POLITECNICO DI MILANO
Scuola di Ingegneria dell'Informazione

Master of Science in
Computer Engineering

Techniques for automatic dissonance suppression in harmonic mixing

Candidate

Vittorio Maffei
Student Id. number 819631

Thesis Supervisor
Prof. Fabio Antonacci

External Supervisor
Dr. Matthew E. P. Davies

Academic Year 2014/2015

POLITECNICO DI MILANO
Scuola di Ingegneria dell’Informazione

Laurea Magistrale in
Ingegneria Informatica

Tecniche per soppressione automatica di dissonanza nel mixing armonico.

Candidato

Vittorio Maffei
Matricola 819631

Relatore
Prof. Fabio Antonacci

Relatore Esterno
Dr. Matthew E. P. Davies

Anno Accademico 2014/2015

Techniques for automatic dissonance suppression in harmonic mixing
Master thesis. Politecnico di Milano

Author’s email: vittorio.maffei@mail.polimi.it

Sommario

Negli ultimi decenni, la figura del DJ è diventata una nuova tipologia di performer musicale. L'abilità di offrire un mix continuo di musica è in costante evoluzione e le tecniche di sequenziamento e mixing sono state perfezionate fino a raggiungere un nuovo livello di interpretazione musicale. Al giorno d'oggi, la possibilità di accedere online ad archivi di musica digitale di grandi dimensioni offre agli utenti la possibilità di scoprire sempre nouve canzoni. Di conseguenza, un nuovo insieme di tecnologie si è rapidamente evoluto, il quale connette la musica digitale con le tecniche dei DJ offrendo software per il mixing automatico di musica. Lo scopo principale di questo progetto è di esplorare il settore del mixing automatico sviluppando nuove tecniche per il mixing armonico. L'idea è quella di creare un modello di compatibilità tra diverse tracce basato sulla combinazione di caratteristiche psicoacustiche della consonanza musicale e informazioni armoniche del segnale audio. In questo progetto presentiamo un sistema che, al fine di produrre un mix tra tracce musicali differenti che risulti più piacevole all'ascolto, applica specifiche operazioni di processing direttamente sul segnale. La novità principale rispetto agli approacci esistenti, consiste nell'introduzione di un metodo di soppressione di dissonanza. Con il fine di esplorare nuove tecniche per il mixing di musica digitale e migliorare la compatibilità armonica del risultato finale, proponiamo tre diversi approacci basati sull'analisi e l'interazione del contenuto in frequenza del segnale musicale. Partendo da due tracce il sistema per prima cosa estrae un insieme di parziali che descrivono il contenuto armonico per mezzo di un modello di analisi sinusoidale. Poi, basandosi su un modello psicoacustico di *asprezza sensoriale*, fornisce il cambio di tono ottimale tra le tracce che massimizzi la consonanza del mixing finale. Per sopprimere la dissonanza del mix finale l'algoritmo identifica e maschera il contenuto più dissonante in tre modi, i) sopprimendo alcuni intervalli di tempo in una delle tracce input, o ii) selezionando e sopprimendo uno specifico insieme di parziali dissonanti per ogni istante di tempo e iii) ampliando questo metodo di soppressione rimuovendo un secondo insieme di parziali armonicamente collegate.

Abstract

In the last decades, DJs have become a new type of music performers. The ability to provide a continuous mix of music is constantly evolving and the skills of sequencing and mixing have been refined to reach new levels of music interpretation and expression. Nowadays, online access to large-scale archives of digital music offers users the possibility to discover many new songs. Consequently, a new set of technologies has been rapidly evolving, which connects digital music with the DJ-mixing techniques by providing software for the automatic mixing of music. The main purpose of this project is to explore the field of automatic music mixing by developing a new harmonic mixing technique. The idea is to design a model of music compatibility between tracks based on the combination of psychoacoustic features related to musical consonance and harmonic information of the music signals. In this project we introduce a system that aims to produce a pleasant final mix between different music tracks by applying specific processing operations on music at the signal level. The main novelty with respect to existing approaches is that we introduce a dissonance suppression method. In order to explore new techniques to mix digital music and improve the compatibility of the final mix, we propose three different approaches based on the analysis and interaction of the frequency content of the music spectra. Starting from two tracks, the system first extracts a set of partials describing the harmonic content by means of a sinusoidal analysis model. Then, according to a psychoacoustic model of sensory roughness, it provides the optimal pitch-shift ratio between the tracks, which maximizes the consonance of the resulting mix. In order to suppress the dissonance of the final mix the algorithm then identifies and masks the most dissonant content in three ways, i) by suppressing time frames of one of the input tracks, or ii) by selecting and suppressing a specific set of dissonant partials within each time frame and iii) by expanding this partial suppression method to remove a set of harmonically related partials. We evaluate our system by several objective examples under highly controlled conditions, and then on real-world music examples via a small scale listening test.

Table of Contents

Table of Figures

Chapter 1
Introduction

Music has the incredible power to influence and change the atmosphere of the surrounding environment. Not only live performed music but also recorded music has such a power. To this end, a disk jockey (DJ) is the person that can maximize the potential of music by playing it effectively through the use of specific techniques and tools. This relatively new type of music performer selects and mixes tracks in order to provide the best music for the atmosphere without letting the music to stop playing. One fundamental technique is to gradually switch from one song to the other, while adjusting the beats of the songs in order to make song boundaries unnoticeable. This technique enables switching between songs smoothly without disturbing the listener, and consequently is the minimum and most fundamental factor for this kind of performance. In addition to this, the DJ must decide and set the next track and the mixing timing before the song that is currently playing reaches its end. Selecting one best mixing track from innumerable possibilities within a limited time is a very difficult task for DJs. Although song selection is an important task, in this project we focus our attention on the technique that provides the continuous mix of different songs. Our reason is based on the fact that the process of song selection is highly dependent on the situation, atmosphere, and meta-information of the song [1]. Moreover, the originality of the DJs performance heavily impacts song selection. As the main motivation which stands behind the research of this thesis, we can state that song selection is what a human being is good at, whereas song mixing is what also a computer system can be good at.

The past few years have been characterized by the advent of digital information that has radically changed the way the world relates with multimedia content. Music has been strongly involved in this innovation and thanks to quick evolution of digital audio formats, has become one of the most investigated fields. Due to the ease of downloading, uploading and sharing the music brought by the Web, the amount of musical content available online has constantly increased. Nowadays, the quantity of songs that are accessible for the users is larger than a person could listen to in his/her entire life, and consequently both the music listening experience and the music interpretation/playing has radically changed. The evolution of music technology has created the need of manage the musical content and find ways to automatically extract information from it. Consequently, it has brought to the development of an interdisciplinary science, called Music Information Retrieval (MIR), which specifically deals with capturing useful information from the musical content. The interdisciplinarity comes from the need of a background in not only signal processing and machine learning techniques, but also in musicology, music theory and psychoacoustics. As a consequence of the rise of these new technologies, also the world of DJs has rapidly changed, and the digital era of music has opened up this music playing technique to a huge range of users. The field of research that specifically relates with DJ reality of music is an emerging area within the MIR community, the so-called creative-MIR [2]. One of the main goals of creative-MIR is to open new possibilities for music creation, interaction and manipulation, matched by the ability to robustly analyse and interpret music signals. Simultaneously with the evolution of this new technology, a growing number of modern DJs are increasingly turning to digital music.

Consequently, a new set of specialized software has been developed in order to manipulate digital music files on laptop computer or portable device and help the DJ in the mixing process. This new type of software technology allows users to automatically re-create in a software environment the process of aligning different tracks with each other to achieve smooth musical transitions. It not only provides the automation of the real world DJ-mixing process but also allow users to explore new techniques for music mixing based on the digital processing of the music signal. We refer to this new type of software technology as Automatic Music DJ-Mixing. Few studies have been searching for new technical knowledge to expand and improve the ways software can automatically mix music. Besides that, we know that a number of manufacturers and researchers are working on the automatic mixing of music and several different pieces of software for this technology are already present on the market.

Automatic music DJ-mixing is a technique used in software applications whose primary goal is to re-create the smooth transition between songs that DJs can provide in real world. The industry leading software tools (e.g. Native Instruments Traktor[1], Serato DJ[2], and Mixed in Key[3]) now offer users of all technical abilities the opportunity to rapidly and easily create DJ mixes out of their personal music collections, or those stored online. To this end the standard automatic mixing system

[1] *http://www.native-instruments.com/products/traktor*

[2] *https://serato.com/dj*

[3] *http://www.mixedinkey.com*

consists of the combination of beat synchronization and harmonic alignment between pieces of music. The beat synchronization process relies on the ability to robustly track tempo and identify beat locations which, when combined with an audio time-stretching operation, allow automatic beat-matching (i.e. the temporal synchronization) of music. The second important aspect related to the success of automatic music mixing, which is the main object of study of this thesis, consists in how the two songs relate to each other in terms of harmonic structure.

Specifically, harmonic mixing is a technique that aims to align tracks harmonically with each other in order to achieve a better result in terms of pleasantness of the final mix. Harmonic mixing is a relatively recent tool, in fact the traditional DJ, equipped with two vinyl record players and a mixer, did not have the possibility to align his set harmonically, with the main focus being on beat synchronization. The harmonic mixing of songs in different tempo was practically impossible. For songs in different tempo, the beat synchronization was achieved by playing one of the tracks at a slightly different speed, which in turn affected the pitch of the record being played. However, with recent development of new audio processing tools, it became possible to automatically beat-synchronize tracks to an arbitrary tempo by the use of time-stretching algorithms that do not alter their fundamental pitches. Vice versa, the ability to pitch-shift a song without changing its tempo, opened up a range of new mixing possibilities. By knowing the fundamental key of different pieces of music it is indeed possible to harmonically mix the tracks, where the goal is to create a mix of music synchronized not only in time but also in key. In order to do that, when the harmonic information on the fundamental key, is combined with an audio pitch-shifting operation (i.e. the process of transposing the pitch of a piece of music independently from its temporal structure) it provides a powerful tool to achieve the harmonic alignment between two pieces of otherwise harmonically incompatible music. From this point of view, it is possible to consider harmonic mixing as a new realm of creativity, which enables smooth transitions between songs and ensures that a larger number of musical components (i.e. vocals, melodies, and bass lines) fit together to create an enjoyable musical result.

Central to each system that attempts to create harmonic mixes of music, there is the idea to compute a measure of harmonic compatibility, which, with respect to a compatibility model, produces a measurement between songs according to how well they will sound when mixed together. This measurement is usually considered also over a range of possible pitch shifted versions of one of the tracks. Nowadays, motivated by the fact that tracks in certain musical keys are considered to be more compatible than others, the most of existing commercial applications for automatic music mixing rely on a model of harmonic compatibility which is based on fundamental key estimation and well established harmony rules of music theory. These key-based algorithms first detect the root key of one of the pieces of music and then compute a measure of compatibility by following the harmonic relationships defined in the circle of the fifths, which is the theory used by traditional musicians to create chords progressions [3]. Even if the act of mixing in key is a robust theory for expert musicians and not just a DJ tool, this kind of approach has several important limitations. Essentially, the measurement of compatibility through key estimation does not sufficiently guarantee harmonic mixing, since it ignores fine tuning aspects and mostly because assigning only one key for the duration of an entire piece of music cannot indicate where in time the best possible mixes between pieces of music might occur.

To attempt to address these limitations of the key-based harmonic mixing, the recent results obtained by the system AutoMashUpper [4] demonstrate that by expanding the ways in which the harmonic compatibility is defined, it is possible to achieve better automatic mixes of music in terms of harmonic pleasantness of the listening. The motivation which stays beyond the harmonic compatibility model of this system, follows directly from existing commercial software for harmonic mixing which, as explained above, places strong emphasis on related key. The model used in the AutoMashUpper does not consider musical key directly, but looks for a measure of similarity between beat-synchronous *chromagrams*. The chromagram is defined as the spectral representation of the twelve semi-tones of the pitch classes over time. In addition to looking at the current properties of the two songs, the model also considers how the two songs could match by applying key-transposition under a range of possible key shifts. In this way the system can increase the possibility to find pleasant harmonic matches by considering the matching process in a kind of transform domain. The mixing system based on this model produces mixes which result in a more pleasant listening when compared to those of the key-based one and, as investigate in [5], provide to overcome some of the limitations which characterize it. Although, both these harmonic compatibility methods are based on principles and rules only related to rigid music theory and thus do not take into account any information about how the perception of the final mix affected the listening.

Even if these kind of theory-based methods could be considered the most reasonable way to approach the automatic mixing, music is not just about theory and thus some psychoacoustic aspects have to be considered. By following this idea, in order to attempt to address in a more consistent way the limitations of theory-based harmonic mixing, the recent article “Harmonic mixing based on roughness and pitch commonality” [5] presents a new harmonic mixing method based both on harmonic information and psychoacoustic features. The compatibility model has his roots in established psychoacoustic principles of consonance and harmony, and the goal is to discover the optimal consonance-maximizing alignment between two pieces of music. The results of a listening evaluation test show that the most consonant alignments generated by this psychoacoustic-based method were preferred to those suggested by an existing commercial DJ-mixing system.

This psychoacoustic approach achieved better results in terms of pleasantness of the listening. Although, just as all of the existing systems that aim to produce a harmonic mix between tracks, this method is based only on a key synchronization process achieved through specific pitch-shifting operations. Even if the research field of sound engineering provides us with techniques and tools that allow to analyse, explore and specially to modify and transform the audio music signal through appropriate processing operations, none of the existing systems take in consideration the idea to modify the spectral composition of the music signals when mixing music. This last statement stands as the main motivation related to this thesis. Hence, the main purpose of this project is to expand the way in which music can automatically and harmonically be mixed together. In order to do that, we explore the existing model of roughness described in the consonance based mixing [5], which can provide us with a lot of information about how different songs relate to each other in terms of musical consonance. In the original work the consonance measurement was solely used to compute the optimal pitch-shift ratio that maximizes the pleasantness of the final mix. In our approach, by exploring the spectral content of the music signals through the use of a sinusoidal model, we investigate the evolution over time of the consonance that relates two different tracks. Furthermore, we introduce a new harmonic suppression method, which aims to reduce the dissonance of the final mix by applying precise modification on music at the signal level. The goal is to identify the most highly dissonant spectral regions in the music signal and be able to apply precise suppression operations via audio processing. Furthermore, we know that the most of existing strategies to automatic mix music are based on the idea that if the final goal is to find the optimal mix between two tracks then the information related to one of the tracks has to be fixed and is just the other one that has to be modified to reach the desired harmonic alignment. It is thus reasonable to think that, by applying some processing modifications on both the tracks instead of just one, it will be possible to achieve better results in term of pleasantness of the resulted mix. In order to reach a significant improvement, the idea is to try to modify the track which wasn't the subjected to pitch-shift operations during the compatibility measurements. By searching for those regions in the spectra that produce a significant dissonant effects when combined with the pitch-shifted version of the other track, we are able to apply a suppression method on a precise set of identified frequencies.

The designed system takes as input two tracks from a defined set of music samples. In order to focus on the harmonic structure present in the musical input, the algorithm first extracts a set of partials describing the harmonic content by means of the sinusoidal analysis model. Afterwards, in order to be able to modify and then reconstruct a masked version of one of the input tracks, the algorithm runs the selected input into a Sines Plus Residual model [6]. This model allows us to separate the harmonic content, which will be the object of the investigation, from the residual part, which, when combined with the reshaped sines, will be used to synthesize a modified version of the song. The last step of the pre-processing operations, in order to reduce the computational complexity, is to average the harmonic content over sixteenth note temporal frames. Then, according to the psychoacoustic model of sensory roughness [5], the system provides the optimal pitch-shift ratio between the tracks that maximizes the consonance of the resulting mix. Once the optimal pitch shift is computed, the algorithm, by exploring the spectral content within each time frame, compares the frequency partials of one track with the pitch shifted version of the other one and identifies the set from the fixed track which produce a significant dissonant effect on the final mix. The last step of the process is to identify, within the same time frame, if the selected partials are somehow part of a harmonic pattern. If the algorithm finds in the frequency spectra a harmonic structure somehow related to the selected partial, it identifies the other partials as to be suppressed. Once the selection process is complete, the algorithm applies a suppression operation on the selected partials of the original (i.e. not averaged) harmonic content. The final post-processing operations consist in the re-synthesis of the masked track, and the computation of the pitch shifted version of the other one.

The remainder of this thesis is structured as follows: in Chapter 2 we provide an overview of the state of the art and the underlying music analysis techniques for automatic music mixing. Chapter 3 gives an overview of the theoretical methods used in this thesis to design the system related with the project. In Chapter 4 we provide a detailed description on the implementation of the system and in Chapter 5 we give an explanation of the experiment that we have run in order to evaluate the method. The last chapter is Chapter 6, where we define the final conclusions of this work and some ideas for future developments. This thesis ends with appendices and bibliography.

Chapter 2
State of the Art and Motivation

In this chapter we provide an overview of the literature related to the existing software for the harmonic DJ's mixing of music and for the compatibility measurement between songs. Starting from the role of the DJ and the impact of digital music on the DJ's workflow and practice, we explore the principles of western tonal music theory relevant to the concept of key classification, and present an overview of the most recent software for DJ's mixing based on audio signal processing.

