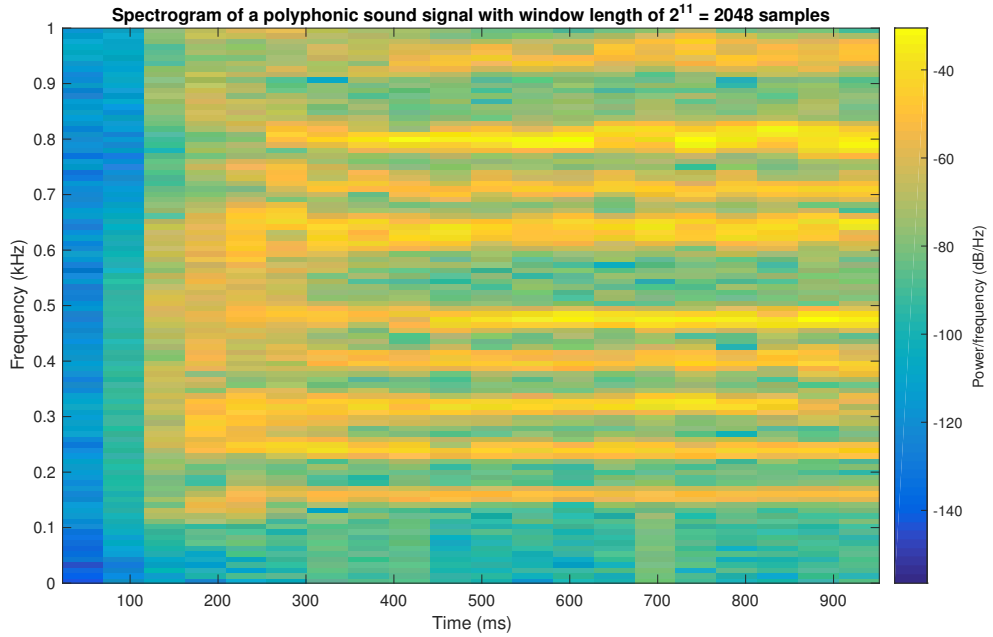

Figure 4: Time-frequency representation with high time resolution of a vocal ensemble singing one chord. The figure shows the frequency range from 0 to 1 kHz including all the fundamental frequencies and some of the lowest partials of each singer.

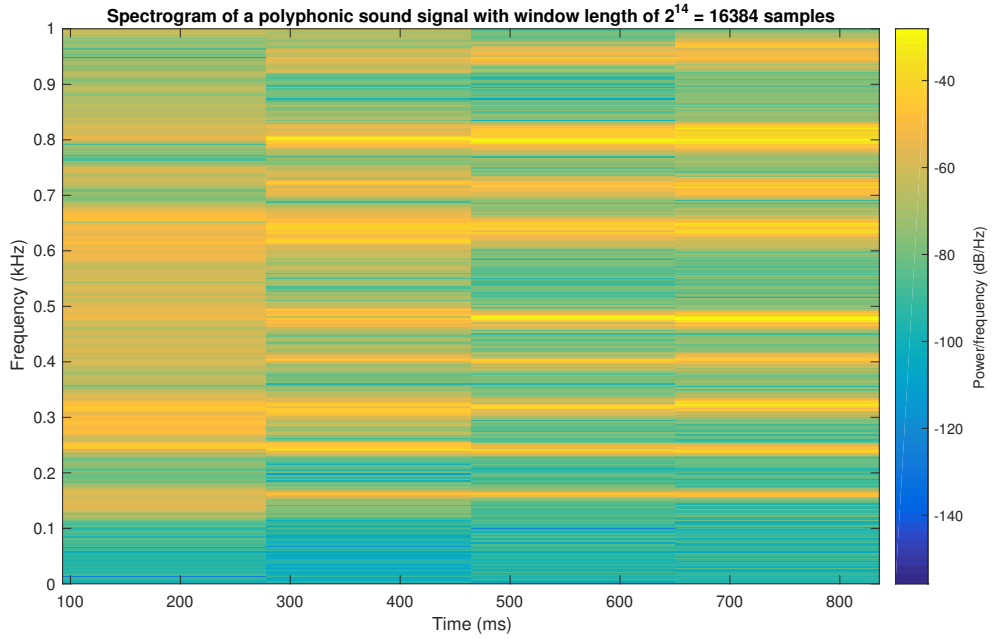

Figure 5: Time-frequency representation with high frequency resolution of a vocal ensemble singing one chord. The figure shows the frequency range from 0 to 1 kHz.

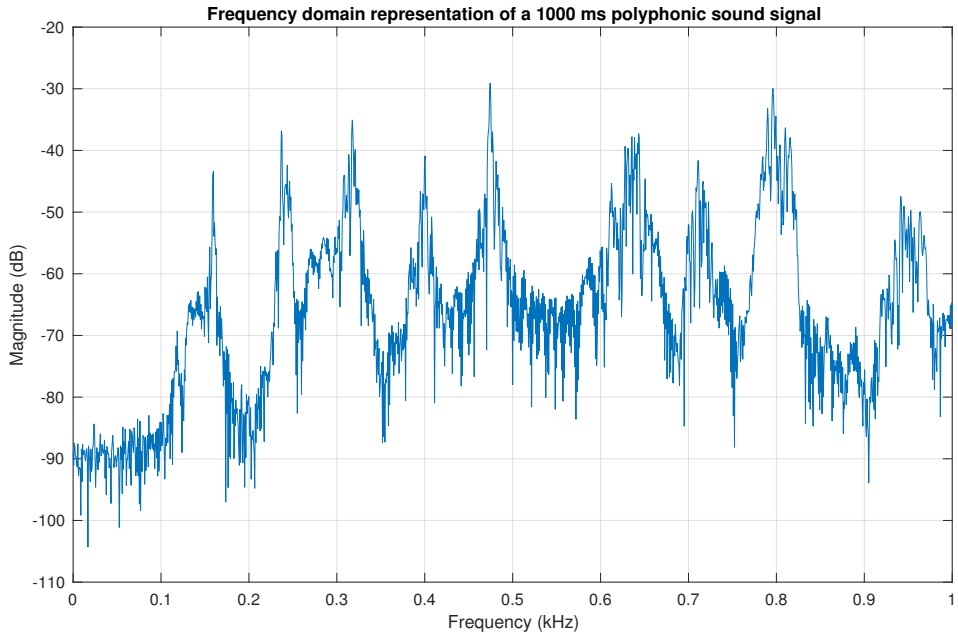

Figure 6: Frequency domain representation of a vocal ensemble singing one chord using a 1000 ms window. The figure shows the frequency range from 0 to 1 kHz including all the fundamental frequencies and some of the lowest partials.

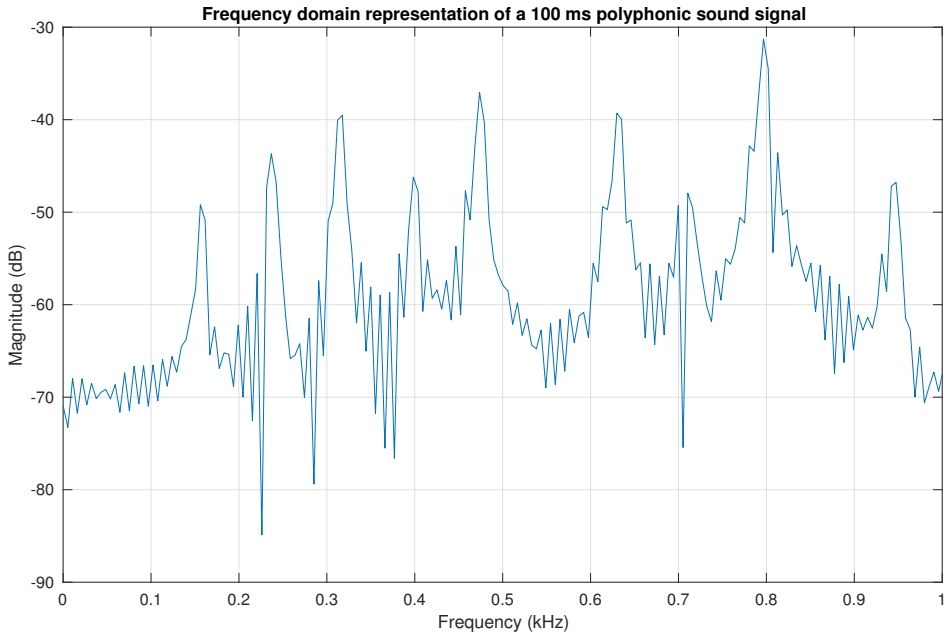

Figure 7: Frequency domain representation of a vocal ensemble singing one chord using a 100 ms window. The figure shows the frequency range from 0 to 1 kHz including all the fundamental frequencies and some of the lowest partials.

over time in the pitches sung by the ensemble, but the frequency estimates might be inaccurate due to the low frequency resolution. Figure 5 has a better frequency resolution, but the changes in the frequency domain over time are less visible than in figure 4 due to the low time resolution.

Figures 6 and 7 represent the frequency domain. Figure 6 contains the whole 1 second sound sample. It has excellent frequency resolution thanks to the long analysis window, but it does not give any information about changes over time during the 1 second period. Figure 7 represents a shorter 100 ms window of the sound sample. It gives more precise information about that specific moment in time, but it is also less accurate over the frequency domain.

3.3 Analysing music signals

This subchapter gathers some methods and aspects of signal processing, which are useful to understand when analysing music signals. In the earlier chapters we have covered the meaning of pitch, loudness, duration and timbre. In this thesis we are mainly dealing with pitch, but it is good to understand the presence of the other three features as part of the music signal at least on a high-level.

In music signal processing these features need to be encoded to a numerical format. Pitch, loudness and duration can quite naturally be encoded into a single scalar value [2]. Timbre on the other hand cannot be explained by a single acoustic property. It rather depends on the coarse spectral energy distribution of a sound and its time evolution. It is a multidimensional concept and typically represented with as a feature vector.

In the previous subchapter we have covered the time, frequency and time-frequency representations for a music signal. These can be seen as mid-level data representations, which function as an interface for further analysis and facilitates the design of efficient algorithms for this purpose [2]. For us, the time-frequency and frequency representations provide information about the harmonic content produced by a vocal ensemble. We can visualise and investigate the signal in the frequency domain in various ways, and obtain information which will help us in designing a proper multiple frequency estimation method that suits our specific use case.

MATLAB provides us with several functions, which can be used to represent a sound sample in the time-frequency or frequency domain. The ones we will use in our experiments are *fft*, *periodogram*, *freqz* and *spectrogram*. The first three produce a frequency domain representation of the input signal. The first one, *fft*, computes the FFT algorithm. To present the frequency domain, it is convenient to express the level of sounds with their mean-square power and apply a logarithmic decibel scale to deal with the wide dynamic range of music signals [2]. The second one, *periodogram*, computes a periodogram power spectral density estimate. It also uses the same FFT algorithm, but unlike *fft* it can be used directly as a one-liner to perform and visualise a frequency analysis. The third option, *freqz*, returns the frequency response of digital filter, but it can as well be used to perform a frequency analysis of a sound sample. This function has a built-in property, which allows analysing only a subpart of the whole frequency range. The *spectrogram* returns a spectrogram of the input

signal.

Due to the logarithmic nature of sound, pitch changes much faster relative to frequency on low notes compared to high notes. For example, the difference between the notes F2 and F#2, a semitone in the range of a bass singer, a low male voice, is only 5,19 Hz. For comparison, the difference between the notes F5 and F#5 in the soprano range, a high female voice, is 41,5 Hz.

The basic unit in the frequency domain is hertz (Hz). In our application we are interested in small pitch variations, which are significantly smaller than one semitone. Since the frequency domain is logarithmic in relation to pitch, it is not very informative to use frequency values in hertz to compare intonation at different pitch levels. Instead, we can use cents to express deviations relative to a given reference pitch. This can be calculated with the following equation:

$$f_c = 1200 * \log_2 \frac{f}{F_r} \tag{9}$$

Here f refers to the frequency in hertz, which we want to express as a deviation in cents, and F_r is the reference frequency to which we are comparing.

In order to change back from cents to a frequency value in hertz, we can use the following equation:

$$f = F_r * 2^{F_c/1200} \tag{10}$$

Here f_c refers to the deviation in cents, which we want to express as a frequency value in hertz, and F_r is the reference tone to which we are comparing.