2.1 The figure of the DJ

Before the 1940s, the DJ was exclusively a radio employee, who was supposed to introduce records and talk to the listening audience. Dancehalls and nightclubs, where people had the possibility to listen and dance to music, were venues solely for live performers. In the 1940s, taking some inspiration from the jukebox and some from the "commercial possibilities of a band-less dance" [7, p. 54], it started to rise the figure of the DJ as we know it today, and the practice of DJing began to evolve beyond the radio presenter style. In 1943, the world's first DJ dance party took place, when jazz records were played in the upstairs function room of the Loyal Order of Ancient Shepherds in Otley, England. 1947 saw the first DJ to use twin turntables for continuous play [7, p. 82]. Also in 1947, in Paris opened the world's first commercial discothèque, or disco (deriving its name from the French word meaning a nightclub where the featured entertainment is recorded music rather than an on-stage band). Since then, the figure of the DJ was constantly evolving, discos began appearing across Europe and the United States, and many DJs have employed a variety of techniques to mix music, using existing records as raw materials to create something new. Twenty years later, in 1973, Jamaican-born DJ Kool Herc, widely regarded as the "father of hip-hop culture," performed at block parties in his Bronx neighbourhood and developed a technique of mixing back and forth between two identical records to extend the rhythmic instrumental segment, or *break*. This signed the beginning of Turntablism, which consist in the technique of using turntables not only to play songs but also to manipulate sound and create original music [8].

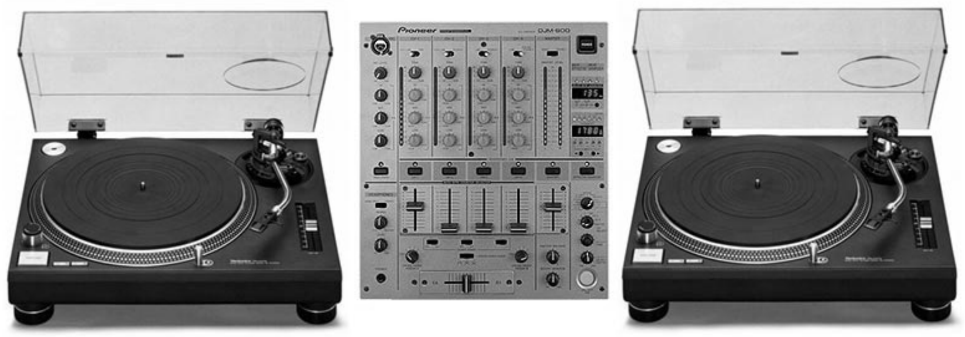


Figure 2.1: DJ vinyl traditional setup, consisting of two turntables and a crossfader mixer. Image from Google.

Nowadays, the standard basic equipment for the DJ set up (Fig. 2.1) is a pair of turntables and an audio mixer [7]. The mixer must be able to output either or both of the turntables' signals to the main sound system, and must also allow the DJ to hear either source independently of the main output (typically through headphones). It is standard for the turntables to include some control over the speed of the motor, allowing the music to be played faster or slower. With this level of control, by manual manipulation of the turntable platter, the DJ is able to synchronize a record with the beat of the track currently playing to the audience [7, p. 108]. The DJ is then allowed to use the controls of the mixer to quickly make a transition or cut from the playing record to the other without a break in the rhythm of the music. Alternatively, he can perform a longer mix, playing the synchronized records together to create new sounds. The range of control over a turntable's speed is limited. The industry standard Technics SL-1200 MK2 allows a deviation up to ±8%, so for example it is not possible to rhythmically mix a record at 90 bpm (beats per minute) with one at 120 bpm, even if they are played at opposite limits of the speed control. A significant effect of the speed change that must be considered is that a record played too fast or too slow plays at a noticeably higher or lower pitch [8]. This is not usually a problem for noisy percussive sounds like drums, but tonal sounds like singing may become comical or unpleasant.

2.2 DJ evolution with the advent of digital music

With the arrival of digital audio some DJs began to migrate to software equivalents of the turntable and mixer setup. A common reason for this initial migration is that a laptop storing digital recordings is easier to carry around than cases full of vinyl records. Many DJs continued to use hardware devices, including turntables, to maintain tactile control over their computerized performances, but such implementations are not the object of study for this project. As mentioned in the introduction, the standard approach to the problem of automatizing the smoothly mixing of different songs by software is built around the ability to robustly identify tempo and beat locations. This information when combined with high quality audio time stretching, allow for automatic “beat-matching” (temporal synchronization) of different pieces of music. Although it is a fundamental step, the automatic beat-matching process is not the object of investigation for this thesis. In fact, as DJing software evolved, the technical capabilities available to the computer-using DJ improved beyond what was possible with traditional equipment. The specific innovation that motivates this project is generically referred as the harmonic mix. In fact, high performance software for automatic music DJ mixing have started to investigate the relations which occurs between the harmonic structures and melodies of different songs. With this type of approach, combined with the new technology and audio processing tools, it is possible to extract specific harmonic information from the music signal and be able to compute and apply precise processing operations on the investigated pieces of music.

2.2.1 Time scaling

Specifically, DJing software and hardware began to implement algorithms that allowed a song to be played faster or slower without significantly affecting the sound of the recording. These time-scale modification procedures are digital signal processing methods for stretching or compressing the duration of a given audio signal. Ideally, the time-modified signal should sound as if the original signal’s content was performed at a different tempo while preserving properties like pitch and timbre. A main challenge for time stretching algorithms is that music signals are complex sound mixtures, consisting of a wide range of different sounds. Preserving these contrasting characteristics usually requires conceptually different time-scale approaches.

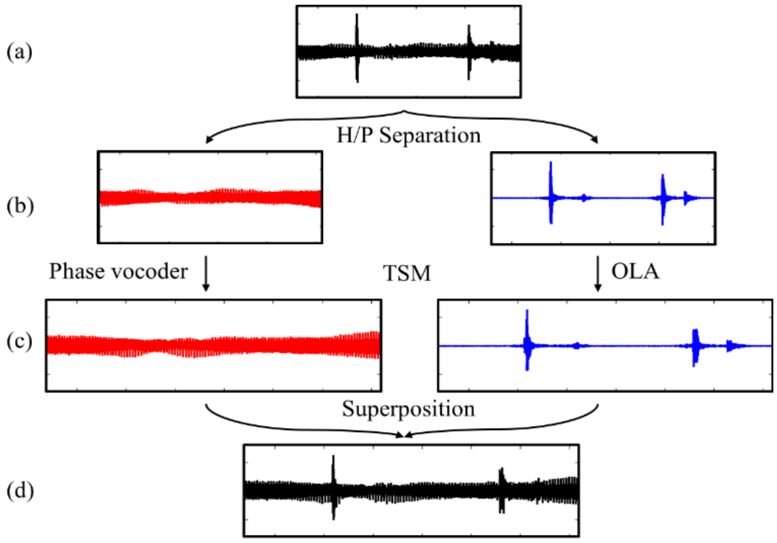


Figure 2.2: Overview of the combined time-stretching approach. (a) Input music recording. (b) Separation in harmonic (left) and percussive (right) components. (c) Time-stretching results for the harmonic content using the phase vocoder (left) and for the percussive component using OLA (right). (d) Superposition of the time-stretching results from (c). Image from [9].

For example, classical procedures based on waveform similarity overlap-add (WSOLA) [10] or on the phase vocoder (PV) [11] [12] [13] are capable of preserving the perceptual quality of harmonic signals to a high degree, but introduce noticeable artifacts when modifying percussive signals. However, it is possible to substantially reduce artifacts by combining different time stretching approaches. For example, in [9], a given audio signal is first separated into a harmonic and a percussive component. Afterwards, each component is processed with a different time-scaling procedure that preserves its respective characteristics (Fig. 2.2). The final output signal is then obtained by superimposing the two intermediate output signals. This new processing tool when applied in the mixing software for DJ gives users the possibility to automatically beat-synchronize tracks to an arbitrary tempo without altering their fundamental pitches.

2.2.2 Pitch scaling

Time stretching algorithms are not the only new technology that contributed to the rise of new mixing techniques. In fact, the inverse ability to pitch-shift a song without changing its tempo, opened up a range of new mixing possibilities. Pitch scaling is the dual operation of the time scaling; in this case the aim is to change the frequency content of a signal without affecting its time evolution. By changing the estimated key of different pieces of music without affecting the tempo, it is then possible to harmonically mix the tracks, where the goal is to create a mix of music synchronized not only in time but also in key. With this powerful tool it became then possible to mix a much wider range of recordings, not only considering their rhythm, nut also their tone.

In general, pitch-shifting algorithms can be divided into two categories; time-domain and frequency-domain techniques [13]. Time-domain techniques are simple and fast, and work fine for periodic and quasi-periodic signals. The standard time-domain pitch-shifting algorithms, commonly used in commercial applications, are based on resampling and time-scale modification [13] [14] (Fig. 2.3). From the basis of signals theory, we know that exists a principle of duality between time and frequency. As might be expected, this is true for both time and pitch scaling. Consequently, performing a time warping operation (i.e. resampling in discrete time) after a time scaling on the signal leads to a new signal that corresponds to the pitch scaled version of the original one. However, the quality of time-domain methods is not good for signals that contain a lot of non-harmonic components. On the other hand, frequency-domain algorithms are more suitable for complex signals, such as music is, but the price of high quality is the computational complexity. Additionally, frequency-domain pitch-shifting algorithms call for large delays and thus, are not appropriate for real-time applications [15]. Frequency-domain algorithms are usually based on the phase-vocoder [11] [12]. In the phase-vocoder technique, the signal is first converted to its frequency-domain representation using a short-time Fourier transform (STFT). After specific modifications of the frequency-domain parameters according to the pitch-shifting factor, the signal is converted back to its time-domain waveform.

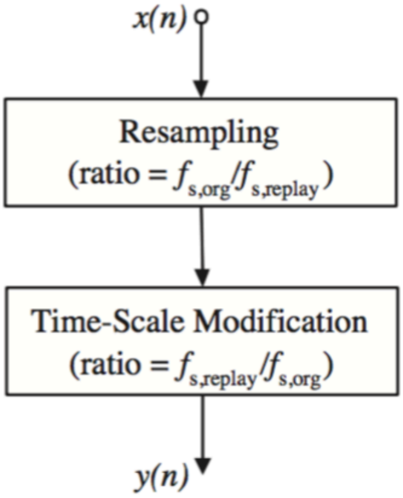

x(n)
Resampling
(ratio = $f_{s,org}/f_{s,replay}$)
Time-Scale Modification
(ratio = $f_{s,replay}/f_{s,org}$)
y(n)

Figure 2.3: Block diagram of the pitch-shifting method based on the resampling and time-scale modifications. In this figure, $f_{s,org}$ and $f_{s,replay}$ are the sampling frequency of the original audio and that of the pitch-shifted signal, respectively. Image from [15].

2.3 Harmonic mixing methods

We now provide an overview over the most significant and diffused methods for the harmonic DJ mix of music. Starting from a description of the standard harmonic compatibility model, constructed around key estimation and circle of fifths, and finishing describing the most recent approaches (Section 2.2.2 Chroma based approach. Section 2.2.3 Consonance based Mixing).

The set of existing software that allow users to create automatic music mixes includes both research-based and commercial systems. Even if it is not a deeply explored field, the research-based approaches address some important technical aspects of music mixing and mashup creation. These, as stated in [4], include interactive user interfaces [16], computational feasibility of time-stretching multiple songs in parallel [17] and the need for accurate tempo synchronization for user appreciation [18].

2.3.1 Harmonic mixing based on key estimation

Each system that aims to provide a harmonic mix of music, have somehow to relate with the harmonic information of the music signal. Typically, in the existing commercial software, due to the fact that rules of music theory define how songs in different musical key relate to each other in terms of harmonic affinity, this information is in the form of an estimated key. The idea behind these applications is to analyse the harmonies and melodies of the music, show the estimated fundamental key of every track, and then helps the user to mix tracks that are harmonically compatibles with each other. In fact, by knowing the key of different pieces of music, these systems aim at align music not only in time but also in key. Indeed, when the key estimated information is combined with audio pitch-shifting functionality it provides a powerful means to match the harmonic alignment between two pieces of otherwise incompatible music.

The key-estimation approach to harmonic mixing consists of applying the knowledge of music theory to find songs in matching or related key. The compatibility model beyond this method has the advantage of being very simple, it mainly consists of two elements: the key estimation process on the pieces of music intended to be mixed and the theoretical estimation of which keys are compatible with it.

2.3.1.1 Key estimation algorithms

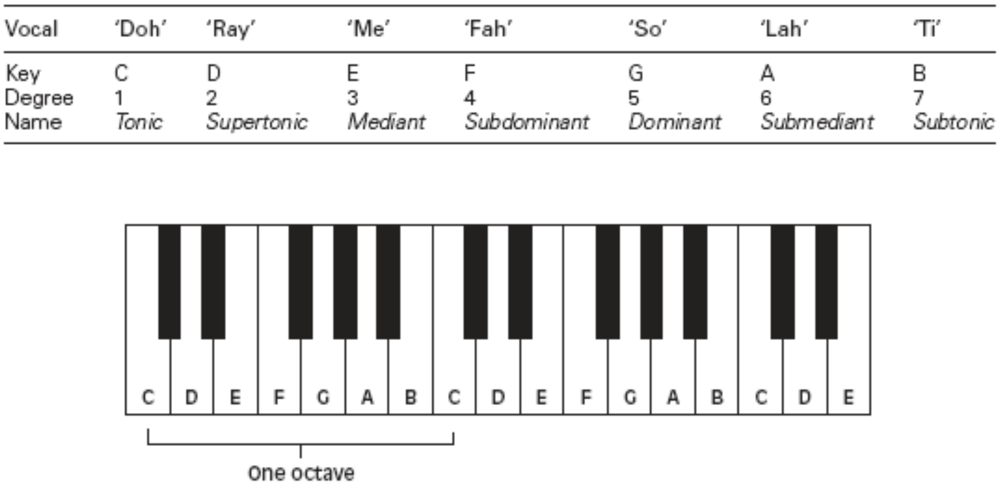
Key finding, as it relates to Western tonal music, is the problem of detecting the key in terms of the most stable pitch, called the *tonic*. Applications of key finding in music understanding reach several different fields as automated music analysis, machine learning, music perception and music information retrieval (e.g. search/query music databases, playlists generation or automatic accompaniment). Considering its numerous possible applications besides software DJ mixing, the automatic estimation of musical key has received a considerable attention in the recent years.

Many models that address the problem of key finding have been reported in the literature and these can be classified into two main groups. The first approach aims at performing transcription to convert audio data into symbolic form before applying a symbolic algorithm. Given that the accuracy of automatic transcriptions is not satisfactory this category has to deal with incomplete, missing or probabilistic data. Furthermore, the existing algorithms for music transcription are still limited and costly. The second approach attempts to estimate the key directly from operations on the audio data. This is a new set of techniques in audio signal processing which take a music signal as input and attempt to find the key of that signal.

From the first group there has been relatively less research than the second. A significant approach is the one from Chew [19], which uses a geometrical representation called the Spiral. As stated in [20], for the second group, many models have been proposed. A few examples demonstrating the variety of approaches are included here. Huron and Parncutt [21] used a psychoacoustic model of pitch perception that employed echoic memory and pitch salience to model key perception. Gómez and Herrera [22] present a comparison of cognition-inspired models based on Krumhansl's method [23]and feature-based machine learning methods for key finding from polyphonic audio. Chuan and Chew's model [24] estimates pitch strength using peaks in the spectrum, which are then used by the Spiral Array model to estimate key. Each of this approaches have its own advantages and disadvantages, but due to the nature of this project we do not proceed on investigating these techniques, we just mention that there are several strategies to achieve the estimated pitch detection.

2.3.1.2 Music theory review

In order to understand how two different sounds are considered harmonically aligned by means of music theory, it is necessary to first give a short overview on what consonance and harmony mean in a musicological sense. This review of music theory is limited to modern, western, tonal music, since this has the most relevance with respect to the investigated subject of this thesis. The principles of this theory do not hold for all historical musical traditions, or for all modern music, especially in different cultures. Besides that, we start to focus our attention on the chromatic scale, a representation of a set of 12 discrete musical intervals or semitones that form a musical octave (Fig. 2.4).



Vocal	'Doh'	'Ray'	'Me'	'Fah'	'So'	'Lah'	'Ti'
Key	C	D	E	F	G	A	B
Degree	1	2	3	4	5	6	7
Name	*Tonic*	*Supertonic*	*Mediant*	*Subdominant*	*Dominant*	*Submediant*	*Subtonic*

Figure 2.4: Notes in an octave (from C to B), and layout of a musical keyboard (2 complete octaves shown) [image from Google].

Two notes at one octave interval are seen as harmonically equivalent and have the same musical note. The interval of an octave is, physically speaking, the doubling of frequency for an octave up and, respectively, the halving for an octave down. Other important musical intervals are the fifth (frequency ratio 3:2), the fourth (4:3), the major and minor thirds (5:4, 6:5) and the major and minor sixths (5:3, 8:5). Depending on intonation, the ratios of intervals may differ. However, since different tunings tend to produce differences that are largely not audible [25] and the consideration of alternate tunings goes beyond the scope of this work, intonation will not be taken into account. Generally speaking, in music theory, the smaller the ratio numbers, the higher is the consonance of the interval [26]. Therefore, different keys show the characteristic of fitting more or less well to each other. These relations are intuitively represented by a powerful tool for music composition, which is called the circle of the fifths (Fig 2.5).

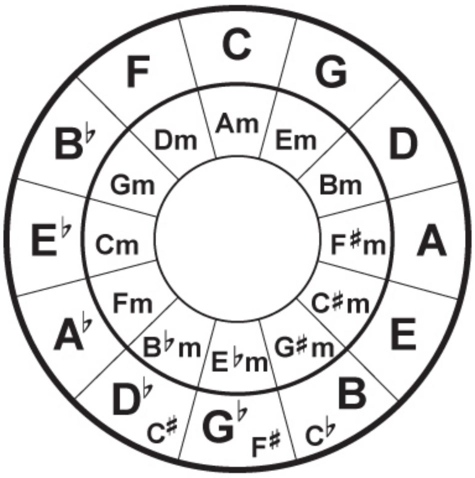


Figure 2.5: Circle of fifths, showing major and minor keys. The majors are on the outside circle while the relative minors in the inner one. [Image from Google].

In music theory, the circle of fifths is a visual representation of the relationships among the 12 tones of the chromatic scale, their corresponding keys, and the associated major and minor keys. This composer's tool finds its first appearance in a treatise on composition written by the composer and theorist Nikolai Diletskii [27]. The author intended the circle of the fifths to be a guide to composition but pertaining to the rules of music theory. Indeed, the circle of fifths is a sequence of pitches or key tonalities, represented as a circle, in which the next pitch is found seven semitones higher than the last, which allows establishing a criterion for the compatibility of different keys.

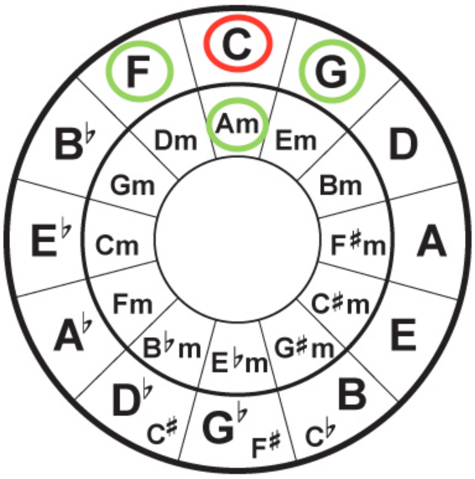


Figure 2.6: In the case of C major the keys to the side are F major and G major. The key exactly below it is A minor. The C is thus compatible with the keys in its immediate vicinity F, Am and G, the ones it shares a border with.

The criteria to identify which keys sounds harmonically good with a chosen one works as follows: once the root key of the song is detected and localized on the diagram, then that song is compatible with songs that have the same key, or keys that are to the side of the selected one, or the key exactly below or above its on the diagram (Fig. 2.6).

Commercial DJ-software like Native Instruments' Traktor Pro 2, takes use of this theory to estimate the harmonic suitability of the tracks to be mixed together by key detection. Musical tracks are analysed by their key and for each of them a number from 1 to 12 is given. This number corresponds to the same in the circle of fifth's relative major/minor pairs (Fig. 2.7). This is to help the user to rapidly see if two tracks fit well together by just looking for a pair with two numbers that are equal or have the distance of one (corresponding to one step on the circle of fifths).

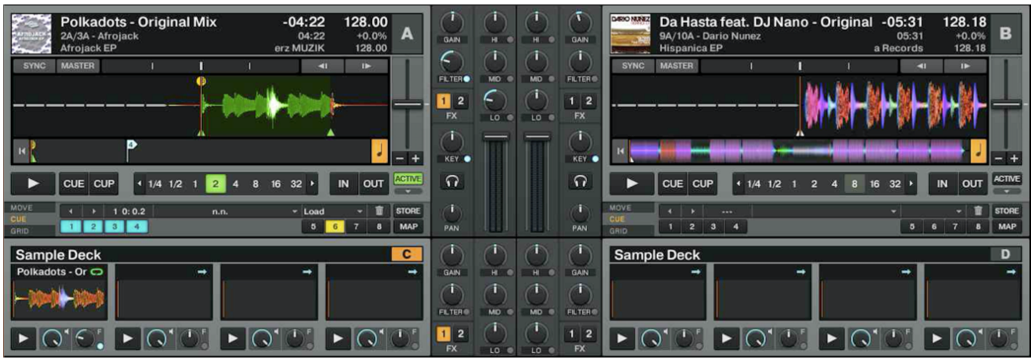

Polkadots - Original Mix -04:22 128.00
2A/3A - Afrojack 04:22 +0.0%
Afrojack EP erz MUZIK 128.00
A
Da Hasta feat. DJ Nano - Original -05:31 128.18
9A/10A - Dario Nunez 05:31 +0.0%
Hispanica EP a Records 128.18
B
SYNC
MASTER
CUE
CUP
1/4 1/2 1 2 4 8 16 32
IN
OUT
ACTIVE
MOVE
CUE
GRID
STORE
MAP
Load
GAIN
HI
MID
LO
FILTER
FX
KEY
PAN
Sample Deck
C
Polkadots - Or
Sample Deck
D

Figure 2.7: Screenshot of the DJ software "Native Instruments Traktor pro 2" with 2 parallel decks. The analyzed key is displayed beneath each trak's title in its tonic note, in numbers and chord. Image from [http://www.native-instruments.com/products/traktor].

2.3.1.3 Problems and limitations

Even if the model of compatibility built around this method results in producing mixes which sound well in term of pleasantness of the listening, we already stated that this model has several important limitations. The first aspect to be considered is that the key estimation process itself, not being an exact science, might lead to some errors. Furthermore, a property of music such as the root key does not provide any kind of information about the musical composition that led to that key, nor how this might affect perceptual harmonic compatibility for listeners when two pieces are mixed [5]. Similarly, music matching based on root key estimation does not provide an obvious means for measuring the compatibility between different pieces of the same key and is restricted to major and minor keys. Any track that has not been composed in this traditional way will likely cause the key detection to fail. Even if the two keys are correctly detected, there is still the unsolved problem of fine-tuning, two pieces of music might have the same key, but their relative pitch might be slightly different. Moreover, one of the most critical limitations is that due to the simple nature of its compatibility measurement, this approach has the strong disadvantage of limiting the knowledge to following a defined scheme, it will not get any further. In fact, it is able to make the system work musically, but does not allow us to advance in either harmonic mixing knowledge or music theory.

2.3.2 Chroma-based mixing

The recent application *AutoMashUpper* [4] which allow users to create mashups between different songs (i.e. take two or more pre-existing pieces of music and combine them to make a new song), tried to address the limitations of the key-based harmonic mixing and introduced a new harmonic compatibility model which does not consider key estimation directly. This kind of approach exploits the power of audio signal processing; indeed, the measurement of harmonic compatibility is calculated by comparing together beat-synchronous chromagrams under a range of possible key transposition.

2.3.2.1 Chromagram

Chroma features are an interesting and powerful representation for music audio signals that have been used in key finding [22] [28], discovering similarity and repetition in audio recordings [29], and chord segmentation, recognition and alignment in audio [30].

A chromagram (or pitch class profile) is a popular feature in music digital signal processing (DSP), in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave (Fig 2.8). Introduced by Fujishima, he originally proposed the Pitch Class Profile (PCP) to be used in chord recognition [31]. His algorithm is based on training the system with synthesized chords and determining the nearest chord to the calculated PCP from the input. Since, in music, notes exactly one octave apart are perceived as particularly similar, knowing the distribution of chroma even without the absolute frequency (i.e. the original octave) can give useful musical information about the evolution of a musical signal [32], and may even reveal perceived musical similarity that is not apparent in the original spectra.

There are several different techniques to obtain a 12-bin chromagram, but it usually proceeds from the following steps. As explained in [33], first, the DFT (Discrete

Fourier Transform) of the input signal $X(k)$ is computed, and the constant Q transform $X_{C,Q}$ is calculated from $X(k)$, which uses a logarithmically spaced frequency to reflect the frequency resolution of the human ear. The frequency resolution of the constant Q transform follows that of the equal-tempered scale, and thus the k_{th} spectral component is defined as:

$$f_k = (2^{1/B})^k f_{min} \, , \tag{2.1}$$

where $f_k$varies from $f_{min}$to an upper threshold frequency, and B is the number of bins in an octave in the constant Q transform. Once $X_{C,Q}(k)$ is computed, a chromagram vector CH can be easily obtained as:

$$CH(b) = \sum_{m=0}^{M-1} \left| X_{C,Q}(b + mB) \right|, \tag{2.2}$$

where $b = 1, 2 \dots, B$ is the chromagram bin index, and M is the number of octaves spanned in the constant Q spectrum.

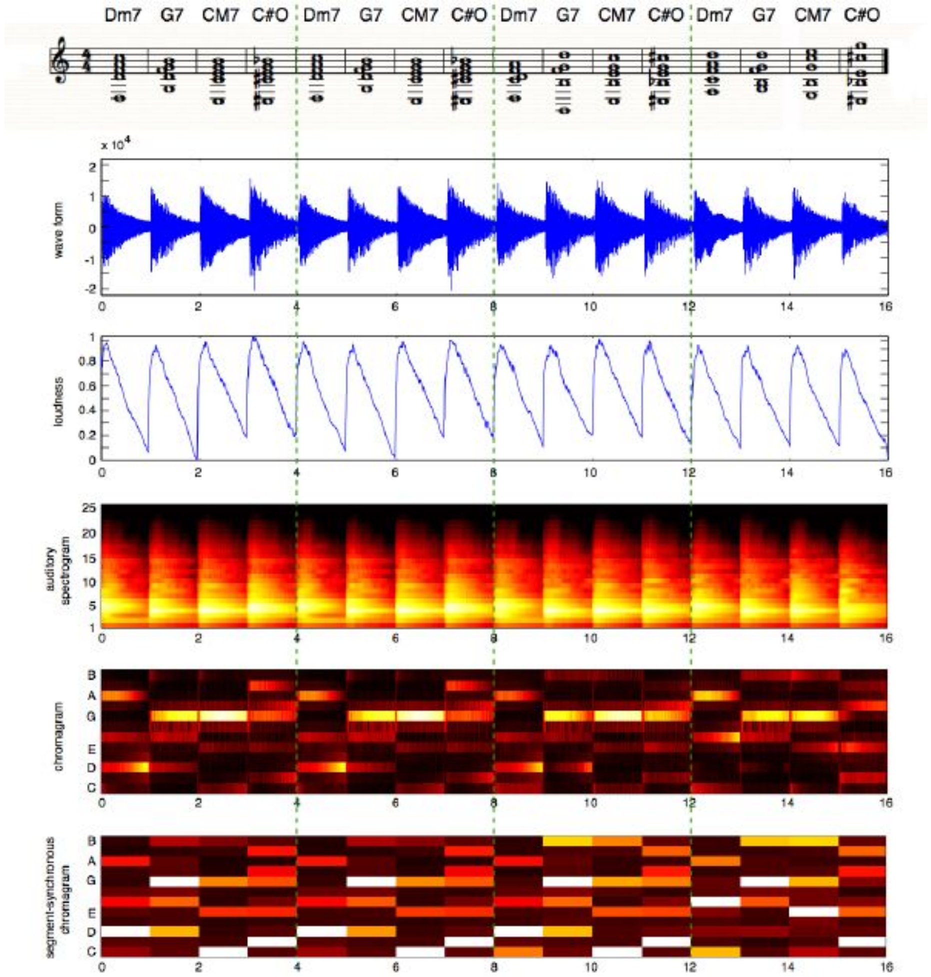


Figure 2.8: Typical progressions played on a piano. No distinction can be made by looking at the waveform, the loudness curve, or the 25-critical band auditory spectrogram since this is only "one" timbre. However, the chromagram allows us to differentiate chords and recognize similarities between chords of the same class even when they are inverted. The last pane shows the chromagram computed on a per-segment basis, while the previous one is computed every 6 ms with a 93-ms long window. Image from [34]

2.3.2.2 Chroma-based compatibility model

In the *AutoMashUpper* [4], the measurement of harmonic compatibility is calculated by comparing together beat-synchronous chromagrams. The system, taking as input two tracks T1 and T2, calculate the combination between the pitch-shift ratio and the time instant that maximizes the harmonic relationships of the two tracks.

The measure of compatibility works as follow: first, the beat-synchronous chromagram C_1 of a piece of music T1 is calculated and then stacked by median averaging across beat windows. The algorithm then, for a second given candidate piece (T2) of a music collection, with beat corresponding synchronous chromagram

$C_{2,k}$, attempts to identify the temporal location k (i.e. starting beat) of the chromagram which maximized the harmonic similarity between the tracks. Furthermore, in addition to looking at the current properties of the two spectral representations, the model considers how the songs could relate to each other after a key-transposition process (i.e. pitch-scaling operations). In this way it increases the possibility of finding good harmonic alignment by considering the key-synchronization task in a sort of transform domain. In their work, the authors implemented this approach by measuring the cosine similarity across a range of rotational pitch shifts q on the chromagram of T1:

$$H_2(q,k) = \frac{C_{2,k} \cdot C_{1,q}}{\parallel C_{2,k} \parallel \parallel C_{1,q} \parallel}, \tag{2.3}$$

where $C_{2,k}$ is the compared chromogram of the other track T2. Given the cosine similarity matrix $H_n(q)$, the final step is to extract the maximum harmonic similarity $M_{H,n}(k)$ and to record the corresponding key shift $Q_n(k)$ necessary to realize this similarity value:

$$M_{H,2}(k) = \max\left(H_2(q)\right) \tag{2.4}$$

and

$$Q_2(k) = -arg\max\left(H_2(q)\right). \tag{2.5}$$

Where the transposition is negative because the algorithm later applies a pitch-shift operation on the second track to match the selected one, which is the reverse harmonic matching process. For a graphical representation of the chroma-based harmonic mixing see Figure 2.9.

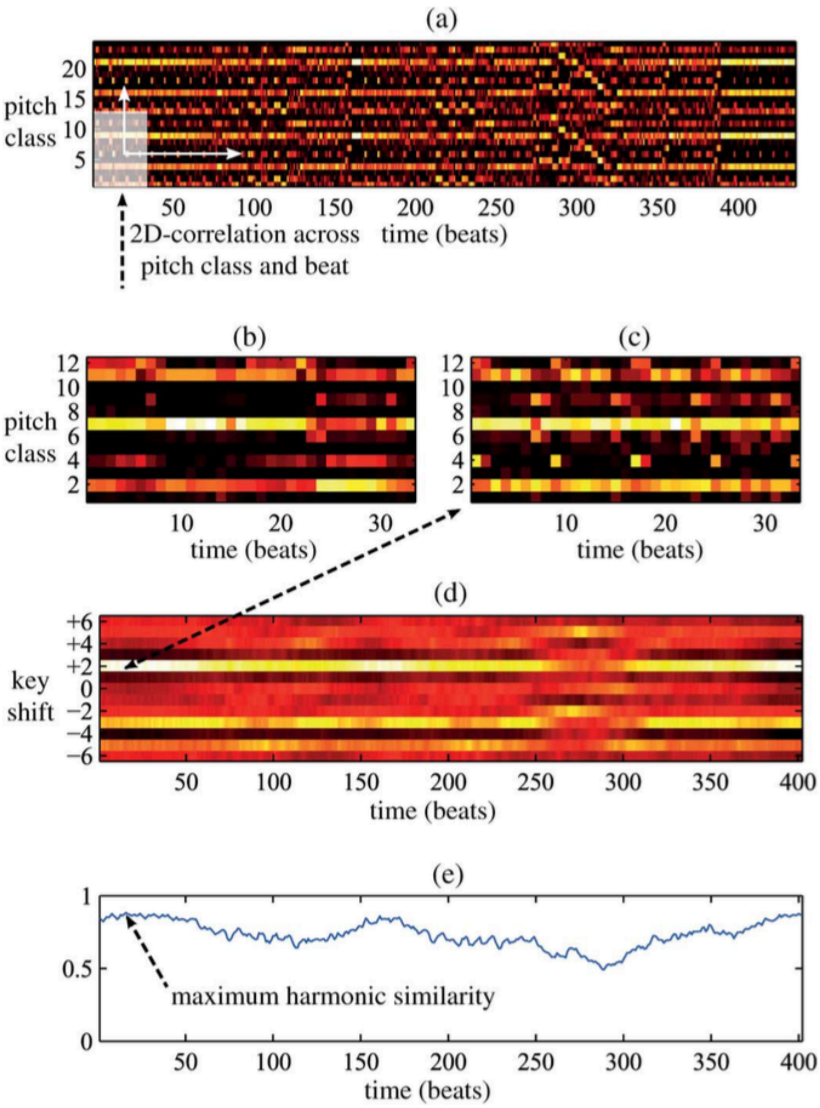


Figure 2.9: Harmonic matching overview. (a) Stacked candidate song T1 chromagram, (b) chromagram for a 32-beat section of T2, (c) the corresponding best matching chroma patch (after a +2 semitone pitch shift) extracted as the global maximum of (d) the harmonic similarity matrix, and (e) the overall maximum harmonic similarity per beat shift $M_{H,2}(k)$. Image from [4]

2.3.2.3 Problems and limitations

The main novelty introduced by this approach, is the idea to expand the harmonic compatibility measurements by taking in consideration complex spectral information (chromagram) about the music content extracted through specific audio processing tools. From the positive results obtained by this system it easy to understand that, in the process of harmonic mixing, by increasing the complexity of the information extracted from the spectral analysis, it is possible to obtain detailed information about the harmonic relations that occur in the music signal.