Using equation 9 we can convert the previous intervals to cents. The distance from F2 to F#2, and from F5 to F#5 is both 100 cents. These results are directly comparable with each other. Similarly, using equation 10 we get the frequency at a given point in cents relative to the reference frequency.

Since the frequency resolution of short time windows are limited, we might encounter precision issues. As we saw in the previous example, a semitone change in pitch is only a few hertz. We can expect resolution issues especially at these lower frequencies that correspond to the vocal range of bass singers, in other words low male voices. One possible way to increase precision is to apply interpolation. This will at least theoretically give a more precise location for the frequency peak, as shown in figure 8.

The interpolation in figure 8 is done using 1 Hz resolution. It is a spline interpolation, where the interpolated value at a query point is based on a cubic interpolation of the values at neighbouring grid points in each respective dimension [31].

Another useful representation for pitch in music applications, apart from the cent scale, is the MIDI format. The pitch of a given note in the equally tempered scale can be represented as a MIDI note number. A fundamental frequency value in hertz can be converted to a MIDI note number and vice versa using the following equations:

$$M = 69 + 12 \log_2 F_0 \tag{11}$$

$$F_0 = 440 * 2^{(M-69)/12} \tag{12}$$

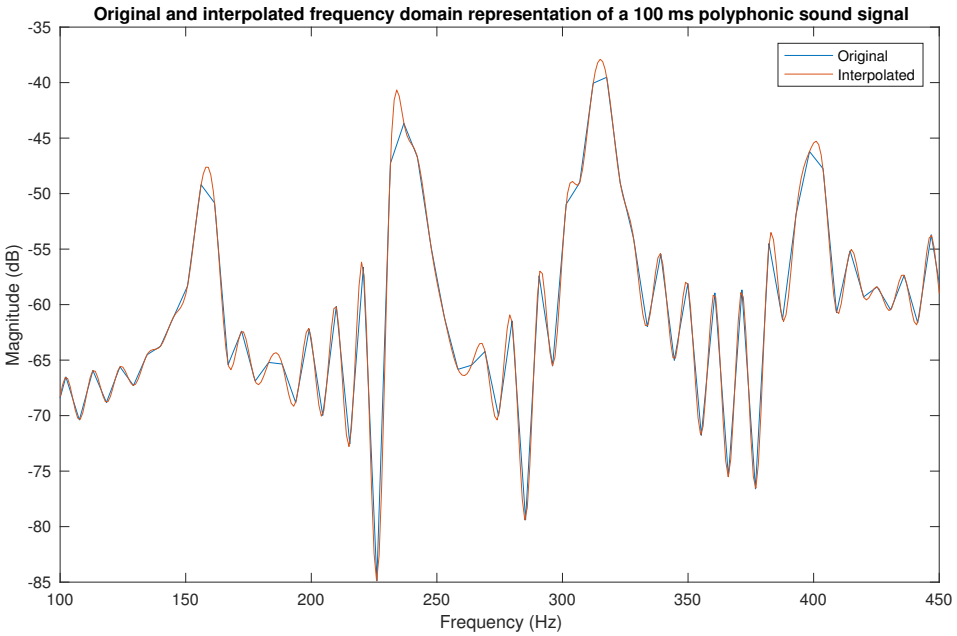

Figure 8: A closeup of the same frequency domain representation using a 100 ms window, that was shown in figure 7, including an interpolation of the signal.

In these equations M refers to the MIDI note number and F_0 to the fundamental frequency of the note. The reference tone is A4 = 440 Hz, which corresponds to the MIDI note number 69. Another reference tone could also be used, but this is the most usual. By replacing A4 = 440 Hz with for example A = 415 Hz, which is the reference tone used in early music performances, or with A4 = 442 Hz, which is the reference tune used in various orchestras, we could apply these equations to other tunings as well.

Figure 9 shows the logarithmic nature of musical tones. The lower we are at the scale, the smaller is the frequency difference between consecutive two tones. The lowest notes in vocal ensemble music are around C2, which corresponds to MIDI number 36. The highest notes are around C6, which corresponds to MIDI number 84.

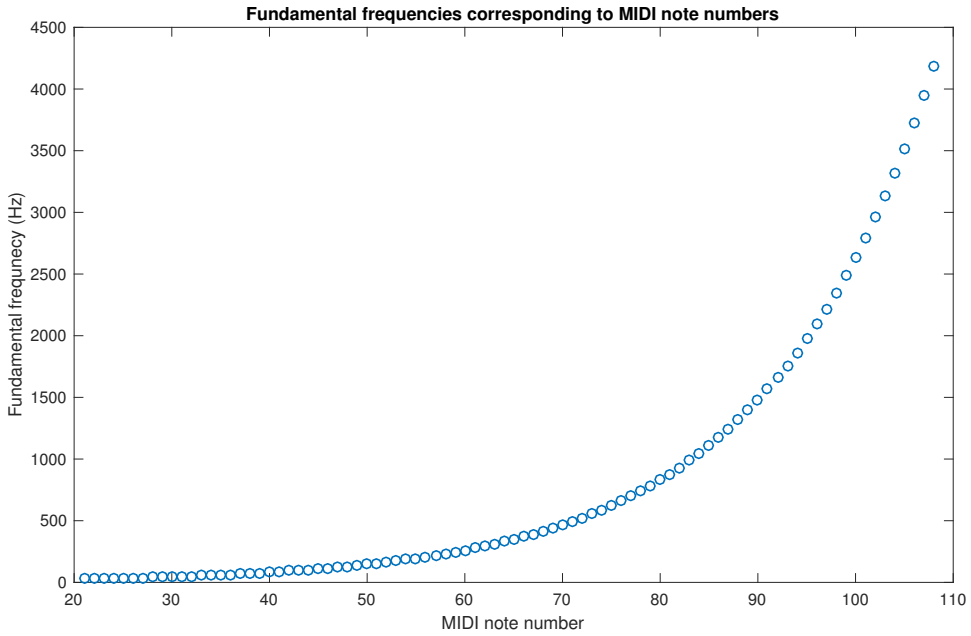

Figure 9: Fundamental frequencies corresponding to MIDI note numbers. The range of the figure corresponds to the range of a standard piano keyboard, which is from MIDI number 21 to 108. The corresponding range in note labels is from A0 to C8.

4 Related research and applications

This chapter covers research and applications related to our research topic. We start with the fields of signal processing for music analysis and automatic music transcription. In addition, we review a couple of interesting studies, which involve the analysis of very small pitch variations in a polyphonic music signal. The other one is a study on tuning classification, and the other on intonation in vocal ensemble singing, like our task is. Finally, we review one study on efficient pitch detection techniques for a real-time application.

4.1 Automatic music analysis and transcription

Muller et al. [3] provide an excellent overview of various signal processing techniques that specifically address musical dimensions, such as melody, harmony, rhythm, and timbre. Like in our research topic, this study focuses on analysis rather than synthesis, which is another wide music signal processing topic. Synthesis and analysis are inverse problems. While synthesis focused in constructing synthetic audio signals based on a model or other abstract description like a musical score, analysis is the problem of recovering a score-level or other abstract description *from* an audio signal.

A lot of emphasis in this overview is put on the key properties of music signals, that give rise to the techniques of music signal processing [3]. We have already discussed a variety of these properties in this thesis. Muller et al. describe pitches as the pre-eminence of distinct fundamental periodicities, polyphony as the preponderance of overlapping sound sources in musical ensembles, timbre as the variety of source characteristics, and beats as the regular hierarchy of temporal structures. Many of the automatic music analysis techniques are initially borrowed from speech processing or other areas of signal processing, but the unique properties and stringent demands of music signals have dictated, that simple repurposing is not enough. This has led to some inspired and elegant solutions.

According to Muller et al. [3], a major challenge in the automatic analysis of music signals is the ubiquity of simultaneous pitches, with coincident or near-coincident harmonics. In other words this means polyphony, and it is exactly what we need to deal with in our research on automatic vocal ensemble analysis. The worst case, according to this overview, is when two simultaneous notes are played one or several octaves apart.

There are two main strategies for dealing with polyphony, as described by Muller et al. [3]. The other is to process the signal globally, directly extracting information from the polyphonic signal. The other strategy is to attempt to first split up the signal into individual components, in other words sources, that can then be individually processed as monophonic signals. In source separation the goal is to extract all individual sources from a mixed signal. In musical context, this translates in obtaining the individual track of each instrument, or in our case of vocal ensemble singing the individual parts sung by each member of the ensemble.

Most multiple fundamental frequency estimation approaches work in the spectral domain, at least partially [3]. Some of the methods following the global strategy

aiming at jointly estimating all fundamental frequencies include parametric methods, a dedicated comb approach, methods based on machine learning paradigms, and the least-square strategy. Methods relying on source separation principles include the matrix factorisation framework and iterative procedure.

Different algorithms for multiple fundamental frequency estimation are regularly evaluated and compared directly in the MIREX evaluation campaign[4] [3]. Despite the inherent limitations of iterative procedures, they are regularly shown to be among the most efficient to date.

Muller et al. [3] also suggest using timbre information as prior information to better separate the musical sources. This method could be applied for source separation of different instruments. For our case however it might not be efficient, since in singing different phonemes have different spectra, which means that timbre at least on the lower partials change in relation to the phoneme.

As a final conclusion, Muller et al. [3] raise better understanding in auditory perception of sound mixtures as a major area for future advancements in music audio analysis. An interesting question related to this area is why a trained musician has no problem in analysing a chord containing two notes one octave apart. Better understanding in which way two overlapping partials interact and how their amplitude can be precisely estimated plays a key role in solving the octave problem.

Music transcription refers to the analysis of an acoustic musical signal so that it can be written down as the pitch, onset time, duration, and source of each sound event in it [2]. Transcription can be complete or partial. Complete transcription can be very hard or sometimes even theoretically impossible. For this reason the goal is usually refined as to notate as many of the constituent sounds as possible, or to transcribe only some well-defined part of the music signal, like the most prominent drum sounds of the dominant melody. Our study is related to partial transcription, since we only need to transcribe the pitch from a polyphonic music signal.

Music transcription is closely related to the field of structured audio coding [2]. Music perception is another related area of study. In addition, music transcription can be applied to many use cases including music information retrieval based on the melody of a piece, score type-setting programs, musically oriented computer games, music-related equipment, ranging from music-synchronous light effects to paniment for a soloist, and transcription tools for amateur musicians who wish to play along with their favourite music, to name a few. Our application can also be defined as an applied transcription task, although our aim is to detect small variations of pitch rather than annotate music into notation.

Music transcription is generally laid on the low-level signal analysis where sound events are detected and their parameters are estimated [2]. This provides us with note data. Subsequent processing can be applied on the note data to obtain larger musical structures.

Automatic music transcription can be compared with real human music transcription. When an average listener listens to a musical piece, she is able to perceive a lot of musically relevant information [2]. Typically an average listener is able to

[4] http://www.music-ir.org/mirex/wiki/

tap rhythm, hum melody and recognise musical instruments. On the other hand, harmonic changes and various details are perceived less consciously.

Similarly to natural language, learning to read and write music requires education [2]. A trained musician has studied not only notation, but also recognising different pitch intervals and timing. These properties need first to be encoded into a symbolic form in the mind of the listener, before being able to write them down.

An untrained listener is typically not able to hear the inner lines in music, in other words, sub-melodies other than the dominant one [2]. In order to develop an analytic mode in listening where and be able to distinguish inner lines, musical ear training is needed. As the polyphonic complexity of a musical composition becomes richer and richer, the more musical ear training and knowledge of the particular musical style and of the playing techniques of the instruments involved is needed from a human listener to be able to perform the transcription task.

At the moment, state-of-the-art music transcription systems are still clearly inferior in comparison to skilled human musicians, both in accuracy and flexibility [2]. Typical restricstions in the transcription of pitched instruments include limited number of concurrent sounds, no interference of drums and percussive sounds, or poorly specified instruments. In our case the instrument and its properties, the singing voice, is well known. We can assume that there are no percussive sounds in the signal. The number of voices in the test material used in this study is four, but the size of a cappella vocal ensembles typically range from three to eight.

There is an open source application available for viewing and analysing the contents of music audio files, called Sonic Visualiser[5]. It has been developed at the Centre for Digital Music of Queen Mary University of London. A screenshot of the user interface is represented in figure 10. The architecture of the application allows easy integration of third party algorithms in the form of VAMP plugins, which can be used for the extraction of low and mid-level features from musical audio data. There are a couple of multiple fundamental frequency algorithm implementations available in the VAMP plugin library.

Based on an informal test, transcribing a vocal ensemble rehearsal recording accurately using a generic multiple fundamental frequency estimation algorithm provides fairly poor results. The transcribed and synthesised version was almost not recognisable as the original piece. This only demonstrates how challenging the transcription task really is. In order to get results from this study, we need to scope the analysis task to an even smaller and more specifically defined than generic multiple fundamental frequency estimation.

4.2 Automatic tuning classification and intonation tendency studies

Tidhar et al. [17] have made a study on automatic harpsichord tuning classification. Based on their conclusions, existing high-precision pitch estimation techniques are sufficient for the task. This study is particularly interesting from our perspective,

[5] http://www.sonicvisualiser.org/

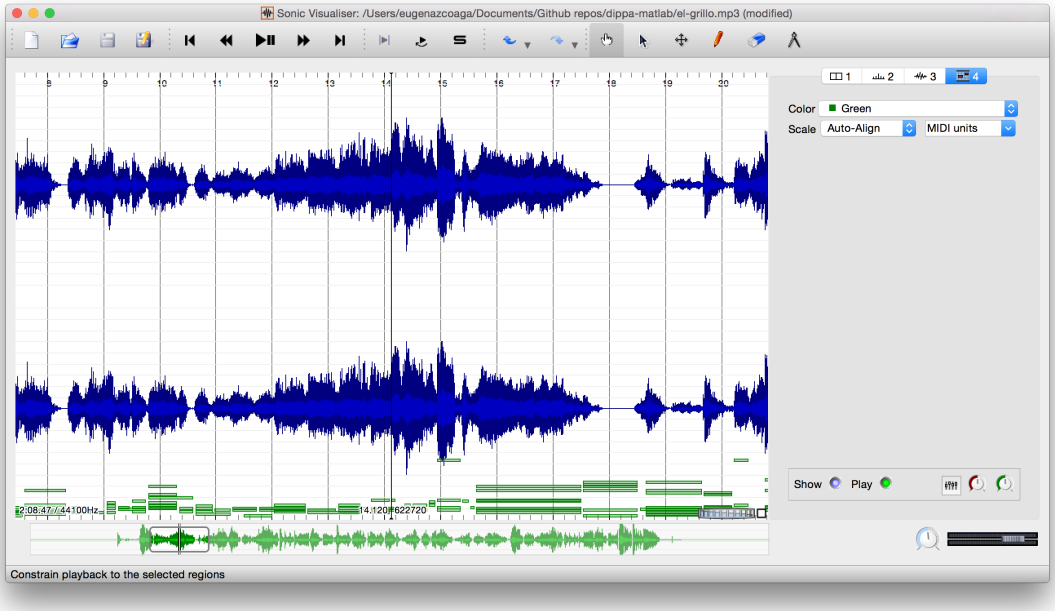

Figure 10: User interface of Sonic Visualiser, an open source audio and music analysis application. A polyphonic transcription VAMP plugin provided by the Queen Mary University of London is applied to a rehearsal recording of a vocal ensemble. The green rectangles on the bottom of the white area of the screen represent the estimated tones in MIDI units.

since tuning classification requires high precision in the same way as our automatic intonation analysis task does.

According to Tidhar et al. [17], building a system to classify musical recordings by temperament presents three particular signal processing challenges. The first one is, that differences between temperaments are small, of the order of a few cents. A window of several seconds duration would be required in a spectrum. This on the other hand introduces time resolution problems, since musical notes are not stationary and generally do not last this long. The second challenge is that notes rarely occur in isolation in musical recordings. There are almost always multiple notes sounding simultaneously, in other words polyphony. This has the potential to bias any frequency estimates. In addition, in music favoured intervals are the ones where the harmonic partials coincide. The third challenge is not knowing when each note is played. When detecting a sinusoid, it is not certain whether it is fundamental frequency or a partial of another fundamental. The ability to distinguish between these cases is crucial to successful temperament classification.

Tidhar et al. [17] summarize their framework of the classification task as follows: "Given an audio recording of an unknown musical piece, we assume the instrument is tuned according to one of the temperaments mentioned above, and that we know the approximate standard tuning (A=440 Hz, A=415 Hz, etc.), but we allow for minor deviations from this nominal tuning frequency or from the temperament." They have

defined six different temperaments from which they choose one for each analysed piece.

The initial dataset for the experiment was produced specifically for this purpose, since obtaining ground-truth data for a temperament-recognition experiment proved to be non-trivial [17]. The reasons for this were that most of the commercially available recordings do not specify the temperament, and those that do might not be completely reliable because of a possible discrepancy between tuning as a practical matter and tuning as a theoretical conduct. The dataset consists of both real and synthesised recordings. The tracks were chosen in a way that various degrees of polyphony, various degrees of chromaticism, as well as various speeds are well represented.

The method chosen by Tidhar et al. [17] is a two-stage approach. First they apply a concept they have named conservative transcription. It is a specific music transcription method designed for this specific purpose, and it results in high precision and low recall. It only identifies the subset of notes which are easily detected. This first step is followed by step two, an accurate frequency domain pitch estimation step for the notes determined in the first stage.

The reason Tidhar et al. [17] did not consider time-domain pitch estimation methods such as the autocorrelation function (ACF) and YIN, another widely used method based on autocorrelation, is, that they are unsuitable due to the bias caused by the presence of multiple simultaneous tones. In other words, these time-domain methods work only for monophonic signals. Instead, they focus on three frequency domain techniques:

1. Quadratic interpolated FFT (QIFFT)
2. QIFFT with correction for the bias of the window function (CQIFFT)
3. Instantaneous frequency calculated with the phase vocoder (PV)

More advanced estimation algorithms which admit frequency and/or amplitude modulation the authors deemed unnecessary in this study.

Each note object given by the conservative transcription gets a pitch estimate, which is generated from the frequency analysis [17]. The QIFFT is the estimate computed in the conservative transcription step, which was obtained from the signal downsampled to $f_s = 11025$ Hz using STFT and a Hamming window, a frame length of 4096 samples (370 ms), a hop size of 256 samples (23 ms, in other words 15/16 overlap), and zero padding factor of 2. The QIFFT estimate is computed for each frame in the note, and the pitch of the tone is calculated as the mean. In the second technique, the CQIFFT estimate was computed with no downsampling, a Blackman-Harris window, support size of 4096 samples, hop size of 1024 sample, and zero padding factor of 4. A a bias correction was applied on the window shape and zero padding factor after quadratic interpolation. The frequency of the first 12 partials for each note were estimated.

Various avenues lie open according to Tidhar et al. [17]. As future work they suggest that the conservative transcription step could be replaced with score alignment.

Another of their ideas is to perform statistical analysis of the data, which would allow them to evaluate the reliability of pitch estimates and then use them as weights in the classification step. Comparing the results obtained from these estimation experiments with estimates of human test subjects was left out of this study, but the authors think it is unlikely that a human expert could reach this level of accuracy.

There are many similarities between this study on harpsichord tuning classification and our task. First of all, high precision is required both in intonation analysis and tuning classification. In both cases the music signal is polyphonic. We will also apply frequency domain analysis in our experiments.

There are many differences as well. In the tuning classification task, the whole piece can be used at once for the analysis. Our aim is to perform analysis in real time, and in all the cases we need to display the estimate to the user at any given time, for each and chord, and each and every tone at a time. For this reason the conservative transcription method presented in the tuning classification study is out of the question in our case. Our advantage is that we define the reference chord as known information for the analysis task. In this sense we do not need to replace the conservative transcription step of the tuning classification with anything, since the notes are already given, and there is no actual actual need to the transcription from the signal to a note.

As explained in the study, the harpsichord is slightly inharmonic [17]. In other words the partials are slightly inharmonic in comparison to each other. Our advantage is that the singing voice is purely harmonic. The harmonic partials of the singing voice are purely harmonic, and we do not need to introduce any inharmonicity factor in our signal model. The harpsichord on the other hand produces relatively stable tones with clear onsets. The challenge with the singing voice is that it does not have as clear note onsets and offsets as the harpsichord. In addition singing tones may include portamento, glissando, vibrato, and other features that are not present in harpsichord sound signals.

Devaney and Ellis [18] have studied intonation tendencies in recorded polyphonic vocal performances using an empirical approach. Their aim has been to analyse intonation tendencies by statistically analysing large amounts of recorded data, and then compare the results with hypotheses based on the intonation theories we already covered in chapter 2. They use twelve-tone equal temperament as a reference because it remains consistent in spite of changes to the tuning references. Another reasoning stated by them is that it is also the standard system used both by music software and in Western art music practices as a whole, making it the most universal of any available reference points.

Devaney and Ellis [18] define the problem setting exactly the same way as it is in our study. They need to extract the pitch of multiple voices in polyphonic contexts. They can exploit prior knowledge of the intended, in other words notated, pitch sequence. They are interested in recovering a single, effective perceived pitch per note, and they want to be able to measure tuning differences in that pitch that may be far smaller than a semitone.

While we will use the reference chord from the score as known information, and Tidhar et al. uses their conservative transcription approach to determine which

notes are played, Devaney and Ellis [18] align a MIDI score of the work to the audio recording as a first step. This is not an entirely trivial task. The main challenge of using score information like a MIDI version of the piece as a frequency template to guide the pitch estimation is that of temporal alignment.

Devaney and Ellis [18] identify three main challenges in aligning polyphonic vocal recordings. First, the note onsets are often difficult to determine, particularly when notes change under a single syllable. Second, the very nature of a vocal ensemble means that all of the parts have roughly the same timbre, making it more difficult to distinguish between parts. Third, there is often a considerable amount of reverberation present in the recordings.

The second step, after the score has been successfully aligned, is developing a method to accurately extract pitch data from the polyphonic vocal recordings. Devaney and Ellis [18] describe this step as the main obstacle in their study. They also point another potential issue, which is when there are several voices singing a single part. Nevertheless, they state to have demonstrated successful pitch extraction for data with multiple voices per part, although the signal processing implications of this are not well understood.

To obtain the high-resolution estimate of the perceived pitch, Devaney and Ellis [18] use instantaneous-frequency (IF) based spectral analysis, which calculates a phase derivative, in other words instantaneous frequency, within each time-frequency cell of a conventional short-time Fourier transform [32]. This method gives an estimate of the frequency of a sinusoid within each cell whose resolution is not limited to the quantised bin centre frequencies of the underlying Fourier transform. The estimated energy and frequency of sinusoids at every time-frequency cell is represented as the IF spectrogram.

The weakness of this method is that the amount of other energy that may be present in the bin, whether its noise or the harmonics of other tones present in the signal, limits the the accuracy of this estimate [18]. Nevertheless, with the parameters the authors report successful results using this method with their example material of vocal ensemble recordings. They report using 100 ms windows with 75 % overlap, and a criterion that the IF in three spectrally-adjacent cells must differ by less than 25 % to reject cells that are not dominated by a single sinusoid, since in these cases IF estimates will not be stable. After this, a matching procedure looks for sinusoid components within the time-span indicated by the aligned score, whose frequency is close to the notated pitch. There is a matching score, which descreases as the observed frequency moves away from the expected pitch.

To estimate the single pitch value, Devaney and Ellis [18] use an energy-weighted average of the instantaneous frequencies aligned to each note. The authors find this simple averaging a reasonable approximation to the perceived pitch [33], but also see it as potential weakness in their approach in the presence of vibrato, since pitch variation within each note can be far larger than the tuning nuances being measured.

Devaney and Ellis [18] identify only the fundamental harmonic of each note is, and then use it for the pitch estimate, which according to them is valid, since the spectrum of the voice is purely harmonic. We intend to experiment with multiple partial harmonics in order to increase the robustness and reliability of our algorithm.

4.3 Real-time pitch tracking

De la Cuadra et al. [34] examined several pitch detection algorithms for the use in interactive computer-music performance. Regarding our work, the interest in this research is that that they define criteria necessary for successful pitch tracking in real-time, and evaluate four tracking techniques from this perspective.