Even if this new type of approach is actually a great step forward in the evolution of the digital harmonic mixing, it has a significant limitation. The more general problem comes about due to the discretization of the chromatic scale. Even if the two keys are correctly detected, there is still the unsolved problem of fine-tuning: two pieces of music might have the same key, but their relative pitch might be slightly different. If we assume the width of the categories that represent the tonal not of a key to be half a semitone down and up respectively, the maximum error mounts up to a whole semitone. The interval of a diminished second is not only seen as the most dissonant from a musicological point of view, it also catches a high value of dissonance in psychoacoustic models. From this last statement, we can state that the most significant limitation for this type method, as much as for the approach based on key estimation, is that it only takes into account matching criteria strictly related to harmony theory and musicological principles. As a matter of fact, we do believe that music is not just about theory and thus, especially in the context of harmonic mixing, the way we perceive sound plays a fundamental role in the definition of a measurement of harmonic similarity. However, the psychoacoustic aspect of harmony is rarely taken under consideration in the literature of harmonic mixing.

Psychoacoustic theory represents a promising alternative to classic musical theory for harmonic mixing. First and foremost, this is because it can be applied in a way that solves the main problems of the model of key detection. The physical correspondence to the musical chromatic scale is the frequency scale, which all psychoacoustic models are primarily based upon. Its continuity allows a subdivision into arbitrarily small intervals, which is an important advantage when it comes to fine- tuning. This means that the alignment of pitch of two audio tracks is no longer restricted to semitone-categories, but can be refined as needed for the application. Furthermore, there is the new possibility, not only to analyse music in major or minor scales, as none of the psychoacoustic approaches take musical key into account.

2.3.3 Consonance based mixing

By following this idea, in the recent article "Harmonic mixing based on roughness and pitch commonality" [5], a new harmonic mixing method based on the combination of psychoacoustics features and harmonic information of the music signal is introduced. In order to address the limitations of a harmonic mixing process based only on music theory criteria, in this article the authors propose a new approach based on a harmonic compatibility model built around the measurement of musical consonance at the signal level.

The idea is to introduce psychoacoustic principles of sensory consonance in the measurement of harmonic compatibility, where the goal is to discover the optimal consonance-maximizing alignment between two music excerpts. To this end, the algorithm first extracts a representation of the music's harmonic content using a sinusoidal model and then, after fixing the data of one excerpt, applies a logarithmic scaling to the information of the other one over a range of possible pitch shift. Through an exhaustive search the system can identify the frequency shift that maximizes the consonance between the two excerpts and then applies the appropriate pitch-shifting factor prior to mixing the two excerpts together. For a graphical representation of the consonance based harmonic mixing see Figure 2.10. With this kind of implementation which searches across a wide range (+- 6 semitones) of possible pitch shifts in small intervals (1/8th semitone steps), it is possible to investigate over a large number of potential harmonic alignments without allowing hypothetical differences in tuning.

In this system, the theoretical approach to the computational measurement of consonance needed to estimate the most harmonic combination of two tracks is defined by means of Terhardt's psychoacoustic model [35]. By following this interpretation, consonance is defined as the union of two different categories. The first, *sensory consonance,* which as in the most of existing consonance models can be approximated as *roughness*. And the second harmony, which is mostly built upon Terhardt's virtual pitch theory and inherits *root relationship* and *pitch commonality*. These two categories stand as the substructure of the consonance compatibility model described in this system, and thus the harmonic similarity is computed as the combination of a roughness measurement and a pitch-commonality evaluation.

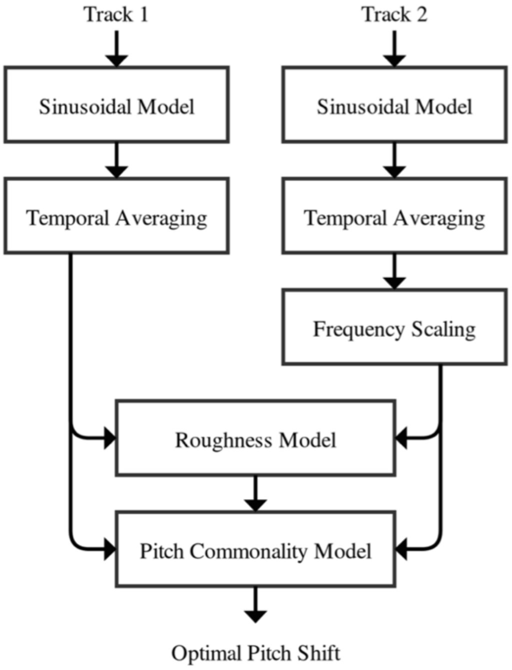


Figure 2.10: An overview of the approach for consonance based mixing. Image from [4].

2.3.3.1 Roughness model

In order to estimate a measure of sensory consonance, the authors used a modified version of Hutchinson & Knopoff's [36] roughness model. In this model, for each of the partials extracted from the spectrum of the music signal, the roughness that is evoked by the co-occurrence with other partials is computed. The basic structure of this procedure is a modified version of Hutchinson & Knopoff's [36] roughness model for complex sonorities that builds on the roughness curve for pure tone sonorities proposed by Plomp & Levelt [37].

Parncutt in [38] proposed a function that approximates the graph estimated by Plomp & Levelt

$$g(y) = \begin{cases} \left(\exp(1) \dfrac{y}{0.25} \exp\left(-\dfrac{y}{0.25}\right) \right)^2 & y < 1.2\,; \\ 0 & otherwise \end{cases} \tag{2.6}$$

where y is the frequency interval between two partials (f_i and f_j) expressed in the critical bandwidth (CBW) of the mean frequency and $g(y)$ is the degree of roughness between a pair of partials. The roughness values $g(y)$ for every pair of partials are then weighted by the amplitudes (M_i and M_j) to obtain a value of the overall roughness D.

2.3.3.2 Pitch-commonality

On the other side, for calculating the pitch commonality of a combination of sonorities, the authors proposed a model that combines Parncutt & Strasburger's [39] pitch categorization procedure with Hofmann-Engl's [40] virtual pitch model. The roots of this approach lie in identifying harmonic patterns over the spectrum of the music signal. The extraction of these patterns is taken from the pre-processing stage of the pitch categorization procedure of Parncutt & Strasburger's [39] tonalness model. As a result, in this implementation given a set of extracted partials, the audibility of pitch-categories is produced. Since this corresponds directly to the notes of the chromatic scale, the degree of audibility for different pitch-categories can be attributed to a chord. Hofmann-Engl's [40] virtual pitch model then is used to compute the "Hofmann-Engl pitch sets" of these chords that will be compared for their commonality.

2.3.3.3 Consonance based alignment

Based on the presented models of roughness and pitch commonality, this system introduces an original approach for harmonic mixing two pieces of music. Specifically, for two input musical excerpts, the algorithm seeks to find the optimal consonance-based alignment between them. To this end, the system fixes all the information regarding one track and attempts to modify the other. This approach is centred on the calculation of consonance as a function of a frequency shift, and is based on the hypothesis that under some frequency shift applied on one of the tracks, the consonance of the final mix is maximized, and this, as a consequence, will lead to the optimal mix between the two excerpts.

The system, taking as in input two tracks T1 and T2, calculates a measurement of the sensory consonance that occurs between each pair of beat-synchronous time frames, using the roughness model described above. Then, to calculate the overall roughness as a function of frequency shift, it averages the roughness values across the whole time length of the track. Once it has calculated the roughness across all possible frequency shifts, it then turns the focus towards the measurement of pitch commonality. However, due to the high computational cost of the pitch commonality model, the algorithm does not calculate the harmonic consonance as a function of all possible frequency shifts. Instead it extracts a set of local minima from the roughness measurement (Fig. 2.11), mark these frequency shifts, and then proceed with the pitch commonality evaluation only over this subset.

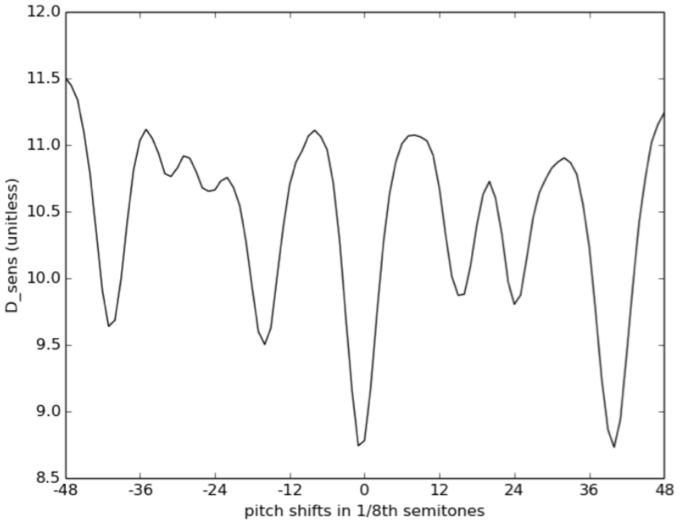


Figure 2.11: Roughness curve generated by the sensory dissonance model for two different pieces of music. The global minimum at pitch-shift index 40 indicates the result with the highest sensory consonance. Image from [5].

The overall harmonic consonance can be then calculated for each possible selected pitch-shift by averaging across the temporal frames. Finally, since no prior method exists for combining the roughness and harmonic consonance, the system adopts a simple approach to equally weigh their contributions to give an overall measure of consonance based on roughness and pitch commonality by simply adding the two different measurements. Once the overall measure of consonance is computed the system can identify the frequency shift which maximizes the consonance between the two excerpts and then apply the appropriate pitch-shifting modification on one of the two input songs.

2.4 Motivation

As evidenced in this literature review, the state of the art system for software DJ mixing has some important limitations. The most significant aspect that we want to underline is that the existing applications and systems which aim to produce an optimal mix between tracks are based only on processing operations which consist in finding the optimal beat-matching asset, combined with a key synchronization process through specific pitch-shifting operations. Indeed, both the key estimation based mixing, as well as the chroma-based approach and the consonance based model belong to this category. Specifically, when relating with harmonic DJ mixing, these systems only provide the user with the optimal pitch-shift, which, with respect to a harmonic compatibility model, hypothetically maximize the harmonic alignment between different tracks.

2.4.1 State of the problem

Even if in the research field of sound engineering it is well known that is possible to analyse, explore and especially, modify and transform the audio music signal through specific processing operations, none of the existing systems take in consideration the idea to modify the spectral composition of the piece of music when mixing it.

Starting from this point of view, when the purpose is to find the optimal harmonic mix between tracks, we assume that deconstructing the music signal and then re-synthetizing it after some specific modifications, could lead to a better final mix in terms of pleasantness of the listening. This last statement, as stated above in the introduction, stands as the main motivation for this thesis. By following this idea, one of our main aims in this work is therefore to use the existing theoretical background to develop a model that, starting from two music signals, does not only analyse the harmonic content to compute the optimal pitch-shift, but also applies specific processing operations to modify the spectral composition of the investigated music signal. In fact, it is thus reasonable to think that by applying some processing modifications on both the tracks instead of just one, it will be possible to achieve better results in term of satisfactoriness of the resulted mix. Thus, in order to reach a significant improvement in terms of consonance of the final mix, in our approach the idea is to attempt to modify the track that was fixed during the pitch-alignment.

2.4.2 Implemented solution

In order to produce a measurement of compatibility we design our system by means of a compatibility model based on psychoacoustic consonance. First and foremost, as highlighted in this literature review, this is because this type of approach can be applied in a way that solves the main problems of the methods based on key detection and chromagram computation. Indeed, there is the new possibility to analyse music not only in major or minor scales, but also in other modes or even completely atonal music, as none of the psychoacoustic approaches take musical key into account. Specifically, we based our measurement of compatibility on the sensory consonance model developed by Gebhardt in [5]. The main motivation of our choice is based on the fact that, by analysing short spectral frames and comparing them in proceeding order with each other, the object of investigation becomes the vertical dimension of music, which defines the perception of sound at one point in time. Furthermore, for the purpose of estimating the optimal pitch-shift that maximizes the consonance between two tracks, in this model, the pitch alignment is no longer restricted to semitone-categories, but can be refined as needed for the application.

In the original work the sensory consonance model was solely used to compute a numerical value intended to describe the consonance similarity over the whole time length of two pieces of music. In our approach we expand the investigation of the roughness measurement by looking at the roughness that is generated from the interaction of the spectral components of two beat-synchronous time frames. Furthermore, the compatibility model is not only used to compute the consonance maximizing pitch-shift of the final mix. Indeed, we use it to compute specific suppression operations, which will be applied on the spectra describing one of the two tracks. The main idea behind our investigation is that if the roughness measurement provides us with a value describing the psychoacoustic consonance of two different sounds, at the same time is giving us information on the opposite perception of dissonance. The investigation is based on the idea that, instead of searching for the minimum value of roughness, which corresponds to a measure of consonance, we search for the maximum values, which, as the other way around, represent dissonance.

It should be noted however that a single value of consonance or dissonance between two tracks does not give a meaningful statement about if they are harmonically perfectly aligned, as every piece of music consists of different tones sounding simultaneously, which implies the presence of dissonance. After all, a certain amount of dissonance is indispensable in music, as a completely consonant piece otherwise could only consist of pure sinusoids, which definitely would not lead to a very exciting listening experience. However, two different pieces of music generally hold fairly complex spectral characteristics themselves, thus to mix them and preserve a pleasant sounding result, the mixing process should align their pitches in a way that leads to the lowest level of dissonance.

To this end, we introduce a new dissonance suppression method, which as a novelty with respect to existing models of harmonic mixing, aims at reduce the dissonance of the final mix by applying specific modifications on the music at the spectral level. The goal of our suppression method is to identify in one of the input songs, the regions of the spectra that produce a significant dissonant effects when combined with the pitch-shifted version of the other track. This is based on the hypothesis that, by suppressing these spectral regions, the dissonance in the final mix will be reduced, and this as a consequence, will lead to a better mix between the two excerpts in terms of pleasantness of the listening.

The system of this work therefore consists in a model for sensory consonance estimation that suits the scope of harmonic mixing for DJs. By means of that, the algorithm calculates the consonance maximizing pitch-shift between two tracks, then computes and applies a partials suppression method that aims to minimize the dissonance of the final mix.

2.4.3 Architecture of the system

In this section we explain the general scheme that stands behind the architecture of this project. Figure 2.12 describe in a block diagram the main process and the implemented architecture, in Chapter 4 we will explain in specific details all the stages one by one. Starting from some necessary pre-processing operations, the system takes as input two tracks, T1 and T2, and provides as outputs a pitch-shifted version of T2 and a re-synthetized and modified version of T1.

The system general scheme works as follows:

- First, in order to extract a set of partials (i.e. frequencies, magnitudes and phases) that describes the harmonic contents of the tracks, it runs the two input samples into a sinusoidal model. Furthermore, from the original signal and the extracted partials of T1, it subtracts the residual part, which will be used in the end of the process to re-synthetize T1 after the spectra modification on the harmonic content.
- The second step takes the frequencies and magnitudes values of the two tracks and averages those over sixteenth note temporal frames. With these values, the harmonic compatibility model computes the roughness measurement describing the

consonance relationship between the two tracks. By means of that, the system calculates the optimal pitch shift-ratio that will be applied on the original signal of T2.

- The next phase represents the core of our system and consists in the spectral modification process. To this end, we present two different strategies for the dissonance suppression.

    - Our first approach consists in the exploration of the overall roughness value computed within each singular time frame. The idea is to select the time frames that produce a consistent dissonance effect in the final mix and cut off them by applying a masking operation. To this end, the algorithm, by looking at the roughness matrix generated from the optimal pitch-shift computation, is able to select the time frames that maximize the measurement of roughness with respect to its evolution over time. The time frames are selected by setting a threshold value extracted from a percentile computation over the singles measurements of roughness. The suppression method then silences the selected time frames by decreasing the amplitudes of all the partials describing those.

    - In the second approach, the algorithm mathematically applies the optimal pitch-shift on the averaged frequencies values that describe T2. The pitch-shifted frequency's values of T2 and the averaged ones from T1 are then used to explore, within each time frame, the contribution of each partial on the roughness computation. This means that the algorithm is now able to evaluate how, with respect to the roughness model, in beat-corresponding time frames, the partials of two different music samples relate to each other in terms of sensory roughness when played together. The suppression method selects from the partials describing T1 those that generate dissonance when combined with the pitch-shifted partials of T2. The process looks at the roughness measurement generated for each pair of partials and selects those of T1 that overcome a certain value, extracted from a percentile computation on the overall measurements. Those partials will be processed in order to reconstruct a masked version of T1 that is more compatible with the pitch-shifted version of T2.

    - The last operation is an improvement of our partials suppression method, which consists in the harmonic exploration of the frequency spectra describing T1. The algorithm, for each of the most dissonant selected partials within the same time frame, seeks the frequencies spectra looking for some related harmonics. Starting from the frequency value corresponding to the selected partial, it explores the spectra by looking at those regions that correspond to its ideal harmonics. The algorithm, seeks through a chosen number of octaves, by looking at an interval of two semitones, centred on the harmonics value mathematically computed for the investigated partial. If it finds a partial that relies on the observation interval, it selects that partial as one to be suppressed. Once the selection process is completed, the system applies a suppression operation on the original not-averaged partials of T1. Specifically, the algorithm decreases the magnitude of the selected partials by a prescribed amount.