The requirements for real-time algorithms, as defined by de la Cuadra et al. [34], are the ability to function in real time, minimal output delay, in other words latency, accuracy in the presence of noise, and sensitivity to musical requirements of the performance. The authors state that a pitch tracker designed for non-real time applications will not be successful when applied to interactive music if it cannot satisfy these requirements. The main reason is that while in recorded audio pitch-tracking errors can be cleaned-up with more robust secondary processing of the pitch-tracking data, in real-time applications there is not much room for initial pitch-tracking errors.

De la Cuadra et al. [34] discuss the aspects of real-time application, like how much error checking can be added to the system while still allowing the algorithm to run in real-time, and that some processor time must be left over for using the resulting pitch measurements. Aside from minimising pitch errors, the authors state that in a pitch detection algorithm the next most critical factor to consider is latency. There is attack latency and pitch identification latency. It takes the ear at least seven cycles to accurately identify a pitch, once a note is played.

The following general criteria is established by de la Cuadra et al. [34] to cover the most common characteristics of an interactive music environment needing pitch tracking. Frequency resolution needs to be at least semi-tones, including the correct octave. Recognition and quality of instantaneous pitch for possible real-time conversion into symbolic pitch needs to be timely. Instruments should produce with well-behaved harmonics, such as cello and flute. These do not completely match with the characteristics of our application. The main difference is that we need to be able to achieve much higher frequency resolution than a semitone. The similarity is that we also need timely conversion into symbolic pitch. The singing voice does produce purely harmonic partials, but pitch may vary greatly as the function of time, as we have already discussed.

The four tracking techniques surveyed by de la Cuadra et al. [34] are harmonic product spectrum (HPS), cepstrum-biased HPS (CBHPS), maximum likelihood (ML), and weighted autocorrelation function (WACF). The first three are based on the discrete Fourier transform, and the last is a time-domain technique. Frequency-domain algorithms are generally more robust than time-domain algorithms. DFT loses resolution at lower frequencies according to the authors. This makes our task particularly hard for estimating low tones in the bass register, as we will later see.

The most computationally efficient technique is HPS according to de la Cuadra et al. [34], and it can therefore afford the expense of zero padding. It is also the simplest method to implement, and it works well under a wide range of conditions. The primary drawback according to the authors is the need to enhance low frequency resolution with zero padding of the signal before transforming so that the spectrum can be interpolated to the nearest semi-tone. This wastes a lot of computational

power, because high frequencies are also being unnecessarily interpolated as well. However, the authors report that it it runs well in real-time tests using a 200 MHz processor.

ML considers the relative position of partials without the need to zero pad the signal frames and behaves well at low frequencies, as reported by de la Cuadra et al. [34]. it searches through a set of possible ideal spectra and chooses the one which best matches the shape of the input spectrum. The ideal spectrum is defined to be an impulse drain starting at a frequency and convolved with a signal window's spectrum. The algorithm tries to minimise the error between the spectral frame and possible candidate spectra. The efficiency of the algorithm depends on how much pitch resolution is required. According to the authors, it works well if the input source is in a fixed tuning. Keyboard and woodwind instruments are more appropriate than strings of voice since the latter instruments can easily produce non-discrete pitches, particularly in vibrato.

CBHPS is not as inexpensive as HPS, but needs less zero padding due to the enhanced low frequency resolution provided by the frequency-indexed cepstral component [34]. The cepstral analysis technique is possibly the most popular tracking method in speech analysis. According to the authors, it is good for multi-pitch detection, since it robustly handles noise and pitch errors.

WACF is an accurate but computationally inexpensive time-domain technique, although its resolution is limited by the sampling frequency [34]. Time-domain techniques are not considered for our multiple frequency estimation task, because in principle they only work for monophonic signals.

According to de la Cuadra et al. [34], pitch tracking in real-time situations usually involves additional steps beyond frame-by-frame pitch detection to enhance the quality of the measured pitch. The pitch detection algorithms presented above generate the instantaneous pitch for the input signal, but this will always contain some tracking errors. In particular, the resulting pitch measurement may be misleading if the input signal changes pitch during an analysis frame or there is any significant transient due to a note attack.

De la Cuadra et al. [34] provide some good observations regarding real-time pitch tracking applications. Pitch data can remain continuous or be converted into discrete symbolic form such as MIDI depending on the application. For instantaneous pitch control, it is usually sufficient to filter out spikes in the pitch data due to transients and octave errors. Depending on the noise tolerances of the application, it even may not be necessary to remove erroneous pitch. Monitoring the steady-state behaviour and amplitude of the detected pitch over several spectral frames is a good way to prevent accidental pitch assignment to transient regions. The signal frame overlap can be done at any arbitrary overlap factor, since the pitch detection algorithm does not involve re-synthesis of a signal. To avoid significant perceptual delay in a real-time application, the authors suggest to make correct new pitch updates need to be made within about 30-40 milliseconds. To recover from transients, an overlap factor at about 10 milliseconds per frame usually leaves enough time.

There is a major difference in our problem setting in comparison to the design principles of all these pitch tracking techniques. While our task is to detect small

intonation differences near to a pre-defined note, these methods are intended for transcribing unknown notes from a set covering the whole musical scale from the lowest to the highest possible tone. Nevertheless, the ideas related to real-time applications can be useful for our purpose.

4.4 Commercial applications

There are some commercial applications available, which are based on multi-pitch estimation. They can be roughly divided into transcription applications and tuning applications. This subchapter presents a selection of some of the most established products in these application areas.

Rocksmith[6] is a game by Ubisoft, one of the major console game companies. The game is played using a real electric guitar. The instrument is plugged into the computer of game console, and the game is capable of recognising which chords and tones the player is playing.

Melodyne[7] is a desktop software for professional music post-production. Among other things, it allows editing the pitch of individual tones in a polyphonic context. To achieve this, the software needs to perform full transcription of note pitches with high accuracy from a polyphonic audio signal.

Capo[8] is an application for Apple computers, iPads and iPhones, which recognises the chords of recorded music tracks and also performs isolation of instruments and vocals. For this purpose, the application needs to be capable of transcribing polyphonic signals at the level of accuracy that allows to estimate all the chords.

Polytune[9] is a polyphonic guitar tuner by TC Electronic, a well-known Danish audio equipment manufacturer. Traditional tuners are able to detect only one tone at a time. When tuning a guitar for example, each string needs to be played and tuned individually. The idea of polyphonic guitar tuning is that all the strings can be played simultaneously at once, and the tuner indicates which of them are flat, which are sharp, and which are in tune. Pitchblack Poly[10] is an equivalent polyphonic pedal tuner by Korg, another major audio equipment manufacturer.

[6] http://rocksmith.ubisoft.com/
[7] http://www.celemony.com/en/melodyne/what-is-melodyne
[8] http://supermegaultragroovy.com/products/capo/
[9] http://www.tcelectronic.com/polytune-2/
[10] http://www.korg.com/us/products/tuners/pitchblack_poly/

5 Application concept

In this chapter we develop a preliminary application concept for the purpose of understanding the intended practical use of our algorithm. We will then use this understanding to define the required functionality of the algorithm. It is also possible to build a working end-user prototype based on this application concept.

5.1 Intended use and end users

We have already discussed the topic of vocal ensemble intonation in practice and theory in chapters 1 and 2. The purpose of the application is to give accurate real-time information to a vocal ensemble about their intonation. In practice, we need to have a reference chord defined, then estimate the pitch of each singer in real time, calculate the deviation between the reference chord an the estimated multiple pitches, and finally visualise the result to the users.

This application will be primarily aimed for vocal ensembles where each part is performed by one single individual. This definition simplifies the problem significantly, both in terms of the end-user application and the algorithm itself. Another interesting topic would be to apply the same concept to a choir where each part is performed by multiple singers. At least two questions arise immediately about the case of choir intonation analysis: 1. How do we control the possible and very likely dispersion inside each part, when multiple singers are performing the same part in unison? 2. How do we implement the practical end-user application in order to give individual feedback for each and every singer in the choir? We first want to see good results with a small ensemble. After that it can be considered to start researching the particular problems related to choirs with a larger amount of members.

5.2 End-user application specification

First of all, our aim is to develop a user-friendly application that does not require custom hardware or heavy equipment. The end-user application should be possible to be implemented for a regular laptop, tablet or smartphone, which the majority of people already possess. The vocal ensemble should be able to use the application during a regular rehearsing session, without the need for any additional arrangements.

Since we are planning on using the built-in microphone of a regular device, we cannot count on the possibility of isolating the different parts of a polyphonic piece from each other. In addition, a regular rehearsing session requires the users to be in the same physical space and for to hear each other well while performing a polyphonic piece of music. This eliminates the possibility consider estimating each singer separately from an isolated monophonic signal. That is why we need to focus directly on multiple fundamental frequency estimation techniques.

The next open question is how to define the reference chord. What is the "correct" pitch that each singer in the ensemble should aim to sing at each given time? In our primarily case of classical Western vocal ensemble singing, this information is encoded into the sheet music and combined with the conventions of intonation practices for the

musical sub-genre of each given musical piece. In theory it would be possible to build a model that detects all the factors related to the musical context encoded to the sheet music and conventions about intonation practices, and intelligently formulates the reference chord for the vocal ensemble. This is a remarkably fascinating topic already alone, and could be investigated as a separate research question.

We will nevertheless keep the scope in how to estimate the multiple fundamental frequencies, calculate the deviation and show the result to the users. In principle, we will simply apply equal temperament to define the reference fundamental frequencies. An intonation analyser based on equal temperament is not able to guide singers in terms of small intonation nuances specific to the musical context, but it can serve to spot pain points where one or more singers consistently sing out of tune, or the key consistently tends to drift for the whole ensemble. This is already much more than what most of the a cappella vocal ensembles are able to accomplish without this kind of auxiliary application, as we have previously discussed.

This still leaves open the question about what are the reference tones within the precision of one semitone that the singers of the ensemble are supposed to aim to. One option would be to first automatically detect which chord the ensemble is singing, and then perform the intonation analysis. A complete transcription of a musical piece is known to be very hard or even theoretically impossible in some cases [2]. On the other, hand there is a lot of ongoing research on music transcription. There are plenty of techniques available, and we can expect many advances on specific aspects of music analysis in near future [3]. In order to keep our scope focused, and simply take the reference chord as a given input.

This is a very specific and well defined music analysis task, which is the very core of the intonation analysis of vocal ensembles. This task can be wrapped into a module with two inputs and one output, which then can be combined with other modules in order to build a complete end-user application. We will call this the intonation analyser module.

There are still several ways to combine the intonation analyser module with other modules in order to build the entire end-user application. The other modules define the complete functionality of the application. The reference chord for each given time can be inputted in many ways. The application can include a software keyboard, and the chord can be entered by the users manually. This would already beat the traditional training method where the vocal ensemble double-checks the intonation by playing each chord with a piano one by one, and try to subjectively determine if all the singers are singing the right pitch. A more sophisticated method would be to have the whole piece already inside the application, and the users could either automatically or manually advance chord by chord. The piece can be inputted in some standard format, like MIDI or MusicXML[11].

A user interface mockup for this kind of end-user application is shown in figure 11. In this application, the visualisation follows typical conventions from traditional sheet music, which are familiar to the target users. The parts are organised vertically from low to high, and time flows from left to right. The idea in this UI is that the

[11] `http://www.musicxml.com/`

chords move from right to left, and the line indicating which chord is active, remains in the horizontally centred on the screen. In the case of manually advancing though the whole piece, there would be a separate next button to continue to the next chord. In the automatic case there should be either a timer or a score alignment algorithm that is able to track the singers.

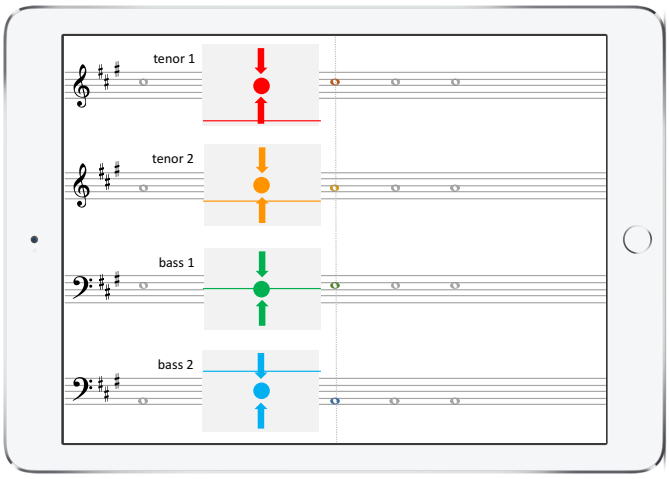

Figure 11: A mockup of the application concept. Sheet music moves on the staff under the tuning meters indicating the intonation. The intonation is indicated both with a vertically moving needle and a colour scale.

On top of the standard-notation-like UI there is one box for each voice containing a visualisation similar to a tuning meter. It indicates both with a vertically moving horizontal needle and a changing colour, what is the deviation of the individual singer from the reference note. The needle is specifically designed to move vertically, as it is a known convention in music to sing the tone too low (flat) or too high (sharp).

5.3 Intonation analyser module specification

The intonation analyser module takes in two inputs, a vector representing the fundamental frequencies of each note in the target chord from low to high in MIDI number format, and a vector representing the sound signal to be analysed. The output is a vector representing the deviations in cents between the fundamental frequencies sung by the singers estimated from the sound signal and the fundamental frequencies of the target chord. The module is illustrated in figure 12.

The MIDI number format for the input chord is a natural choice in this application. It is a universal and compatible format. If the user would for instance want to use a MIDI keyboard to input the chords, it would be straight-forward to start supporting this kind of compatibility. More importantly, sheet music in a digital format is based

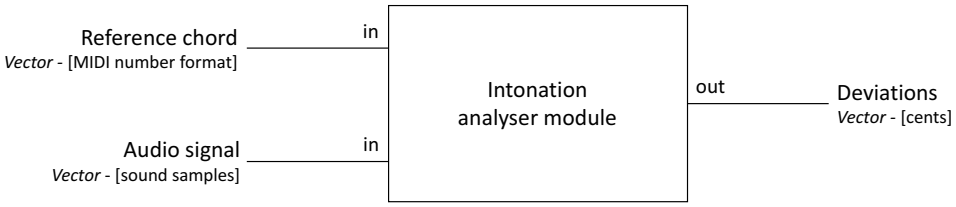

Figure 12: The intonation analyser module with two inputs and one output.

on the MIDI format. In order to read the input chords from a digital music sheet, MIDI is the most probable format. In addition, the MIDI format supports any type of tuning, temperament or intonation. Even though equal temperament is the default tuning system for MIDI numbers, another intonation can be expressed in cents, indicating how much flat or sharp the final tone is relative to the standard equally tempered MIDI tone.

This task is different from typical multiple fundamental frequency estimation, since we are interested in small deviations from known reference frequencies. Multiple fundamental frequency estimation is typically used for automatic music transcription, where the task is to match the estimated fundamental frequency with the nearest tone. Here we want to measure the deviation between the estimated fundamental frequency and the written tone.

In this case the task is to estimate the intonation of a vocal ensemble when the tones are known. This falls into the category of informed content analysis since the algorithm uses a pre-defined set of reference tones as initial input. A different case would be to try to estimate which tones were present from the set of all possible existing tones.

We assume that the singers hit the tones within an error marginal of ± 1 semitone. We also assume that smallest difference between two reference tones is one semitone. These assumptions will make practical algorithm development more straight-forward, and we still cover almost every practical case.

For the purpose of starting the development of the algorithm, we will next summarise the facts, constraints, criteria and aims. Here is a summary of the facts and constraints known regarding the usage of the intonation analyser module. The sampling frequency of the system is known. The music genre is known to be Western music. The instrument of each part is is known to be the singing voice. The number of voices is known, and it is given as the length of the reference chord input vector. The reference tone is set in advance in the application before the estimation task. In most cases it is A4 = 440 Hz. The reference chord is known and given as an input. The minimum distance between two tones in the reference chord is one semitone. All voices are recorded simultaneously and given as the input sound signal. We assume that each voice is able to hit the right pitch within the error marginal of one semitone above and below the reference tone.

This is a summary of the criteria and aims for a working intonation analyser

module, in other words for the algorithm we are about to develop. The algorithm needs to work in acoustical conditions that correspond to a typical rehearsal situation of a vocal ensemble. It also needs to work for singing signals, and it needs to be able to estimate multiple fundamental frequencies from a polyphonic signal. Frequency resolution is aimed to be at least 7 cents for being able to be as accurate as the human ear [11].

6 Experiments and results

This chapter covers the experiments and results of the development and evaluation process for the vocal ensemble intonation analysis algorithm. The aim of these experiments is to fully exploit the very basic signal processing methods, and discover the possible weaknesses in this approach. Based on the results we can evaluate the need and possibilities to apply more advanced methods for this application.

The most challenging part of the ensemble intonation analysis task, as we have defined it in the analyser module, is the multiple fundamental frequency estimation from a polyphonic signal. Once this part works, the calculation of the deviation from a reference tone and the visualisation of the output should be straight-forward.

In chapter 4 we have covered a wide range of approaches for multiple fundamental frequency estimation. Most of the current approaches are purely or at least partly based on frequency domain analysis. We will choose this approach as well, and start with a very basic experiment of finding magnitude peaks from the frequency response of the sound signal around the fundamental frequencies of the reference tones. Our hypothesis is that it will be fairly easy to get some estimation, but the precision will not be accurate enough, especially at low frequencies.

The second phase of the experimentation is to exploit the harmonic partials. We will perform interpolation around the fundamental frequency and a pre-determined set of harmonic partials. Then we estimate the fundamental frequency by finding the maximum value of the sum of these interpolations.

Before proceeding we derive more specific questions from our main research question. First of all we will start with more generic questions, and answer them right away without performing any formal testing:

- Do we need to take background room noise into consideration?
- Do we need to take room reverberation into consideration?
- Is the quality of standard built-in laptop or mobile device microphone enough for our use case?
- How are our results affected by special characteristics of the singing voice, like vibrato, tremolo or glissando?

The experiments will be carried out in several realistic vocal ensemble rehearsing environments, and using a built-in microphone of an Apple MacBook Pro (Retina, 13-inch, Mid 2014). Background noise should not be an issue in principle, since typically vocal ensembles rehearse in quiet environments with low background noise. Long room reverberation could cause issues in the intonation analysis when analysing fast note sequences. For long notes on the other hand, reverberation should not affect in principle. According to informal preliminary testing, the quality of the built-in microphone should be sufficient. It also reflects the realistic usage of the application, since it is intended to work without additional hardware equipment. We will not research these questions further unless additional issues arise during the formal experiments.

Special characteristics of the singing voice are something to take into consideration especially when determining the length of the analysis frame. Features like vibrato, tremolo and glissando are not as common in a cappella vocal ensemble singing as in solo singing. Nevertheless, singing voice can be readjusted quickly, and our hypothesis is that the analysis frame length needs to be significantly short in order to catch a stable tone sung by a vocal ensemble singer. With too long analysis frame lengths the analysis will be performed based on an unstable pitch.

6.1 Test data and signal model

We will use both real and synthetic sound samples to test and evaluate the performance of the algorithm. The purpose of using real test data is to evaluate the performance of the algorithm in real-life conditions. The drawback of using real data is that we do not have absolute certainty about the which is the exact pitch of each singer on each given time frame of the sound sample. The purpose of using synthetic test data is to be able to compare the estimation result of the algorithm directly with the parameters of the synthetic data. If we for example generate a synthetic signal with the fundamental frequency 223 Hz, set the reference tone to 220 Hz, and we get a deviation of 3 Hz as a result from the intonation analysis, we can assume that the algorithm works correctly. If the result were for example 2 Hz, we know that for some reason the algorithm has a measurement error of 1 Hz for this particular point in the test data. We can not make these evaluations when using real test data, because we do not know with absolute certainty if the fundamental frequency of the test signal was 223 Hz in the first place.

The real data consists of four types of recordings:

1. Combinations of pre-recorded individual voices both approximately in tune and purposely out of tune
2. Male a cappella vocal ensemble live recordings from rehearsing sessions
3. A male a cappella choir live recording from a concert performance
4. A soprano hitting a very high note.

The synthetic test data is generated in MATLAB using a signal model developed for this particular purpose. The model is based on the mathematical representation of complex tones presented in equation 2. Each individual singer of the vocal ensemble is modelled as one complex tone with purely harmonic overtone partials. The whole ensemble is the sum of all the individual singers. The model can be expressed mathematically as a sum of sums of sinusoids.

$$x(t) = \sum_{m=1}^{M} \sum_{n=1}^{N} A_n \sin\left(2\pi n f_m t + \phi_n\right) \tag{13}$$

Here N is the number of overtones modelled for each voice of the vocal ensemble, and M is the number of voices. f_m is the fundamental frequency of each individual

voice. For example, if we want to model the ten lowest overtones for each voice, and the vocal ensemble consists of four singers, $N = 10$ and $M = 4$.

For the purpose of evaluating the effect of glissando on the estimation results, the signal model also supports this feature. The fundamental frequency of each voice can be set to different values at the beginning and end of the generated synthetic sound sample, and the sample will contain a glissando from the start fundamental frequency to the end fundamental frequency of each voice.

6.2 Trivial fundamental frequency estimation

The first iteration of the intonation analysis algorithm performed a standard Fast Fourier Transform to a selected analysis frame from the test signal. It then searched for local peaks in the magnitude response around each reference tone one the range of ± 1 semitone, in other words 100 cents. The local peak around each reference tone was the estimated fundamental frequency. As a result, the algorithm calculated the deviation of the estimated fundamental frequency from the fundamental frequency of the reference tone.

The algorithm was implemented as a MATLAB function. As an input it took an array of samples, the sampling frequency and an array containing the fundamental frequencies of the reference tones in MIDI number format. As an output it provided the measured deviations of intonation in cents. The output variable was returned as an array of the same length as the reference tone array given in the input.

In this first phase of the experimentation we wanted to get answers to the following questions:

- How short must the analysis frame be to achieve enough time resolution?
- How long must the analysis frame be to achieve enough frequency resolution?
- Is the frequency resolution of a standard Fast Fourier Transform enough?
- What tricks can we apply in order to increase frequency resolution without losing time resolution?

The first performed formal tests were done using a one second analysis frame. In this case, the frequency resolution was good enough to reach as much as a 5 cent accuracy in the deviation analysis. Nevertheless, a one second analysis window turned out to be a far too poor time resolution, since the singers may vary their pitch significantly during this time, even when the tone would theoretically be the same during the whole analysis.

With our current understanding, it turned out to be difficult to determine what enough time resolution for this specific use case. The question of what is a suitable time frame to get a stable estimate for the fundamental frequency of a tone performed by singing, which also corresponds to the ability of humans to perceive pitch, remains partly open. Based on own subjective testing, it is possible to perceive pitch shifts from a very short time frame. In this sense, it would be beneficial to be able to use

frame lengths as short as 5–10 ms, which correspond to around 2048–4096 samples with a sampling frequency of 44.1 kHz.

We started to decrease the size of the analysis frame, and observed how it affects the frequency resolution. The resolution for low frequencies, corresponding a typical vocal range of a bass singer, started to drop dramatically with analysis frame sizes of $2^{14} = 16384$ and $2^{13} = 8192$ samples.

In addition, we performed tests both using a Hamming window and a square window, in other words no windowing. Based on these tests, windowing did not significantly affect on the measurement errors due to poor frequency resolution on the low frequencies.

At this point we needed to question whether the standard frequency analysis using Fast Fourier Transform would be enough for this purpose. Then we decided to perform tests using the MATLAB *freqz*[12] function. The function can take a vector of frequencies as an input argument. According to the documentation [35], the function evaluates the polynomials at each frequency point using Horner's method of nested polynomial evaluation, dividing the numerator response by the denominator response. Our analysis function uses a ruler of frequencies to set the peak measurement points around the fundamental frequency of the reference. In our tests with the *freqz* function, we used these frequencies as an input argument. This method significantly decreased the estimation errors we were experiencing with shorter analysis frames at low frequencies. We started to get good results using a frame size of $2^{13} = 8192$ samples, and promising results with a frame size of $2^{12} = 4096$ samples.