- Finally, the system produces a pitch-shifted version on T2, with respect to the optimal pitch shift ratio computed from the roughness model. For this pitch shifting operation we use the open source pitch-shifting and time-stretching utility "Rubberband"[4].

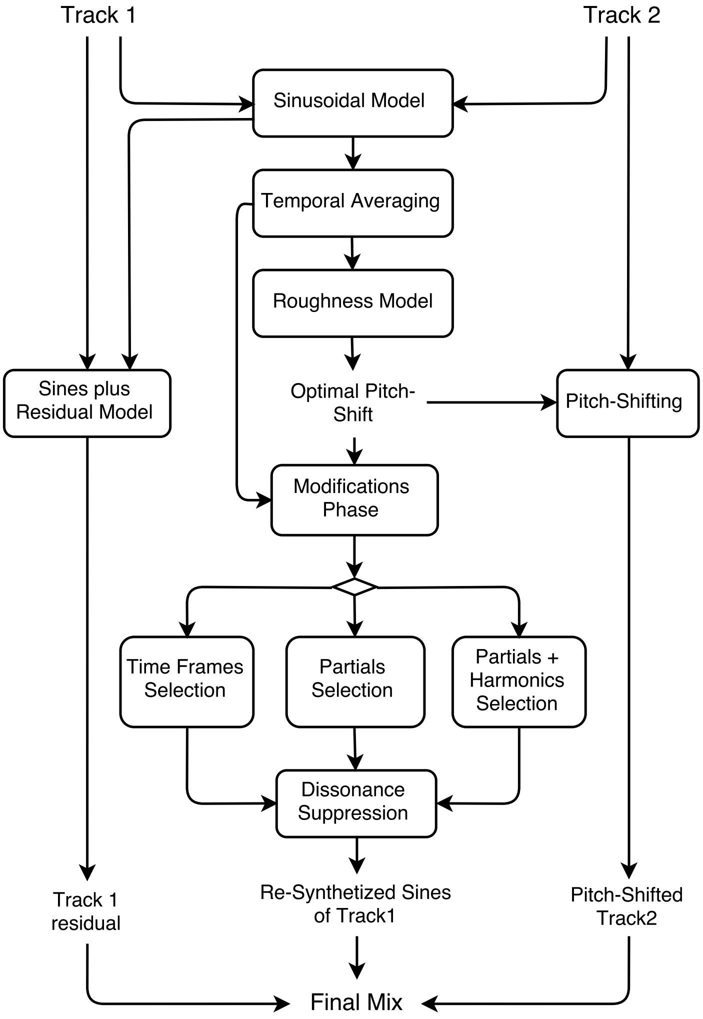
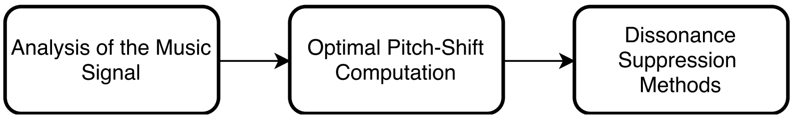


Figure 2.12: An overview of our approach to the automatic harmonic music mixing. Block diagram describing the main architecture of the system.

[4] *http://rubberbandaudio.com/*

Chapter 3
Theoretical background

In this chapter we present the theoretical concepts and the processing tools needed to understand the architecture of the system developed with this project. We present the theory that stands behind the implementation of our consonance based mixing system, and outline the basic tools defining the structure of our approach. The designed architecture is divided in three fundamental and related steps: the analysis of the music signal, the optimal pitch-shift computation, and the modification phase.


Analysis of the Music Signal
Optimal Pitch-Shift Computation
Dissonance Suppression Methods

First of all, we focus our attention on how digital signal processing (DSP) techniques are used to analyse the audio signal and to extract relevant information in terms of musical content. In particular, we will focus on the Fourier analysis of music signals and how this tool provides us with a powerful representation of the harmonic content through the use of the sinusoidal model. Afterwards, we present the musical concepts that have led us to the design of our modification process and relative selection criteria. In particular, we describe the notions of music consonance, presented in a psychoacoustic way and explore the notions of pitch and musical intervals.

Furthermore, we discuss the concept of harmonic partials of a sound and investigate on the relationship that occurs among the harmonics series and the sinusoidal model. Finally, in order to define the compatibility model of our system and how it computes the consonance maximizing pitch-shift ratio between two tracks, we provide an overview of the existing psychoacoustic models of consonance. Specifically, with an accurate investigation over the structure of the model presented in the consonance based mixing approach, we show how the measurement of sensory roughness can provide us with new information on how different pieces of music relate to each other in terms of psychoacoustic consonance.

3.1 Signal Processing Tools

In the audio processing field, one of the most classic methods to describe and analyse the musical content is the *sinusoidal modelling*. Utilizing this model, the sounds produced by musical instruments and other physical systems can be modelled as a sum of sinusoids. In this section we explain how, by means of Fourier analysis, these can be extracted from the audio waveform and represented by their frequency and intensity in the spectrum. Specifically, we focus on the Short-Time Fourier Transform (STFT), which provides us with a time versus frequency representation of the musical content by tracking down sinusoids from the windowed musical signal. Moreover, we investigate on how the spectrum of the signal is modified by the windowing operation and how this process affects the frequency resolution of the partials extraction.

3.1.1 Sinusoidal modelling

With the sinusoidal model the sounds produced by musical instruments and other physical systems can be described as the sum of a set of time varying sinusoids [41]. This means that the audio signal is decomposed into a finite sum of sinusoidal components, usually referred as *partials*, and each of them is composed by a set of parameters, which include amplitude, frequency and phase. Consequently, the extraction of harmonic content from music consists in estimating these parameters for each sinusoid present in the signal. From this point of view, this model offers a parametric and time-varying representation of music.

Mathematically, with respect to the standard sinusoidal model, the harmonic information of the signal $y[t]$ is represented as a sum of R sinusoidal trajectories, each one described by $A_r[t]$ and $f_r[t]$, respectively the instantaneous amplitude and frequency values characterizing the partial at the time instant t:

$$y[t] = \sum_{r=1}^{R} A_r[t] \cos\,(2\pi f_r[t] t). \tag{3.1}$$

With sinusoidal modelling the sinusoids estimation procedure is performed by calculating the short-time spectrum on small intervals of the signal, which we will see in detail in the next section. Once computed, the spectrum is then analysed, prominent spectral peaks are detected and their parameters, amplitudes, frequencies, and phases, are estimated (Fig. 3.1). The main core of the partial extraction process therefore consists in the identification and tracking of these spectral peaks from the audio signal.

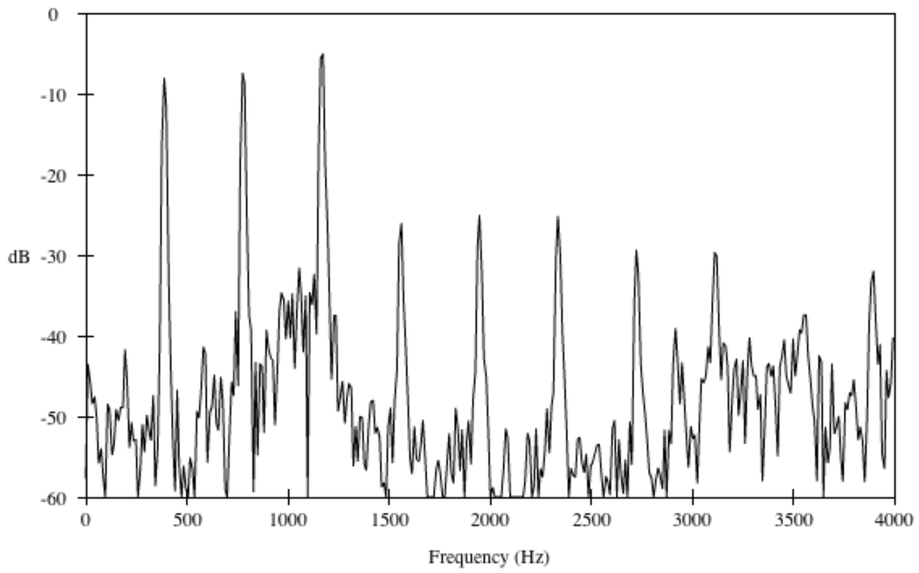


Figure 3.1: Spectral peaks, evoked by a sung vowel with a fundamental frequency of 392 Hz. Image from https://gerrybeauregard.wordpress.com/

Sinusoidal modelling stands as a powerful tool to describe music, indeed rather than representing a sound by its entire time or frequency representation, it allows us to accurately represent it by a model containing only the frequencies, amplitudes and phases of each extracted partials. This model is used to process audio in most of the applications that need structured knowledge of these basic sound properties, because it allows us to fully define a single sound. Sinusoidal modelling has been successfully used in audio applications like matching algorithms [42], source separation [43], voice effects and resynthesis [44] In fact, once the sinusoidal model of a sound has been constructed, the parameters can be changed at will, and the manipulated sound can be resynthesized from the model using an inverse Fourier transform. For this reason, the sinusoidal model represents a powerful descriptor of audio signals when mixing digital music, because it allows us to analyse the harmonic content, possibly modify it, and then reconstruct it by using well-defined signal processing tools.

3.1.2 Short Time Fourier Transform

The sinusoidal modelling analysis and the partials extraction process is computed through an estimation procedure by way of the Fourier Transform. To this end, we need to recall that the original signal and its Fourier transform contain the same information. However, this information is represented in different ways. While the audio signal displays the information across time, the Fourier transform displays the information across frequency. In other words, the Fourier transform yields frequency information across the entire time duration of the signal. However, the information on *when* these frequencies occur is hidden in the transform. As put by Hubbard [45], the original signal tells us when certain notes are played in time, but hides the information about which frequencies. In contrast, the Fourier transform of music displays which notes are played, but hides the information about when the notes are played.

Due to the time-varying nature of the musical signals, which are the object of investigation of this project, we need a time-varying representation of the sinusoidal content over the entire time progression of the musical structure. This computation is approached using the Short-Time Fourier Transform, which is calculated applying the Discrete Fourier Transform (DFT) to small regions of the audio signal. The DFT is the relative of the Fourier transform applied to discrete-time signals, such as music signals are; as a consequence, it is mathematically simpler and computationally faster. Nevertheless, due to the highly non-stationary nature of the musical signal, the main drawback of the DFT is that the temporal information is lost. For this reason, it is necessary to compute a local analysis of the frequency components instead of calculating it over the whole signal. To this end, for the computation of the STFT, a so-called *window function* has to be fixed; which is a function that is non-zero for only a short finite period of time (defining the considered interval). The original signal is then multiplied with the window function to yield a *windowed signal.* To obtain the frequency information at different time instances, the window function has to be shifted across time and then faster implementation of the DFT, called Fast Fourier Transform (FFT), can be applied for each of the resulting windowed signal.

The series of FFTs, which form the implementation of the STFT, are computed by means of a windowing process. The window is a simple function, with M samples (i.e., the interval where the function is not zero-valued), which slides over the signal with a hop size H_{hop}. The hop size parameter is specified in samples and determines the step size in which the window is to be shifted across the signal. The parameter M represent the length of the window function and determines the duration of the considered sections, which amounts to M/F_s. The result is the splitting of the signal into a set of frames of length M, which can be made overlapping if $H_{hop} < M$. When the frames overlap, the obtained representation provides more information on the temporal dynamics of the signal. Mathematically we can express the STFT of a signal x as:

$$X_r(\omega_k) = \sum_{n=0}^{M-1} x(n - rH_{hop})\omega(n)e^{-j\omega_k n}, \qquad k = -\frac{M}{2}, \dots, \frac{M}{2}; \tag{3.2}$$

where $x(n - rH_{hop})\omega(n)$ is the n-th sample of the r-th frame; $\omega_k = 2\pi \cdot F_s \frac{k}{M}$ is the k-th frequency bin; $X_r(\omega_k)$ is the r-th frame content at k-th frequency bin and k, assuming N_{stft} is even, is the frequency index corresponding to the Nyquist frequency. Note that for each fixed time frame r, we obtain a spectral vector of size

$k + 1$ given by the coefficients $X_r(\omega_k)$. The computation of each such spectral vector amounts to a DFT of size M, which can be done efficiently using the FFT.

From this equation it is easy to understand that the STFT computes a spectrum for each frame, and that this leads to a time versus frequency representation of the music signal. Since the spectrum of real signals present Hermitian symmetry [46], it is possible to discard the components related to the negative frequencies without losing information. Moreover, since the phase is not as informative as the magnitude [47], generally only the latter is considered. The main drawback of using this method is that there is a trade-off between temporal and frequency resolution. This is due to the fact that the multiplication in the time domain of the signal $x(n)$ with the window $\omega(n)$ corresponds to the convolution between their spectra $X_{stft}(\omega_k, r)$ and $W(\omega_k, r)$ in the frequency domain.

It is important to notice that the windowing operations on the signal produce a significant change in the way we look at the sinusoidal content. Furthermore, the shape and the length of the type of the window affect in a consistent way the frequency resolution of the partials extractions. Hence, finding the right configuration for the windowing process is an important task, which allows us to define how we want to represent the musical content from the signal, and specifically how much precision we want in the resolution of the partials. To this end, in the next section we present a brief overview over the concept of windowing, and we introduce the main differences between the most commonly used types of windows.

Before looking in detail at the windowing operation, we first introduce the concept of *spectrogram,* which is a two dimensional representation of the squared magnitudes of the STFT. The spectrogram of a signal, as showed in Fig. 3.2, can be visualized by means of a two dimensional image, where the horizontal axis represents time and vertical axis represents frequency.

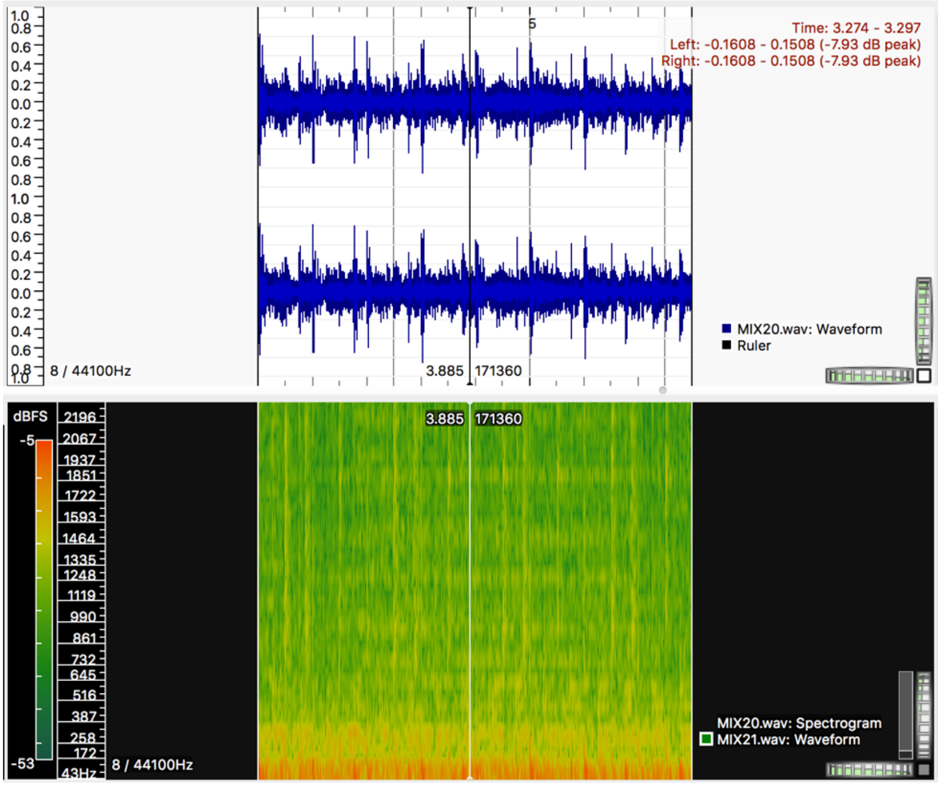


Figure 3.2: Waveform representation and Spectrogram of an audio sample extracted via Sonic Visualizer[5]

3.1.3 Windowing

We now discuss how effects of the window function over the music signal, which plays an important role from an audio processing point of view. The design of suitable window functions and their influence is a science by itself, which is outside the scope of this project. However, there are several well-known types of window function that serves various purposes and exhibit various properties. Windows impact many attributes of harmonic analysis; these include detectability, resolution, dynamic range, confidence and ease of implementation. We would like to identify the major parameters that will allow performance comparisons between different windows. To understand how a given window affects the frequency spectrum, we have to explore more in depth the frequency response characteristics of windows.

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

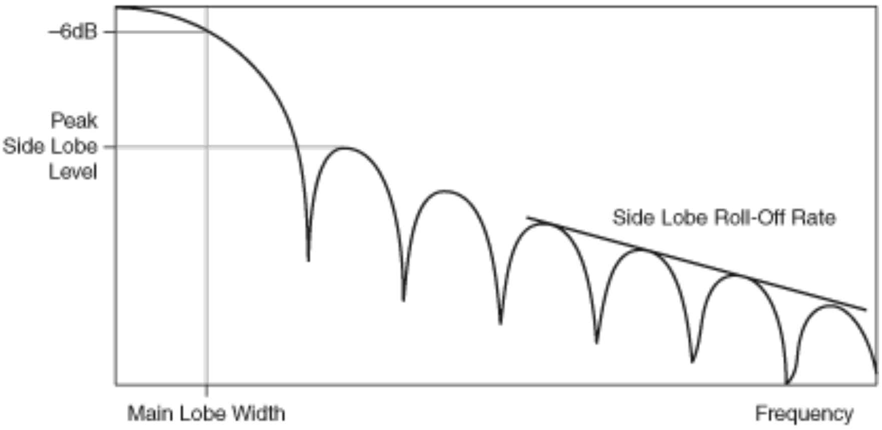


Figure 3.3: Frequency response of a window. Image from [http://www.ni.com].