There were two consistently problematic intervals, the unison and the octave. The problem with the unison is that both voices singing the interval have the exact same fundamental frequency. When the two voices are at a slightly different pitch, it becomes hard to distinguish which of the estimated fundamental frequencies is representing which singer, in other words who is singing flat and who is singing sharp. The problem with the octave is similar. In this case the fundamental frequency of the higher tone is always shared with the second partial of the lower tone. While the lower tone can be estimated without interference from the fundamental frequency, the higher tone is always interfered by the lower tone. When these two are sung with a slightly different intonation, with our current understanding our estimation of the fundamental frequency of the higher tone is almost randomly either correctly based on the right magnitude peak or mistakenly based on the second partial of the lower tone or a combination of these two.

6.3 Fundamental frequency estimation using solitary partials

A great amount of information of the vocal ensemble intonation is in the harmonic spectrum of a polyphonic signal. As a hypothesis, a possible solution for the encountered challenges could arise from systematic experimentation with the overtone partials. This is the reasoning for the second phase of the experimentation.

In this second phase of the experimentation we wanted to get answers to the following questions:

[12] `https://se.mathworks.com/help/signal/ref/freqz.html`

– Can we further shorten the analysis frame?

– Does this make the algorithm more robust?

In this phase we developed the concept of "solitary partials". These are partial overtones that are not affected by the other tones, and could be used for a reliable fundamental frequency estimation. In this experiment, before the actual fundamental frequency estimation, an auxiliary function of the algorithm performs a selection of partials frequencies ranging from the fundamental frequency up to an upper limit, which was set to 8 kHz. All the partials that are not affected by the other reference tones are selected for the fundamental frequency estimation. The remaining partials are not used, since the other voices would most probably affect the result.

Once the partials are selected, the actual fundamental frequency estimation is performed. The estimation is done by interpolating over each partial and summing the results.

The criteria for a solitary partial is defined by setting an upper and lower limit for the clear zone, or what we could also call "patch". In our first iteration we set this to be ±100 cents, same as the analysis area around the reference fundamental frequency from where the peaks are searched for the fundamental frequency estimation. It turned out that partials of other tones in the polyphonic signal affect strongly at the fringe areas of the estimation ranges, resulting in lots of estimation errors. This got fixed by setting the clear zone to ±100 cents further from the analysis area. This means that if the clear zone is set to ±150 cents, the analysis area must be set to ±50 cents, which is one semitone in total, but only half of a semitone flat or sharp in relation to the reference tone. If we set the clear zone is set to ±200 cents, the analysis area must can set to ±100 cents.

Using these parameters, ±200 cents for the clear zone and ±100 cents for the analysis area, we performed systematic testing using analysis frame sizes of $2^{13} = 8192$, $2^{12} = 4096$ and $2^{11} = 2048$ samples. Based on the results, $2^{13} = 8192$ samples is still the most reliable option in terms of reducing estimation errors, although the overall robustness of the algorithm increased.

The main reason why applying the concept of solitary partials did not dramatically increase the results, is most certainly the fact that Western music, especially tonal vocal ensemble music, tends to be extremely consonant. In other words, there is always a great amount of overlapping present in the frequency spectrum. We conducted a test, where the number of available solitary partials were defined for all the intervals in the twelve tone equal temperament from unison to octave. We tested three different clean areas, ±100, ±150, and ±200 cents. The upper limit of the frequency range for the analysis was set to 8 kHz. The results are shown in table 2.

The results show that both the unison and the minor second have zero solitary partials, not even the fundamental frequency is solitary. The minor second did not cause issues in other tests, which is probably because it is not very common in Western tonal music. The unison on the other hand is present quite often. In the case of the octave, while the lower tone has plenty of solitary partials even using the widest clear zone, the upper tone has zero solitary partials in all the cases. The octave is a very common interval in Western tonal music, and errors in estimating the

Table 2: Number of available solitary partials in each interval, presented separately for the lower and upper interval tone. The results are presented for three different clean zone sizes, ±100, ±150, and ±200 cents.

	±100 c		±150 c		±200 c	
	Lower	Upper	Lower	Upper	Lower	Upper
Unison	0	0	0	0	0	0
Minor second	0	0	0	0	0	0
Major second	6	5	5	4	4	3
Minor third	4	3	4	3	3	2
Major third	4	2	3	2	3	2
Perfect fourth	6	3	4	2	3	2
Tritone	9	5	3	1	3	1
Perfect fifth	6	3	4	2	3	1
Minor sixth	4	1	4	1	3	1
Major sixth	7	2	5	2	3	1
Minor seventh	8	3	6	2	2	0
Major seventh	12	0	3	0	3	0
Octave	8	0	6	0	4	0

upper tone of an octave interval relation has turned out to be the greatest issue in the whole task of analysing the intonation of a vocal ensemble. There is nevertheless benefit in using the solitary partial approach for estimating the lower tone in the octave interval, since this is often the lowest tone in the chord, and we need to increase accuracy especially in the lower parts of the frequency range.

All the other interval relations have at least some solitary partials available for taking into account in the fundamental frequency estimation. Especially when using a clean zone of ±100 or ±150 it is worth performing the estimation using the solitary partial approach.

We also took a popular four-part Finnish a cappella vocal ensemble piece, Finlandia-hymni by Jean Sibelius, and conducted the same tests as we did for the intervals. It turned out that a vast majority of the tones in the whole piece have either zero or one solitary partials, and having more than three solitary partials is an exceptionally rare case. The number of tones with zero or one solitary partials decreases as the clear zone becomes wider.

In addition to the already known challenges of estimation errors occurring on low frequencies, and the unison and octave problems, one new issue was encountered in this phase. In fact, this was already present on the first phase, but due to the increased amount of testing as a whole, it became more visible. In some occasions, typically when two tones are close to but not exactly a pure octave, there appears some kind of jumping phenomena on the frequency response. One hypothesis is that this is beating. Nevertheless, it results in estimation errors in respect to the expected result when testing with a synthetically generated chord constructed by

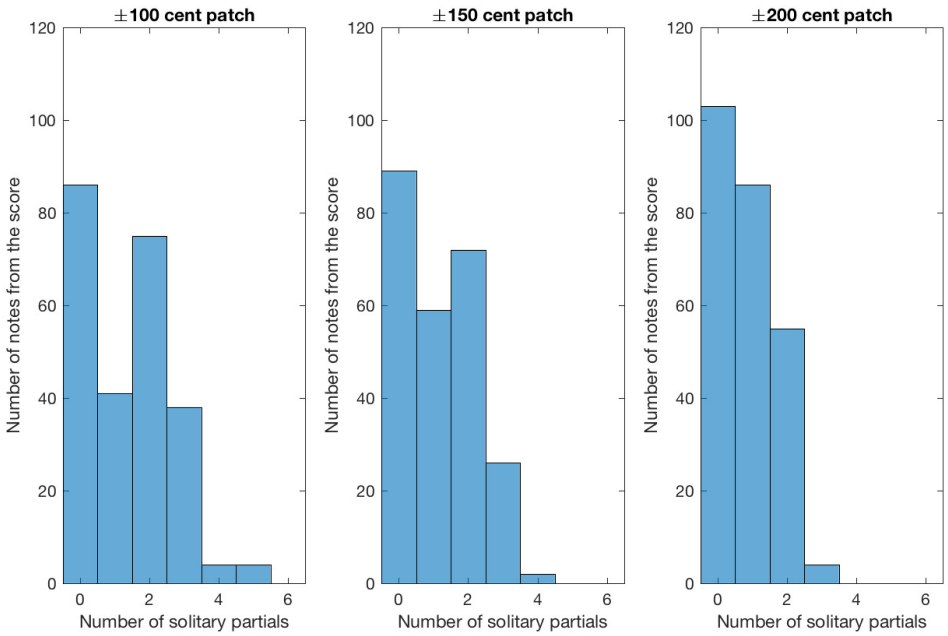

Figure 13: Histogram of the number of tones according to the number of available solitary partials in each tone in a popular four-part a cappella vocal ensemble piece, Finlandia by Jean Sibelius. The results are presented for three different clean zone sizes, ±100, ±150, and ±200 cents.

giving constant pitch values for all the voices.

6.4 Real-time experiment

Although there are some unresolved issues remaining based on the results for the first and second phases of experimentation, we wanted to build a real-time prototype that can be tested with real users. It was built using the MATLAB Audio System Toolbox[13] combined with a graphical UI constructed using the MATLAB App Designer[14]. The user interface of the MATLAB prototype is shown in figure 14.

In this third and last phase of the experimentation we wanted to get the answer to the following question:

– How does this algorithm perform in a real-time application?

During the preliminary technical testing of the real-time application it turned out, that the jumping effect pointed out in the second phase could be more common than what was anticipated based on the previous experiments. The shorter the analysis window, the more jumping is shown on the real-time graphical analysis. Whether this phenomenon is beating or something else, it definitely affects our

[13] http://se.mathworks.com/products/audio-system/
[14] http://se.mathworks.com/products/matlab/app-designer/

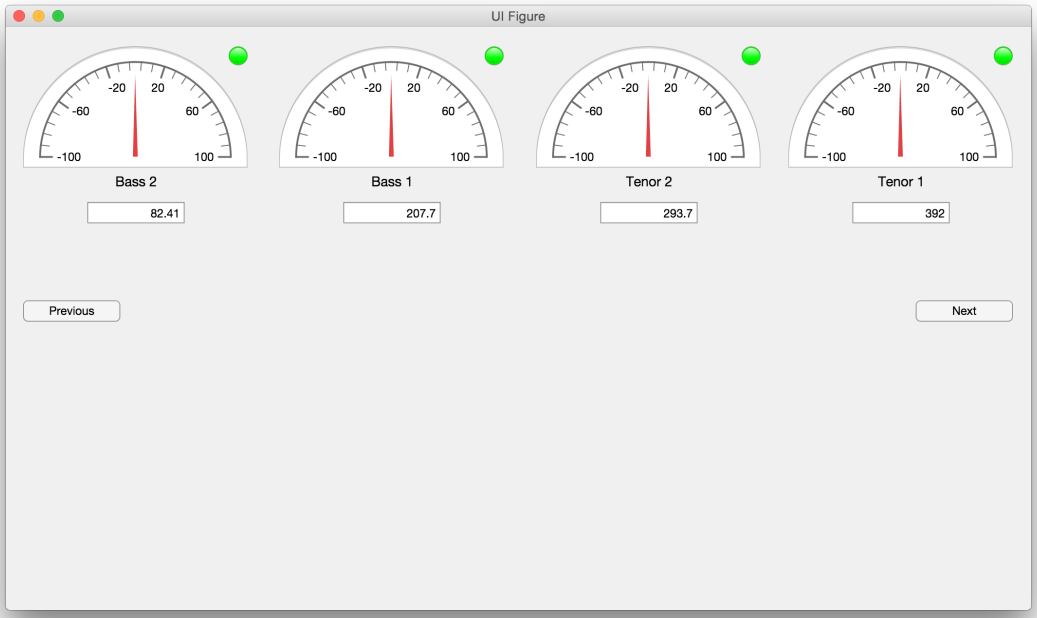

Figure 14: The UI of the real-time prototype tested during a rehearsal session of an a cappella vocal ensemble consisting of four male singers. A sequence from one of the most challenging pieces in the repertoire of the ensemble was hard-coded into the program as reference chords, and the Previous and Next buttons made it possible to switch between chords within the sequence. Each singer has his own gauge showing the deviation from the reference tone in cents, as well as a lamp, which switches between green, yellow and red depending on how far the estimated pitch is from the reference tone. The number shown in the text field under each gauge is the fundamental frequency of the reference tone in hertz.

fundamental frequency estimation in a negative way. By keeping the analysis frame size at $2^{13} = 8192$ samples, we were able to keep the estimation somehow stable.

The real-time application was informally tested with a real vocal ensemble consisting or four male singers in a real-life rehearing scenario. Two of four of the test subjects reported that their needle was unstable while they were fully confident that their pitch was stable. This indicates that the jumping or beating problem was present also during this test. Overall, the test subjects felt that a rough estimate, e.g. visualised as discrete zones instead of a constantly moving needle, would actually be more useful and user-friendly. It would be enough to indicate when someone is out of tune, or if the key of the whole ensemble starts to drift. This observation brings up a new perspective for both the application concept and the actual algorithm requirements. It is possible that the whole estimation task becomes less demanding, if we drop the estimation resolution from 5 or 7 cents to e.g. 10 or 20 cents.

7 Conclusions

Automatic vocal ensemble intonation analysis is not a trivial task. This thesis introduces two main contributions in this topic. First, automatic vocal ensemble intonation analysis is investigated as a practical application. As a result, we present an application concept, which could help vocal ensembles in practicing their intonation. Second, sound signals produced by vocal ensembles are investigated with the purpose of developing an algorithm for vocal ensemble intonation analysis. As a result, we present an algorithm, which is capable of analysing intonation in real time. We take advantage of what we call solitary partials to increase accuracy and robustness of the algorithm.

Our application and algorithm could be used for several purposes. They could be developed into a product for vocal ensembles to analyse their intonation in real time while they practice, which is the primarily intended use case. They could also be developed into a product for analysing vocal ensemble recordings. Both real-time and non-real-time applications could be used for training the intonation of vocal ensemble performers. Both applications could also be used for musicological studies on intonation practices for research purposes.

The algorithm developed in this thesis is based on spectral analysis of the sound signal. The development and evaluation of the algorithm conducted in this work focuses on a few main questions. The first one is finding the optimal window length for the spectral analysis. The requirements for the window length are that it should be short enough to contain only the information at one given point in time, and at the same time long enough to achieve a frequency resolution high enough. The highest requirements for the frequency resolution are dictated by the lowest tones that need to be analysed, since frequency differences between two consecutive tones are especially small at low frequencies. Based on our experiments, a window length of $2^{13} = 8192$ samples gives enough frequency resolution and a reasonable time resolution, even when analysing only the fundamental frequency of each tone, without taking advantage of harmonic partials.

The second question is about finding the solitary harmonic partials for each tone in the polyphonic sound signal that are not affected by harmonics of fundamental frequencies of other tones. These can then be used for increasing the accuracy and robustness of the estimation. Based on our tests, there are many cases where solitary partials can be found and used in the estimation. Nevertheless, there are several harmonic relations in which some of the tones do not possess any solitary partials, and even the fundamental frequency is shared with partials of other tones. In particular the unison and the octave are intervals that cause this situation.

Western tonal vocal ensemble music is extremely consonant, in other words, there is a great amount of spectral overlapping on the music signal of a vocal ensemble. This makes separation of sources particularly hard. Our tests show that in a typical four-part vocal ensemble piece the majority of the tones have either zero or one solitary partials. Nevertheless, our solitary partial approach can be applied for the remaining notes with two or more solitary partials, and this gives us a more accurate and robust estimate for these particular notes. Especially, when the lowest note is

in an octave interval relation with other notes, it has a great amount of solitary partials. By using these partials we are able to compensate the inaccuracy that is characteristic to low notes.

The usability of the proposed real-time application concept was tested with real users only on an informal preliminary test, right after the algorithm had started working on a tolerable level. An extended and more systematic usability study would provide information on the true user value of the application.

Based on our results, an automatic intonation analyser is able to provide accurate information about the intonation of a vocal ensemble in real-time. The next step would be to test how the automatic intonation analyser performs against a trained human listener, such as a vocal coach or a choir director.

Our work focuses purely on analysing deviations in intonation from a single reference chord, without taking the musical context into consideration. The automatic intonation analyser could be combined with musicological models about intonation practices and theory on a wider musical context. This would both provide information for researchers on intonation practices and guide music performers in their training.

The experiments of this research are based on a very basic mathematical approach. In order to tackle the encountered challenges, such as the octave problem, more sophisticated models and approaches could be investigated. One possible proposed path is bayesian model fitting.

This work has been limited to a cappella vocal ensembles and Western tonal music. A possible topic for future work would be applying this task in choir music. The main difference between vocal ensembles and choirs is that in vocal ensembles one part is mainly sung by one individual singer, while in choirs multiple singers share the same part. The hypothesis is that the dispersion in intonation within each part needs to be taken into account in some way.

The adaptability or the algorithm for various musical genres and other instruments could be further studied. Other possible genres which have been out of the scope of this thesis include contemporary atonal vocal ensemble music as well as pop, jazz, folk and ethnic music. Many other instrumentalists also need to take intonation into consideration. The closest relative to a vocal ensemble could be a string ensemble since strings are non-fretted instruments.

References

[1] M. Schedl, E. Gómez, and J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations and Trends® in Information Retrieval*, vol. 8, no. 2–3, pp. 127–261, 2014.

[2] A. Klapuri and M. Davy, eds., *Signal Processing Methods for Music Transcription*. New York: Springer, 2006.

[3] M. Muller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[4] J. S. Downie, "The scientific evaluation of music information retrieval systems: Foundations and future," *Computer Music Journal*, vol. 28, no. 2, pp. 12–23, 2004.

[5] A. Bachem, "Absolute pitch," *The Journal of the Acoustical Society of America*, vol. 27, no. 6, pp. 1180–1185, 1955.

[6] J. G. Roederer, *The Physics and Psychophysics of Music: An Introduction*. New York: Springer, 4th ed., 2008.

[7] T. D. Rossing, F. R. Moore, and P. A. Wheeler, *The Science of Sound*. Reading (MA): Addison-Wesley, 3rd ed., 2002.

[8] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Boston: Academic Press, 5th ed., cop. 2003.

[9] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990.

[10] M. Ryynänen, "Singing transcription," in *Signal Processing Methods for Music Transcription* (A. Klapuri and M. Davy, eds.), Springer, 2006.

[11] J. Sundberg, "The perception of singing," in *The Psychology of Music* (D. Deutch, ed.), Academic Press, 1999.

[12] N. Fletcher and T. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[13] J. Sundberg, *The Science of the Singing Voice*. Northern Illinois University Press, 1987.

[14] R. Maher and J. Beauchamp, "An investigation of vocal vibrato for synthesis," *Applied Acoustics*, vol. 30, pp. 219–245, 1990.

[15] J. I. Shonle and K. E. Horan, "The pitch of vibrato tones," *Journal of the Acoustical Society of America*, vol. 67, no. 1, pp. 246–252, 1980.

[16] C. E. Seashore, *Psychology of Music*. Dover Publications, 1967.

[17] D. Tidhar, M. Mauch, and S. Dixon, "High precision frequency estimation for harpsichord tuning classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 61–64, 2010.

[18] J. Devaney and D. P. W. Ellis, "An empirical approach to studying intonation tendencies in polyphonic vocal performances," *Journal of Interdisciplinary Music Studies*, vol. 2, no. 1–2, pp. 141–156, 2008.

[19] J. Backus, *Acoustical Foundations of Music*. New York: W.W. Norton & Company Inc, 1969.

[20] J. M. Barbour, *Tuning and Temperament: A Historical Survey*. East Lansing: Michigan State College Press, 1953.

[21] R. Rasch, "Tuning and temperament," in *The Cambridge History of Western Music* (T. Christensen, ed.), Cambridge University Press, 2002.

[22] H. Helmholtz, *On the Sensation of Tone as a Physicological Basis for the Theory of Music*. New York: Dover Publications, 1863. Translated by A.J. Ellis. 1954.

[23] R. Plomp and W. J. M. Levelt, "Tonal consonance and critical bandwidth," *Journal of the Acoustical Society of America*, vol. 38, no. 4, pp. 548–560, 1965.

[24] E. Terhardt, "The concept of musical consonance: A link between music and psychoacoustics," *Music Perception*, vol. 1, no. 3, pp. 276–295, 1984.

[25] F. Lerdahl, *Tonal Pitch Space*. Oxford: Oxford University Press, 2001.

[26] F. Lerdahl and C. Krumhansl, "Modeling tonal tension," *Music Perception*, vol. 24, no. 4, pp. 329–366, 2007.

[27] S. Larson, "Musical forces and melodic expectations: Comparing computer models with experimental results," *Music Perception*, vol. 21, no. 4, pp. 457–498, 2002.

[28] L. Meyer, *Emotion and Meaning in Music*. Chicago: University of Chicago Press, 1956.

[29] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2 ed., 1999.

[30] D. Rockmore and D. Healy, *Modern Signal Processing*. Mathematical Sciences Research Institute Publications, Cambridge University Press, 2004.

[31] Documentation for the MATLAB *interp1* function: https://se.mathworks.com/help/matlab/ref/interp1.html, last visited 9.10.2016.

[32] T. Abe and S. I. T. Kobayashi, and, "Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency," *Proceedings of the International Conference on Spoken Language*, vol. 2, pp. 1277–1280, 1996.

[33] J. C. Brown and K. V. Vaughn, "Pitch center of stringed instrument vibrato tones," *Journal of the Acoustical Society of America*, vol. 100, no. 3, pp. 1728–1735, 1996.

[34] P. De La Cuadra, A. Master, and C. Sapp, "Efficient pitch detection techniques for interactive music," in *Proceedings of the 2001 international computer music conference*, pp. 403–406, 2001.

[35] Documentation for the MATLAB *freqz* function: https://se.mathworks.com/help/signal/ref/freqz.html, last visited 4.10.2016.

A Appendix: MATLAB functions

This section presents the source code of the MATLAB functions developed for the experiments. Only the functions related to the automatic vocal ensemble intonation algorithm and the signal model for crating synthetic test sounds are presented. All the auxiliary code for e.g. running tests and evaluating results as well as the MIDI number and cent conversion functions have been excluded from this appendix.

A.1 intonationAnalysis.m

```
function [deviations, nrOfUsedPartials, cleanPartials] = ...
    intonationAnalysis(frame, fs, targets)
% INTONATIONANALYSIS  Analyse intonation from a polyphonic signal.
%   [DEVIATIONS, NROFUSEDPARTIALS, CLEANPARTIALS] =
%   INTONATIONANALYSIS(FRAME, FS, TARGETS) returns DEVIATIONS, the
%   deviations in cent units of individual voices in the signal FRAME
%   in reference to TARGETS, a vector or a scalar containing the
%   reference tones in midi numbers. In addition, returns the number
%   of used clean (solitary) partials in the analysis, and if
%   there were clean (solitary) partials found. The sampling frequency
%   of the sound signal FS must be given.
%
%   See also INTONATIONANALYSIS, FINDCLEANPARTIALS.


%% Initialisations for deviation analysis

% Lower limit in cents
lLim = -100;
% Upper limit in cents
uLim = 100;
% Resolution in cents
res = 1;
% Pitch ruler in cents
rulerC = lLim:res:uLim;

% Highest frequency
upperLimit = 8e3;


%% Deviation analysis

deviations = zeros(1,length(targets));
nrOfUsedPartials = zeros(1,length(targets));
cleanPartials = zeros(1,length(targets));

% MIDI to Hz
targets = midi2hz(targets);

% disp('Clean partials, freqency, deviation')

% For each target frequency (each voice)
```

```
for n = 1:length(targets)

    % Find clean partials
    cleanPartials(n) = 1;
    partials = findCleanPartials(n, targets, upperLimit);
    if isempty(partials)
        partials = 1;
        cleanPartials(n) = 0;
    end
    nrOfUsedPartials(n) = length(partials);

    % Convert ruler to Hz
    rulerHz = zeros(1,length(rulerC));
    for m = 1:length(rulerHz)
        rulerHz(m) = cent2hz(rulerC(m),targets(n));
    end

    % Perform analysis using clean partials
    intp = zeros(length(rulerHz),length(partials));
    for m = partials
        [pxx, f] = freqz(frame,1,m.*rulerHz,fs);
        pxxabs = abs(pxx);
        pxx2 = pxxabs.^2;
        intp(:,m) = interp1(f,pxx2,'spline');
    end
    w = sum(intp,2);
    [peak_value, peak_index] = max(w);
    peak = rulerHz(peak_index);

    deviations(n) = hz2cent(peak,targets(n));

end

end
```