The most complete reference about windowing is the paper by Harris [48], as stated in this study, an actual plot of a window shows that the frequency characteristic of a window is a continuous spectrum with a main lobe and several side lobes (Fig. 3.3). The main lobe is centred at each frequency component of the time-domain signal, the side lobes approach zero and the height of the side lobes indicates the affect the windowing function has on frequencies around main lobes.

The choice of the type of window is mainly determined by two of the window spectrum's characteristics: the width of the main lobe, defined as the number of bins (DFT-sample points or bins) between the two zero crossings, and the highest side-lobe level, which measures how many dBs down is the highest side-lobe from the main lobe. Ideally we would like a narrow main lobe, which results in a good resolution of the partials extraction, and a very low side-lobe level that guarantees no cross-talk between FFT channels. This is not possible in reality because spectral analysis by windowing involves a trade-off between resolving comparable amplitude components with similar frequencies and resolving disparate amplitude components with dissimilar frequencies.

Specifically, the width of the main lobe of the window spectrum limits the frequency resolution of the FFT on the windowed signal. Therefore, the ability to distinguish two closely spaced frequency components increases as the main lobe of the smoothing window narrows. As the main lobe narrows and spectral resolution improves, the window energy spreads into its side lobes decreasing amplitude accuracy, this is the reason why a trade-off occurs between amplitude accuracy and spectral resolution. Side lobes occur on each side of the main lobe and approach zero at multiples of f_s/N from the main lobe (where F_S is the sampling frequency).

The mathematical function which express the relation between the size of the main lobe of a window and its spectral resolution, can be summarized as:

$$M \geq L \frac{F_s}{f_2 - f_1}, \tag{3.3}$$

where M is the size of the window in samples, F_s the sampling frequency and $f_2 - f_1$ is the frequency resolution interval; L is the coefficient reflecting the bandwidth which depends on the type of window selected and corresponds to the width of the main lobe expressed in number of bins (examples: rect=2, hann=3, blackman=6).

Consequently, choosing the optimal window for a specific application requires knowledge of the signals involved, consideration of the frequency resolution, and the dynamic range requirements. Different window functions are available with different frequency response characteristics. The following figure lists the characteristics of several smoothing windows.

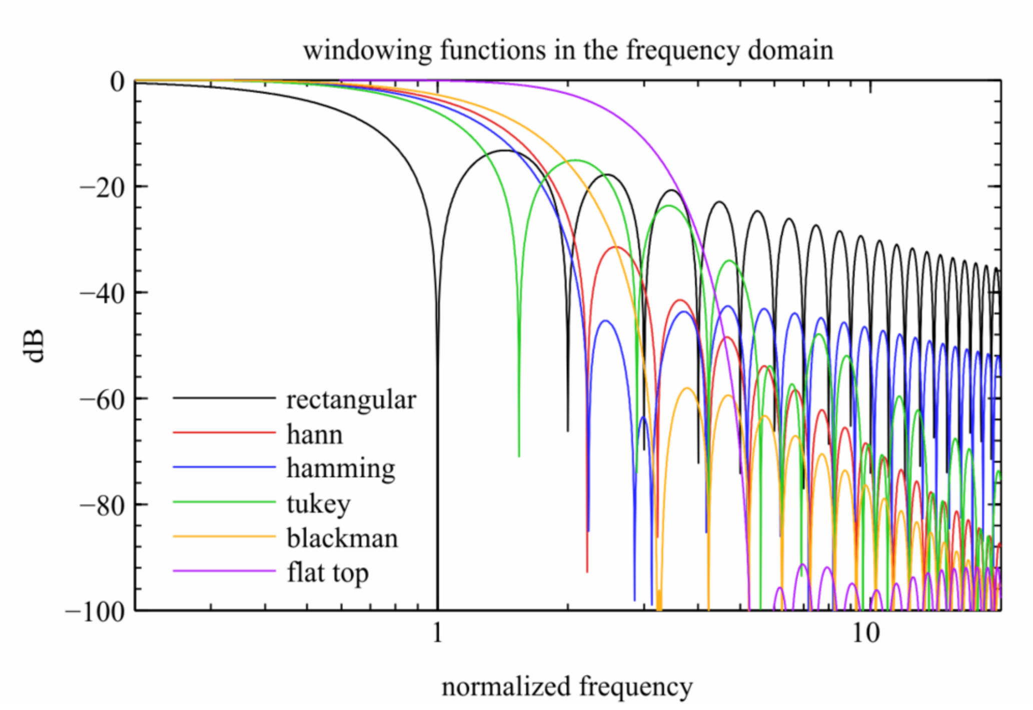


Figure 3.4: Window functions in the frequency domain ("spectral leakage"). The frequency axis has units of FFT "bins" when the window of length N is applied to data and a transform of length N is computed. It is relative to the maximum possible response, which occurs when the signal frequency is an integer number of bins. [Image from Wikipedia]

We now provide a quick overview over the most used types of window, trying to underline the relevant pros and cons for each of it. Even by using no window, by the nature of taking a snapshot in time of the input signal and working with a discrete representation, the signal is convolved with window of uniform height. This convolution has a sine function characteristic spectrum, for this reason, no window is often called the uniform or rectangular window. The rectangular window has excellent resolution characteristics for sinusoids of comparable strength, but it is a poor choice for sinusoids of disparate amplitudes. This characteristic is sometimes described as *low-dynamic-range*. This type of window has the best noise bandwidth, which makes it a good candidate for detecting low-level sinusoids in an otherwise white noise environment. Interpolation techniques, such as zero-padding and frequency-shifting, are available to mitigate its potential scalloping loss. Besides the rectangular window, for the purpose of investigation on the harmonic content of the signal, we need to describe and understand the characteristics of more complex window function.

The Hamming and Hann window functions both have a sinusoidal shape and result in a wide peak but low side lobes. However, the Hann window touches zero at both ends eliminating all discontinuity (Fig. 3.5). This window is useful for analysing transients longer than the time duration of the window and for general-purpose applications. The Hamming window doesn't quite reach zero as the Hann one and thus still has a slight discontinuity in the signal (Fig. 3.5). Because of this difference, the Hamming window does a better job of cancelling the nearest side lobe but a poorer job of cancelling any others. This window function is useful for noise measurements where is wanted better frequency resolution, than some of the other windows, but moderate side lobes do not present a problem. The Blackman-Harris window (Fig. 3.6) is similar to Hamming and Hann windows; it is derived by considering more general summation of shifted sine functions. This type of window is useful for single tone measurement; in fact, the resulting spectrum has a wide peak, but good side lobe compression.

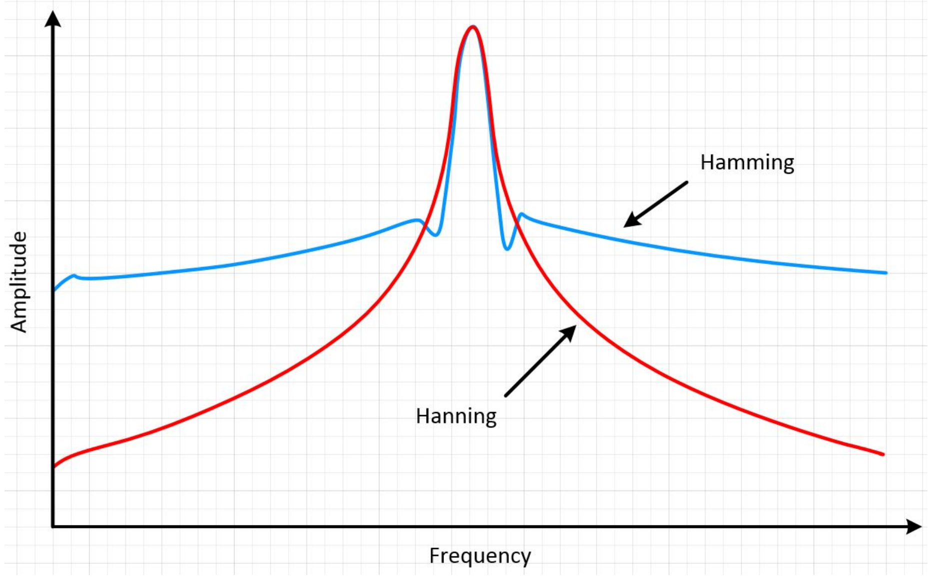


Figure 3.5: Hamming and Hann windowing result in a wide peak but nice low side lobes. [Image from http://www.ni.com].

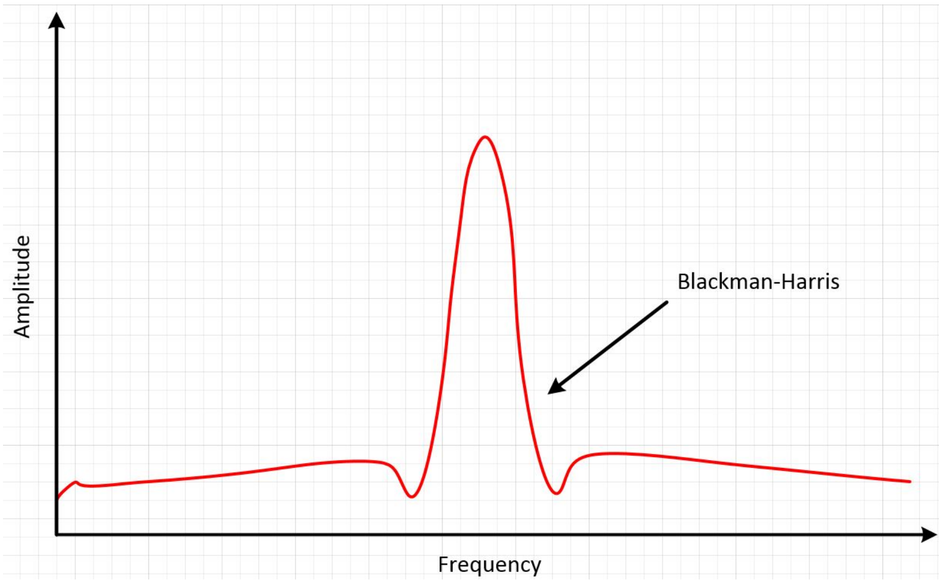


Figure 3.7: The Blackman-Harris results in a wide peak, but good side lobe compression. [Image from http://www.ni.com].

With this review over the windowing process we have seen how the choice of the type of window consistently affect the sinusoidal model which represent the musical content of the signal under investigation. In chapter 4 we will describe and justify the choices that have brought us to the configuration implemented for our system.

3.1.4 Inverse FFT re-synthesis

Windowing operations are not solely used in the harmonic analysis, indeed, a window function also play a fundamental role in the re-synthesis of an audio signal performed by means of the Inverse Fourier Transform. Indeed, since the FFT is invertible, it is possible to synthesize any desired time domain waveform by taking the complex spectrum of the desired signal and applying the inverse FFT. In this case we are interested in the synthesis of sinusoids which, as detailed in section 3.1.1, have well understood spectra. Time domain windowed sinusoid can be synthesized by the following procedure:

1) shift the complex spectrum of the window function so that it is centered on the bin frequency of the desired sinusoidal frequency

2) scale the spectrum of the window function according to the desired complex amplitude (magnitude and phase)

3) accumulate the scaled shifted window function into the FFT buffer

4) perform the inverse FFT

Consider a sinusoid in the time domain: its STFT is obtained by first multiplying it for a time window and then performing the Fourier transform. Therefore, the transform of the windowed sinusoid is the transform of the window, centered on the frequency of the sinusoid, and multiplied by a complex number whose magnitude and phase are the magnitude and phase of the sine wave. Steps 1–3 thus correspond to convolving the window spectrum with the spectrum of a desired sinusoid. Because convolution is a linear operation, steps 1–3 can be performed for each sinusoid to be synthesized. The inverse FFT can then be computed as a final step. If the window has a sufficiently high side lobe attenuation, the sinusoid can be generated by calculating the samples in the main lobe of the window transform, with the appropriate magnitude, frequency and phase values. Although any window function can be used, for computational efficiency it is preferable a window function with a reasonably narrow main lobe and low side lobes. One can then synthesize as many sinusoids as desired, by adding a corresponding number of main lobes in the Fourier domain and performing an IDFT to obtain the resulting time-domain signal in a frame.

By an overlap-and-add process it is then possible to obtains the time-varying characteristics of the sound. Note however that, in order for the signal reconstruction to be free of artifacts, the overlap-and-add procedure must be carried out using a window with the property that its shifted copies overlap and add to give a constant value. A particularly simple and effective window that satisfies this property is the triangular window. The result of the inverse FFT is then a sum of sinusoids (each with constant amplitude and frequency) that have been multiplied by the time domain window function.

A final remark concerns the FFT size: in general, a high frame rate is desired, so that frequencies and magnitudes need not to be interpolated inside a frame. At the same time, large FFT sizes are needed in order to achieve good frequency resolution and separation of the sinusoidal components. As in every short-time based processes, one has to find a trade-off between time and frequency resolution.

3.2 Musical background

This section will focus on the main concepts related to music theory that are needed to completely understand the nature of our investigation and the selection criteria which form the core of our suppression method. To this end, we first discuss the notion of pitch and its perceptual nature. We will then expound the concept of harmonics and how different sound are considered to be harmonically related by means of the harmonic series and harmonic interval.

3.2.1 Musical pitch

Pitch is the perceptual attribute that allows humans to order sound on a frequency-related scale. In other words, it is the property of sound that makes it possible to perceive sounds as higher and lower in a melodic sense [49]. It is possible to estimate the pitch of a sound just as a frequency value, but pitch is not a purely objective physical property. As a matter of fact, it is a subjective psychoacoustic attribute of sound. Historically, the study of pitch perception has been a central problem in psychoacoustics, and has been the object in forming and testing theories of sound representation, processing, and perception in the auditory system [50].

Along with duration, loudness and timbre, pitch is one of the major auditory attributes of musical sounds, and due to its nature of psychoacoustic variable, the pitch perception depends on subjective sensitivity. In the simplest case of a sound with a single frequency (i.e. a sinusoidal tonal sound), the frequency is its pitch [51]. In the past few years, different theories have been proposed to explain the human pitch perception process [51], but it's still not completely clear how the pitch is coded by human brain and which factors are involved in the perception. Besides that, within most of the musical range, perceived pitch approximates a linear function of the logarithm of frequency. Equivalently, constant ratios of frequencies give rise to constant perceived pitch differences.

3.2.2 Harmonic intervals and Pitch Class

In music, the phenomenon of pitch perception brought to the definition of the concept of notes, which are associated to perceived frequencies. The distance between two notes is called interval. The perceptual dimension of harmony, which is the way humans perceive different sounds to fit "pleasant" together, is related to ratios of frequencies. We can describe a harmonic interval as the distance in terms of frequency values between two different sound that are perceived as agreeable, smooth, and giving a sense of fusion or unity. Despite the variety of adjectives used, intervals are ordered quite consistently. The unison (1:1) and octave (2:1) are the most consonant, followed by the perfect intervals, the perfect fifth (3:2) and the perfect fourth (4:3). The major third (5:4), minor third (6:5), major sixth (5:3), and minor sixth (8:5) are next most consonant. The least consonant are the minor second (16:15), the major seventh (15:8), and the tritone (45:32). These intervals and their ratios are shown in Figure 3.7.

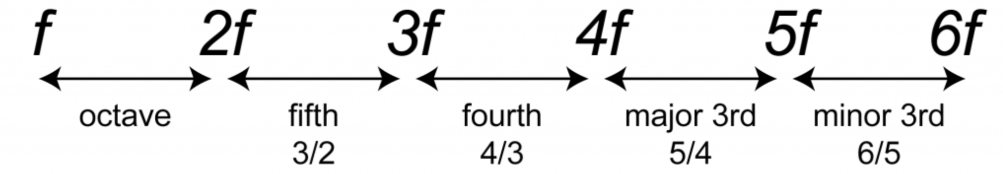


Figure 3.7: Ration of frequency in harmonic intervals. [Image from: digitalsoundandmusic.com/]

An important perceptual aspect that must be taken into consideration is that the human perception of pitch is periodic. This means that, if two notes are played following a unison interval, that corresponds to have the same frequency, the perceived pitch is the same. While if two notes are in octave relation, that corresponds to a doubling of frequency, the perceived pitches have similar quality or chroma. This phenomenon is called octave equivalence and brought to the definition of the concept of pitch class. Humans are able to perceive as equivalent pitches that are in octave relation. Pitch classes are equivalence classes that include all the notes in octave relation (Fig. 3.8). For example, the pitch class C stands for all possible C's in whatever octave position. Every pitch class is enumerated with an integer scale from 1 to 12.