A.2 findCleanPartials.m

```
function cleanPartialIndexes = findCleanPartials(thisFreqI, ...
    allFreq, upperLimit)
% FINDCLEANPARTIALS  Find the clean (solitary) partials of a chord.
%   CLEANPARTIALINDEXES = FINDCLEANPARTIALS(THISFREQI, ALLFREQ,
%   UPPERLIMIT) returns the indexes of the harmonic partials which
%   do not overlap with other harmonics of fundamental frequencies.
%   THISFREQI is the index of ALLFREQ vector that needs to be
%   examined. UPPERLIMIT is the frequency limit for the highest
%   partial that will be examined.
%
%   See also INTONATIONANALYSIS, SIGNALMODEL.


%% Initialisations

% Lower limit of the fundamental frequency in cents
lLim = -200;
```

```
18 % Upper limit of the fundamental frequency  in cents
19 uLim = 200;
20
21
22 %% Find clean Partials for the fundamental frequency at index thisFreqI
23
24 % Convert limits to Hz fo the given fundamental frequency
25 lLimHz = cent2hz(lLim,allFreq(thisFreqI));
26 uLimHz = cent2hz(uLim,allFreq(thisFreqI));
27
28 % Find indexes of the other fundamental frequencies
29 otherI = 1:length(allFreq);
30 otherI(thisFreqI) = [];
31
32 otherTones = [];
33 for ii = otherI
34     % Define the highest harmonic partials for the other fundamental
35     % frequcenies
36     H = 1;
37     while H * allFreq(ii) < upperLimit
38         H = H + 1;
39     end
40     % Find harmonic partials of all the other tones
41     for h = 1:H
42         otherTones = horzcat(otherTones, h*allFreq(ii));
43     end
44 end
45
46 % Define the highest harmonic partial for the given fundamental
47 % frequceny
48 H = 1;
49 while H * allFreq(thisFreqI) < upperLimit
50     H = H + 1;
51 end
52
53 % Create a logical array of the size of number of partials
54 cleanPartials = true(1,H);
55
56 % Find clean harmonic partials - change to false, if interfered by
57 % other tones
58 for h = 1:H
59     for ii = 1:length(otherTones)
60         if otherTones(ii) >= h*lLimHz && ...
61                 otherTones(ii) <= h*uLimHz
62             cleanPartials(h) = false;
63         end
64     end
65 end
66
67 % Output indexes of clean partials
68 cleanPartialIndexes = [];
69 for n = 1:length(cleanPartials)
70     if cleanPartials(n) == true
71         cleanPartialIndexes = horzcat(cleanPartialIndexes, n);
```

```
72      end
73  end
74
75  end
```

A.3 signalModel.m

```
1   function [data, fs] = signalModel(fm, am, fam, devStart, devEnd)
2   % SIGNALMODEL  Create a synthesised polyphonic sound signal.
3   %   [DATA, FS] = SIGNALMODEL(FM, AM, FAM, DEVSTART, DEVEND) returns
4   %   DATA, an array of samples representing a polyphonic sound signal,
5   %   and its sampling frequency FS. All input values must be given.
6   %   FM can be a vector or a scalar containing the pitch values as MIDI
7   %   note numbers. AM, FAM, DEVSTART and DEVEND must scalars or vectors
8   %   of the same length as FM. AM is the loudness value between 0 and 1
9   %   of each tone. FAM produces tremolo with values approximately from
10  %   5 to 20, and dynamic changes with values approximately from 0.1 to
11  %   5. The pitch of each tone can be deviated by giving different
12  %   DEVSTART and DEVEND values in cent units.
13  %
14  %   See also INTONATIONANALYSIS, FINDCLEANPARTIALS.
15
16
17  %% Basic parameters
18
19  fs = 48e3;                  % sampling frequency
20
21  % MIDI to Hz
22  fm = midi2hz(fm);
23
24  % time and signal data vectors
25  t = linspace(0,1,fs);
26  sig1 = zeros(1,fs);
27  sig = zeros(1,fs);
28
29
30  %% harmonic partials
31
32  fn = 146;                       % nr of harmonics
33  an = logspace(1,0.1,fn) ./ 10;  % loudness of partials (attenuation)
34
35  %% Deviation
36
37  % glissando time
38  devTime = 0.75*fs;
39  % deviation matrix in cents
40  dev = zeros(length(devStart),fs);
41  for m = 1:length(devStart)
42      dev(m,:) = horzcat(linspace(devStart(m),devEnd(m),devTime), ...
43          devEnd(m)*ones(1,fs-devTime));
44  end
45  % initiate deviation matrix in Hz
46  devHz = zeros(length(devStart),fs);
47
```

```
48
49 %% Signal model and construction
50
51 for m = 1:length(fm)
52     % randomise phase
53     phi = 2*pi*rand;
54     for n = 1:fn
55         if n*fm(m) < 8e3
56             % deviation array in Hz
57             for o = 1:size(devHz,2)
58                 devHz(m,o) = cent2hz(dev(m,o),n*fm(m));
59             end
60             % phase correction array
61             phaseCorrection = ...
62                 cumsum(t.*horzcat(0,2*pi*(devHz(m,1:fs-1) ...
63                 -devHz(m,2:fs))));
64             % single singer signal construction
65             sig1 = sig1 + am(m) * an(n) * sin(2*pi*devHz(m,:).*t ...
66                 + phi + phaseCorrection);
67         end
68     end
69     % add tremolo / crescendo / diminuendo
70     sig1 = sig1 .* (1 + 0.5*sin(2*pi*fam(m)*t));
71     % add to previous single singer signals
72     sig = sig + sig1;
73 end
74
75
76 %% Post-processing
77
78 % normalisation
79 sig = sig./max(abs(sig));
80 % smoothen the beginning and end in the time domain
81 w = hann(2*256)';
82 w = [w(1:256) ones(1,length(sig)-length(w)) w(257:end)];
83 sig = w.*sig;
84
85 %% Output
86
87 data = sig;
88
89 end
```

B Appendix: Application concept mockups

This section presents the evolution of the application concept user interface (UI). The progress is shown in figures B1, B2, B3, and B4. In addition, the design process generated a list of possible features for the application.

The algorithm developed in this thesis takes the reference chord in MIDI note numbers as an input. The reference chord could be generated in three different ways. It could be entered as a single standalone chord using a software or hardware MIDI keyboard. Alternatively, the whole score of a piece could be imported to the application in MIDI or MusicXML format. In this case, the score could be aligned to the audio signal either manually by the users with using a Next button for example, or automatically using a separate score alignment module. The third and most advanced alternative would be to introduce a robust real-time transcription module, which first detects the chord within the accuracy of a semitone as a previous step for the intonation analysis.

The reference intonation follows equal temperament by default. It could be possible to introduce just intonation as an option. In this case, the key of the music piece should be set, either by automatically reading it from the score or inserted by the user. One very advanced option would be to introduce an intelligent intonation analyser, which would be able to suggest the right intonation based on the musical context.

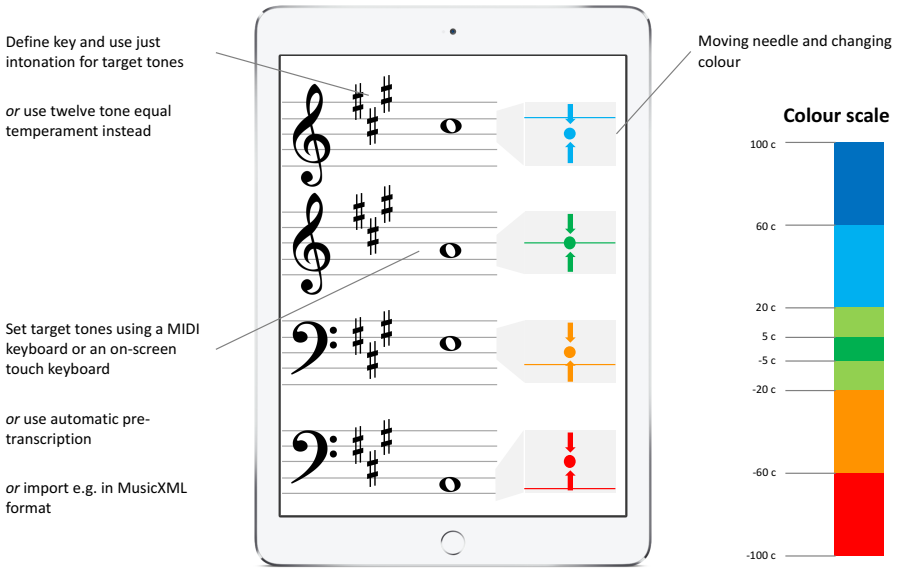

Figure B1: First iteration of the application concept. This illustration also introduces the colour scale for the intonation meter, two shades of green for the right or nearly right intonation, two shades of blue for too sharp intonation, and two shades of red for two flat intonation. The measured intonation is indicated both with the position of a moving needle and with the changing colour.

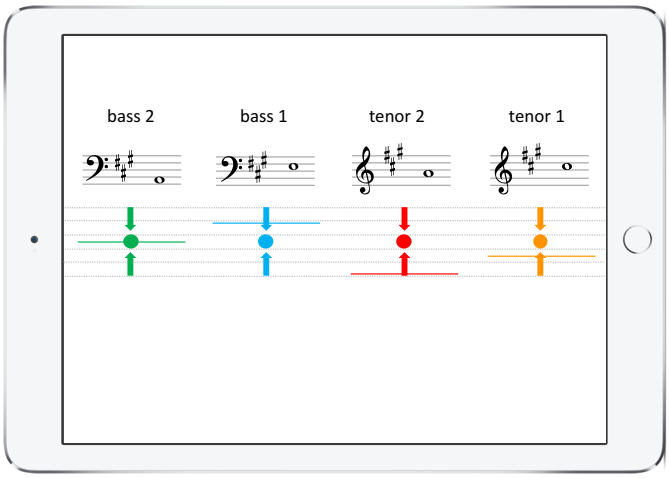

Figure B2: Second iteration of the application concept. Here the staff lines have a smaller emphasis in comparison to the first iteration in order to bring attention to the intonation meters, which are the main part of the UI.

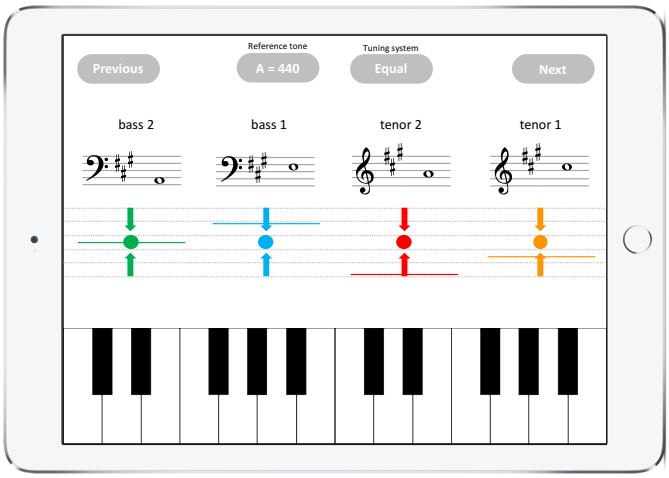

Figure B3: Third iteration of the application concept. Here the basic UI is the same as in the previous iteration, but some extra functionality has been added. The Previous and Next buttons could be used to go back and forward in a sequence of pre-inserted reference chord, for example an entire score. The reference tone could be changed with a button, as well as the tuning system.

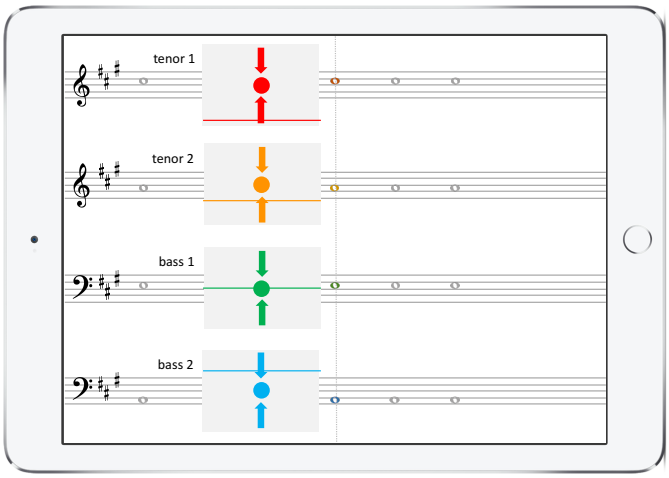

Figure B4: Fourth iteration of the application concept, same as presented in the thesis. We wanted the user experience to be as familiar as possible for vocal ensemble singers, and we decided to align the staff in the same way as it is done in regular sheet music. Here the notes would move horizontally from right to left, when advancing from the previous to the next reference chord, and the other elements on the screen would remain in their places.

Other features of the application could include selecting the reference tone (A4 = 440 / A4 = 442 / A4 = 415 / custom), selecting the clef (SATB / TTBB / SSAA / other), as well as using an external hardware MIDI keyboard. The application could also be turned into a game, like a Singstar[15] for ensembles.

[15] https://www.singstar.com/