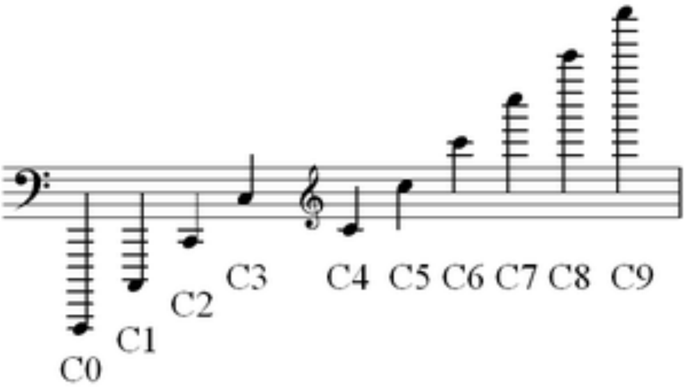


Figure 3.8: Ten Cs in scientific pitch notation [Image from Wikipedia]

3.2.3 Harmonic series

Now that we know how a human being is able to perceive differences between sounds, and how, by means of music harmony theory, different sounds are considered to be more harmonically consonant with respect to the music intervals, we have to focus on the complex nature of the signal produced by a musical instrument. It is important to understand the concept of complex tone, and how the describing frequency spectrum is affected by this complexity.

To this end, we introduce the concept of harmonic series which is an arithmetic series defined as (1×*f*, 2×*f*, 3×*f*, 4×*f*, 5×*f*,...). In terms of frequency (measured in cycles per second, or hertz [Hz]), where *f* is the fundamental frequency (i.e. the lowest frequency of the spectra), the difference between consecutive harmonics is therefore constant and equal to the fundamental. Due to the nature of human ears, which respond to sound nonlinearly, higher harmonics are perceived as "closer together" than lower ones. On the other hand, the octave series is a geometric progression (2×f, 4×f, 8×f, 16×f, ...), and people hear these distances as "the same" in the sense of musical interval. In terms of what one hears, each octave in the harmonic series is divided into increasingly "smaller" and more numerous intervals. The second harmonic, whose frequency is twice of the fundamental, sounds an octave higher; the third harmonic, three times the frequency of the fundamental, sounds a perfect fifth above the second harmonic. The fourth harmonic vibrates at four times the frequency of the fundamental and sounds a perfect fourth above the third harmonic (two octaves above the fundamental). Doubling the harmonic number means doubling the frequency (which sounds an octave higher).

Many instruments produce complex tones containing many individual partials (component simple tones or sinusoidal waves), but the not trained human ear typically does not perceive those partials as separate events. Rather, a musical note is perceived as one sound, the quality of that sound being a result of the relative strengths of the individual partials. Oscillators that produce harmonic partials behave somewhat like 1-dimensional resonators, and are often long and thin, such as a guitar string or a column of air open at both ends. Wind instruments, whose air column is open at only one end, such as trumpets and clarinets, also produce partials resembling harmonics. However, they only produce partials matching the odd harmonics, at least in theory. Some instruments, such as snare drum and wood block, do not produce periodic waveforms and therefore do not give rise to a well-defined sense of pitch.

The simplest case to visualize the harmonic series in a real word musical instrument is a vibrating string (Fig. 3.9). The string has fixed points at each end, and each harmonic node divides it into 1, 2, 3, 4, etc., equal-sized sections resonating at increasingly higher frequencies [52]. Generally, in most pitched musical instruments, the fundamental (first harmonic) is accompanied by other, higher-frequency harmonics. Thus higher-frequency waves occur with varying prominence and give each instrument its characteristic tone quality.

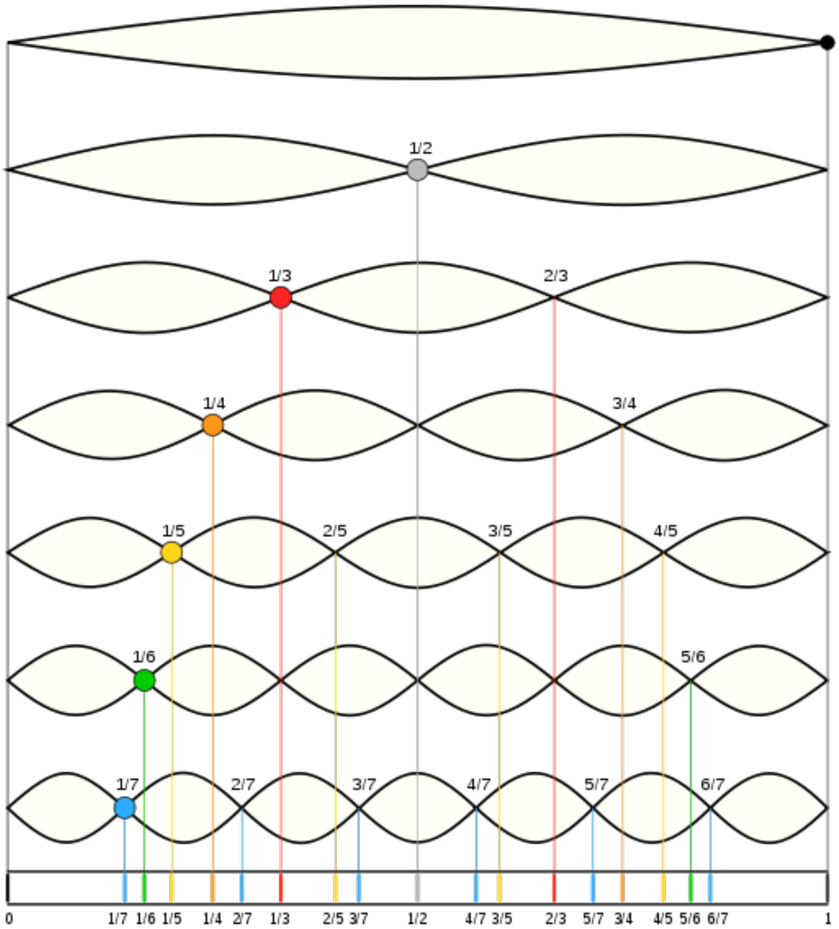


Figure 3.9: Graph representing the nodes of a vibrating string and its relation with the harmonic series. [Image from Wikipedia]

The fact that a string is fixed at each end means that the longest allowed wavelength on the string (which gives the fundamental frequency) is twice the length of the string (one round trip, with a half cycle fitting between the nodes at the two ends). Theoretically, these shorter wavelengths correspond to vibrations at frequencies that are 2, 3, 4, 5, 6, etc., times the fundamental frequency. Physical characteristics of the vibrating medium and/or the resonator which vibrates against it, often alter these frequencies. However, those alterations are small, and except for precise, highly specialized tuning, it is reasonable to think of the frequencies of the harmonic series as integer multiples of the fundamental frequency.

3.3 Psychoacoustic consonance

As stated in the previous chapter, the compatibility model of our mixing system is based on a psychoacoustic approach. In order to understand our implementation and how this model works, we provide a review over the concepts of consonance and dissonance from a psychoacoustic point of view.

It is apparent that consonance, even though it can be described very well by a number of parameters, cannot exactly be defined by one simple formula, as there is a large number of different factors that exert an influence on it. As we saw in the previous section, the musicological definition of consonance is based on the interaction of musical intervals, of which some are more and others less consonant. A short survey of the history of which intervals were seen as consonant or dissonant though, already indicates the difficulty of the definition. While the unison and octave were always seen as consonances, the thirds, fourths, fifths and sixths were alternately considered as perfect or imperfect consonances or even understood as dissonant [26]. Throughout the 20th century, what is perceived or defined as consonant or dissonant is strongly connected to cultural background, musical education and in the end simply personal taste. From the psychoacoustic point of view, there are different ways to describe the perception of consonance, under physiological and physical aspects

One of the most prominent models describing psychoacoustic consonance was published in 1863 by von Helmholtz, who described dissonance (as the opposite of consonance) as the result of beatings produced by two or more interfering tones in the often cited *On the Sensations of Tone* [53]. This perceptual event is based on the amplitude modulation of the sonorities' overall waveform and is dependent on the spectral distance of the pure-tone components that form the sonority and is defined as *roughness* [25]. In the simplest example, let us consider two sinusoids. If these sinusoids have the same frequency, they are perceived as one single tone and the value of roughness is at a minimum. If the frequency difference increases, slow amplitude fluctuations appear that grow faster with the interval between the sinusoids and the perception becomes rougher, until the maximum roughness is reached at around one quarter of the critical bandwidth. While in some literature [35], the terms of fluctuation and roughness are handled independently; in the following only the term roughness will be used, as both are results of the same effect. Other factors that influence the sense of consonance and dissonance are sharpness and tonalness. Sharpness is based on a sound's spectral characteristics; it measures the quantity of energy at high frequency values and contributes the perception of dissonance positively. The concept of tonalness is a great deal in literature. To avoid misunderstandings, and as the model that will be used here is mainly based on the research of Ernst Terhardt, we will follow his definition. With this, the clearness of pitch in the sound or, in other words, the tonal proportion of a sound is described. White noise can be seen as example for a signal with a minimal level of tonalness, as its energy is spread over the entire spectrum with equal intensity. In contrast, a single sinusoid without the disturbance of other tonal elements has a maximum tonalness, and can be identified very easily by its pitch. According to Terhardt [35], fluctuation/roughness, sharpness and tonalness together represent one of the two categories of indicators for the amount of consonance, that he calls *sensory consonance.*

3.3.1 Roughness model

By following the Terhardt's psychoacoustic definition, the category of sensory consonance is divided into three parts: roughness, tonalness and sharpness. Due to the nature of sharpness, which is closely connected to timbral aspects, we do not attempt to modify these characteristics of the analysed audio. Parncutt & Strasburger [39] mention the strong relation between roughness and tonalness as the reason for the sufficiency to analyse one of the two properties. The fact that roughness has been more extensively explored than toneless and most of existing sensory consonance models build exclusively on it, motivates the use of roughness as single descriptor for the sensory consonance.

In order to understand the concept of roughness perception we have to introduce the experiment made by Plomp and Levelt in [37]. In their investigation, they asked to 90 musical untrained participants to rate how dyads of pure tomes relate to each other in terms of perceived consonance. The results are shown in Figure 3.10, and underline a relationship between consonance and the frequency interval between the tones, related to the critical bandwidth (i.e. the interval of frequencies within a second tone will interfere with the perception of a first one).

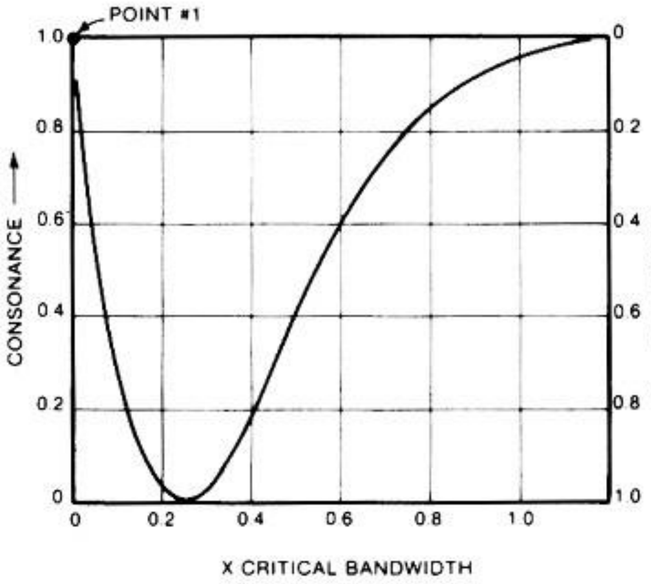


Figure 3.10: Dissonance function, which gives a dissonance value as a function of the frequency ratio of the two tones, expressed in units of the critical bandwidth. Image from [37]

As we can see, two unison frequencies are rated as the most consonant. As the frequency difference between the two sinusoids rises, their combination is perceived as more dissonant, until at about one quarter of the critical bandwidth, their dissonance reaches a maximum value. From this point on, the dissonance falls back towards zero, which is reached at an interval of about 1.2 times the critical bandwidth. What their measures imply is the presence or absence of roughness, corresponding to dissonance and consonance respectively.

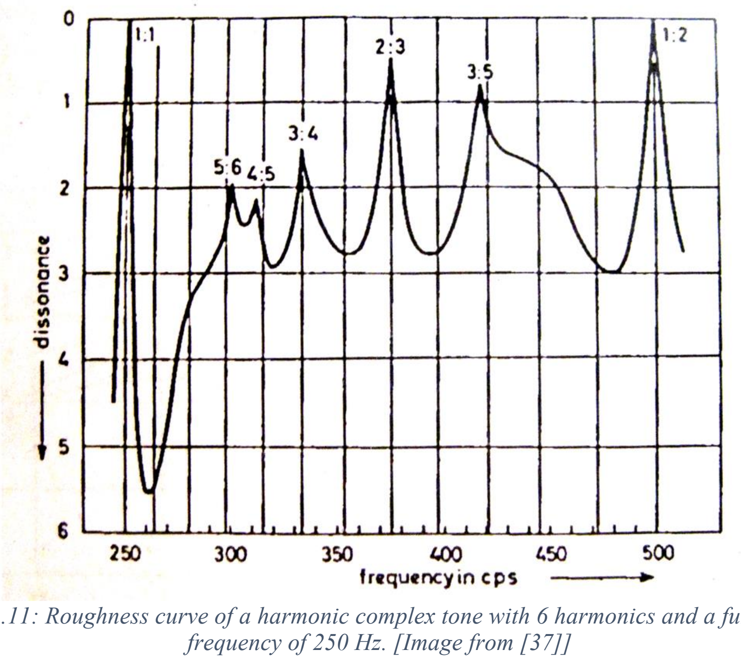


Figure 3.11: Roughness curve of a harmonic complex tone with 6 harmonics and a fundamental frequency of 250 Hz. [Image from [37]]

There is another important aspect to be considered before analysing the computation of a roughness measurement. In fact, pure sinusoids, as used for the experiment made by Plomp and Levelt, represent an artificial signal that barely exists naturally. Musical sounds, as we saw in the previous section, generally consist of a harmonic complex tone, as every partial is a harmonic of the fundamental. Since all partials of two harmonic complex tones sounding at different intervals can also be seen as a collection of pairs of pure tones, by means of the Fourier theorem, Plomp and Levelt propose that adding up all of the roughness values that are produced by each of the dyads can represent the overall roughness of the two harmonic complex tones. The graph that yields from this assumption is described in Figure 3.11, which represents the evolution of the roughness curve at many intervals. As to be expected, the unison, like in the model with just two sinusoids, has lowest roughness. Other minima of roughness appear at frequency ratios that correspond to the fifth (3:2), the fourth (4:3), the major and minor thirds (4:5 & 5:6) and the major and minor sixths (3:5, 5:8). In the shown graph, the major sixth does not evoke a peak, as its ratio of 5:8, as the number of harmonics of the complex tone is restricted to 6. Interestingly, these are exactly the ratios of the justly intoned scale, which means that the location of low roughness values is closely related to standard musical usage of these intervals, even if classical instruments are usually tuned in equal temperament.

3.3.2 Roughness computation

As explained in the chapter related to the State of the Art of this thesis, in the recent article [5], the authors, with the aim to compute the most consonance mix between two tracks, presented a new model for the computation of sensory roughness. To estimate the degree of roughness that is evoked by different harmonic components of different music pieces, in this approach the musical content is analysed through a sinusoidal model, which, as described above, extracts a defined number of partials from the signal. After that, for each of the partials (i.e. sinusoids) of each resulting spectrum, the roughness that is generated by the combination with other partials is computed, then weighted by the corresponding amplitudes and finally summed for every sinusoid over the spectrum.

The basic structure used to estimate the measurement of consonance is a modified version of Hutchinson & Knopoff's roughness model for complex sonorities [36], which is based on the roughness curve for pure tone sonorities (Fig. 22). Parncutt in [38] proposed a function that approximates the graph estimated by Plomp & Levelt

$$g(y) = \begin{cases} \left(\exp(1)\frac{y}{0.25}\exp\left(-\frac{y}{0.25}\right)\right)^2 & y < 1.2 \\ 0 & otherwise \end{cases}, \qquad (3.4)$$

where y the frequency interval between two partials (f_i and f_j) expressed in the critical bandwidth (CBW) of the mean frequency, such that:

$$y = \frac{|f_j - f_i|}{CBW(\bar{f})} \text{ and } \bar{f} = \frac{f_i + f_j}{2}. \qquad (3.5)(3.6)$$

As highlighted before, the Hutchinson & Knopoff's formula for the calculation of the critical bandwidth was often the subject of criticism. To this end Parncutt states that better results can be obtained by using Moore & Glasberg's [54] equation for the equivalent rectangular bandwidth (ERB):

$$ERB(\bar{f}) = 6.23(10^{-3}\bar{f})^2 + 93.39(10^{-3}\bar{f}) + 28.52, \qquad (3.7)$$

and thus CBW is substituted with ERB. The roughness values $g(y)$ for every pair of partials are then weighted by the amplitudes (M_i and M_j) to obtain a value of the overall roughness D:

$$D = \frac{\sum_{i=1}^{N} \sum_{j=i+1}^{N} M_i M_j g_{ij}}{\sum_{i=1}^{N} M_i^2} \qquad (3.8)$$

3.3.3 Consonance based mixing

Since the purpose of the Consonance based mixing method [5] is to identify by means of the roughness model the pitch shift factor that maximize the consonance between two tracks T1 and T2, the frequencies of T2 are scaled to cover the range of one full octave around the original pitch. This shifting operation is made through steps of an eighth semitones, in order to provide 97 different frequency matrices of which each depicts a simulated pitch shift of one track, accordingly 97 frequency-magnitude matrix. The fact that, roughness has been seen as a major factor of dissonance that should be minimalized to achieve a consonant result, a selection of suitable pitch-shifts is made by computing the sum of the roughness values for all time indexes for all combinations of T1, with the pitch-shifted partials components of T2. The result of this calculation is a roughness curve, of which its local minima represent the most consonant pitch-shifts of their frequency region. In this way, a fine-tuning decision is made to obtain a choice of different possible shifted versions of $T2$.

The optimal pitch-shift estimation process works as follow: consider the two input musical tracks, T1 and T2 with corresponding partials and amplitudes $f_{\gamma,i}^{1}$, $M_{\gamma,i}^{1}$ and $f_{\gamma,i}^{2}$, $M_{\gamma,i}^{2}$ respectively, the algorithm now seeks to find the optimal consonance-based alignment between them. The approach focuses on the calculation of consonance as a function of a frequency shift, and is based on the idea that under some frequency shift applied on T2 the consonance between T1 and T2 will be maximized, and this will lead to the optimal mix between the two tracks. In total it creates s = 97 shifts which cover a range of ± 6 semitones in 1/8th semitone steps (i.e., 48 downward and 48 upward shifts). It then scales the frequencies of the partials $f_{2,i}$ as follows:

$$f_{\gamma,i}^{2}[s] = 2^{log_2(f_{\gamma,i}^{2})+\frac{s-48}{96}} \quad s = 0,\dots,S-1 \tag{3.8}$$

For each 1/16th note temporal frame, and per shift S, it then merges the corresponding partials and amplitudes between both tracks (as shown in Figure 3.12) such that:

$$f_{\gamma}[s] = \left[f_{\gamma}^{1}\ f_{\gamma}^{2}[s]\right] \quad \text{and} \quad M_{\gamma}[s] = \left[M_{\gamma}^{1}\ M_{\gamma}^{2}[s]\right] \qquad (3.9)(3.10)$$

It then calculates the roughness $D_{\gamma}[s]$, with the merged partials and amplitudes as input. Then, to calculate the overall roughness $\bar{D}[s]$, as a function of frequency shift s, it averages the roughness values $D_{\gamma}[s]$ across the temporal frames Γ:

$$\bar{D}[s] = \frac{1}{\Gamma}\sum_{\gamma=0}^{\Gamma-1} D_{\gamma}[s] \tag{3.11}$$

Executing this procedure for every of the 97 different combinations of frequency-scaled versions of $T2$ with $T1$, the result is a sequence of 97 sensory dissonance values for different pitch-shifts (Fig. 3.12).

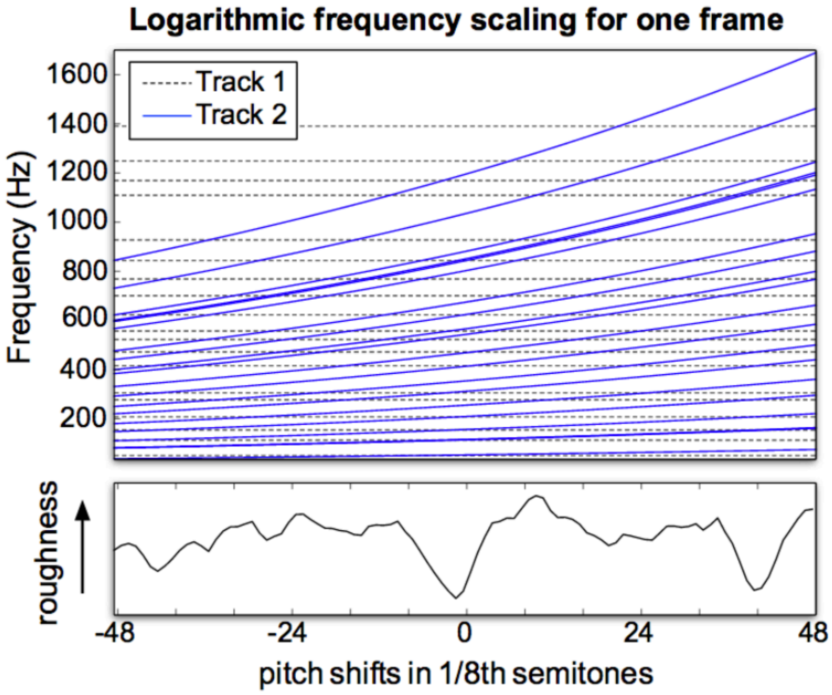


Figure 3.12: (upper plot) Frequency scaling applied to the partials of one track (solid lines) compared to the fixed partials of the other (dotted lines) for a single temporal frame. (lower plot) The corresponding roughness curve as function of a frequency scaling over that frame. [Image from [5]]

Chapter 4
System Implementation

In this chapter we provide a detailed overview of our approach to the harmonic DJ's mixing of music. Starting from some necessary pre-processing operation, we describe each phase of our implementation step-by-step. We also explain the choices we made in setting particular parameters defined of our implementation.

4.1 Data collection and Pre-processing

For the purpose of this project, which represents our first investigation about automatic music mixing, and specifically for the intention to investigate the harmonic compatibility between two tracks, it is useful to consider several simplifications and pre-processing conditions concerning the properties of the musical audio signal intended to mix. Our motivation is to compare this approach with those of existing systems that suit the scope of harmonic mixing for DJs. For this reason, it will be currently considered only electronic music. This genre of music is in most of the cases both harmonically stable and it typically has a fixed tempo. From a selection of recent electronic music, in order to obtain a time-synchronous comparison, we manually annotated the tempo and beat locations and then extracted a set of short musical samples, each of the duration of precisely 16 beats (i.e. 4 complete bars). In order to focus entirely on the issue of harmonic compatibility and spectral analysis without considering the temporal alignment, tempo is forced in each of the input tracks to be exactly 120 beats per minute. For the quantization process and for extracting the set of experts from the original tracks, we used the software Ableton Live[6], which allows us to implement any necessary tempo changes through time-

[6] *https://www.ableton.com/*

stretching operations. Consequently, the test set of musical samples consists of a collection of 8s mono WAV files sampled at 44.1 kHz with 16-bit resolution. Specifically, for the creation of the dataset used in the evaluation experiment, 20 pieces of contemporary electronic music have been chosen and then processed to satisfy the needs of the test. A list of the tracks can be found in the Appendix.

4.2 Sinusoidal analysis

Once we have defined the set of input samples which will provide our system, a first step in better understanding the given music signal and its complexity, is to decompose it into a sum of sinusoids. As we saw in chapter 3 of this thesis, this operation results in a representation of the harmonic content that is more accessible for the subsequent processing step.

To this end, the algorithm gets access to the harmonic contents of the input signal and extracts a starting set of partials, described by frequency, amplitude and phase. In our implementation, we used the Xavier Serra's software for the STFT and the Sinusoidal Model from his software package "Spectral Modelling Synthesis Tools"[7], namely the SMS. This library consists of a set of sound analysis/synthesis tools for music applications implemented in python, that we used to extract the sinusoidal content of the tracks. This powerful mathematical tool is one of the basic principles of audio signal processing. Changing the configuration of the set of parameters which characterize it, results in significant changes in the way we look at the audio signal. As a consequence, finding the right configuration, which satisfies the needs of this project, is a critical task.

4.2.1 System configuration

The specific function that we choose to apply from the SMS tools package consist of a partials extraction process. The default implementation allows us to choose the configuration of the sinusoidal analysis by choosing the values for a set of input parameters. Specifically: the type of window, the size of the STFT (bins), the window size in samples and the hop size of the STFT. In order to obtain a good representation of music and due to the fact that we do not actually know the spectral nature of the

[7] *Available as free Python source code at https://github.com/MTG/sms-tools*

signals, we choose to use the general Blackman window. This very popular type of window resulted satisfactory in most of the cases when analysed through many tests. For the purpose of this investigation, which does not need an accurate spectral resolution, but a good representation of the harmonic contents of the signal, using a sampling rate of the audio of 44100 Hz, we choose to use and to extract sinusoids with the default window size and hop sizes of 4096 and 256 samples respectively. Furthermore, we couple the number N of STFT bins and the number M of samples that determines the window length by setting M = N. Finally, since harmonic information above 5000 Hz is often unclear, the sinusoidal model was limited to a maximum frequency of 5000 Hz for the partials to be extracted.

For the remaining parameters of the sinusoidal model we adopt the default values: the minimum amplitude of -60 dB for a spectral peak to be detected, the minimum duration of 0.01 s, and the variables for Serra's function for the frequency deviation threshold:

$$f_{dev,r}[n] = f_{devOffset} + devSlope \cdot f_r[n], \tag{4.1}$$

as $f_{devOffset} = 20\ Hz$ and the factor which widens the range of tolerated deviations for higher frequencies, $devSlope = 0.01$.

The output provided by the partials extraction function for a given input signal, consists in three different matrices F, M, P, describing respectively the time varying values of frequency, amplitude and phase. Each of these matrix measure nxN, where n is the chosen number of partials to be extracted and N is the number of STFT time frames.

4.2.2 Number of extracted partials

The sinusoidal analysis function provided in the SMS model allow us to choose also the number of partials/sinusoids that we want to extract from the music signal. We can state that varying the number of extracted partials affects the system in a relevant way. Henceforth, it was a fundamental task to identify the correct amount of partials. In order to gain a deeper insight into the design of our implementation, which as just stated is highly dependent on the extraction of partials using the sinusoidal model, we generate multiple tests under different configurations. In this way we can examine the correlation between the number of partials and the representation of the music signal in a harmonic perspective. Specifically, we focus on how the representation of the music signal is affected by increasing or decreasing the number of partials extracted from it.

In our tests, to explore the parameter variance, we range the number of partials n from 4 up to 100. From listening to a re-synthetized version of the sinusoidal contents of the music signals we can immediately see that the number of sinusoids play a critical role in the reliability of the resynthesized sample. From the results of the tests we can state that using more than 25 sinusoids (per time frame of each track) has an increasingly negative impact and results in generating artefacts in the resynthesized version of the samples. Likewise, using too few sinusoids also appears to have a negative impact, which results in a description of the harmonic content that is unsatisfactory in terms of listening reliability when compared to the original signal.

Furthermore, considering the roughness model, having too few observations of the harmonic structure will very likely fail to capture all of the main partials generating roughness. On the other hand, over populating the roughness model with sinusoids (many of which may result from percussive or noise-like content) will also obscure the interaction of the true harmonic partials in each track. Within the context of our dataset, we found that a range in between 15 to 25 partials was able to provide a sufficient harmonic representation for our automatic mixing application.

4.3 Residual extraction

An important aspect to be considered in order to accomplish the final purpose of this system, which is to find the dissonance-maximizing spectral region and to suppress them, is to identify where on the signal level these transform operations will take place. It is easy to understand that, as we have seen in the previous section, the investigation procedure will have to look only at the harmonic and thus sinusoidal content of the music signal. However, in order to be able to re-construct the track that will be object of spectral modification, the system has to consider also that part of the signal that is not extracted from the sinusoidal analysis (i.e. the non harmonic content of the musical signal or percussive and noise-like content). To do that, the system processes the audio signal trough a model, which allows us to split the music signal between stable sinusoids (partials) and residual (i.e. inharmonic) component. Our approach is based on the "The deterministic plus residual model" (SPR) model as described in the work of Serra and Smith [6]. The SPR model describes the time-varying spectra as a collection of sinusoids controlled through time by piecewise linear amplitude and frequency envelopes (the deterministic part), and a time-varying residual components (the residual part). Therefore, by following this model, the input sound $s(t)$ can be described as:

$$s(t) = \sum_{r=1}^{R} A_r(t) \cos[\theta_r(t)] + e(t), \tag{4.2}$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively, and $e(t)$ is the residual component at time t (in seconds).

The most straightforward approach for the estimation of the residual component is through the *subtraction* of the deterministic component from the original signal (Fig. 4.1). This subtraction can be performed either in the time domain or in the frequency domain. Time domain subtraction has to be done while preserving the phases of the original sound, and instantaneous phase preservation can be computationally very expensive. One the other hand, frequency-domain subtraction does not require phase preservation. However, time-domain subtraction provides much better results, and it is usually favoured despite the higher computational costs. For this reason, in order to obtain a well-defined representation of the residual part of the signal, in our system we choose to adopt the time-domain subtraction.

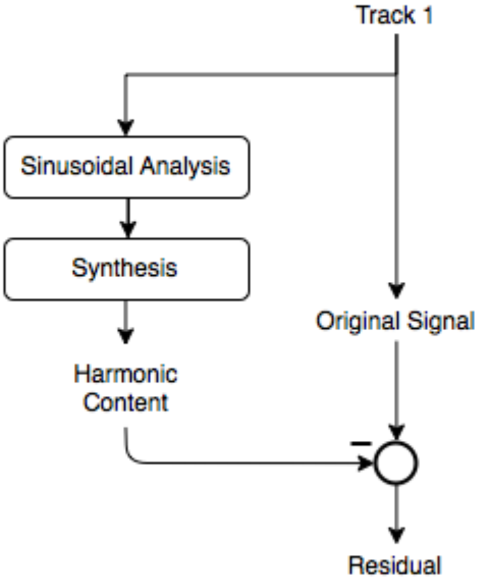


Figure 4.1: Block diagram of the implemented residual extraction.

As explained in the previous section, in our implementation the harmonic component is estimated by running the music signal through a spectral analysis function within the SMS tools. The system then, in order to be able to subtract the harmonic content of the track from the original signal, refers to another function of the SMS library. This specific function allows us to choose the size of the inverse FFT (we adopt the default value of 1024 bins), and the sampling frequency (chosen value of 44100Hz), that will be used in the synthesis process of the extracted partials. This synthesis operation is done by properly interpolating amplitude, frequency, and phase values in order to avoid artefacts in the resynthesized signal. The synthesis function outputs a signal representing the deterministic components of the original sample. This is then subtracted from the original sample, and results in a music signal corresponding to the residual components of the sample.

Consequently, the actual subtraction of the residual can be expressed as:

$$y[n] = s[n] - d[n], \qquad n = 0,1,\dots,N-1; \tag{4.3}$$

where *s[n]* is the original sound signal and *d[n]* are the re-synthesized harmonic sines and deterministic part (an example of the results is shown in Fig. 4.2). Once the subtraction has been performed, there is one more step that we used to improve the analysis of the extracted residual. Simply, several listening tests have been performed on the estimated residual in order to assess how good the analysis was. From this evaluation process we can state that the choice of number of partials also influenced the residual subtraction. In fact, if the spectrum of the residual still contains some partials, then the analysis of the harmonic component has not been performed accurately and the sound signal has to be re-analysed until the residual is free of deterministic components.

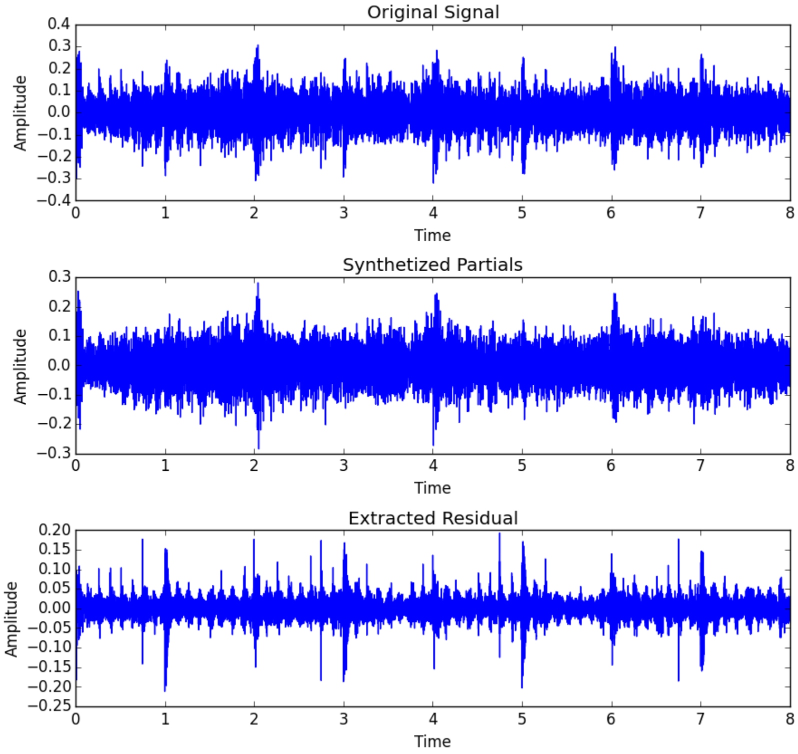


Figure 4.2: Visual representation of the residual extraction results for a training sample. In the first plot is shown the waveform of the original sample. The second and the third represent respectively the re-synthesized harmonic content and residual extracted by means of the previous one from the original signal.

4.4 Temporal averaging

Once the sinusoidal analysis is completed and a set of partials is extracted from the two tracks, the system proceeds with an averaging process of the spectral content over its time evolution. This operation results in decreasing consistently the computation time of the system, but in return it modifies the evolution of the spectra content over time. This means that, by averaging the spectral content over time, a trade-off occurs between computational complexity and temporal resolution. Consequently, defining these operations is as strictly important as choosing the number of sinusoids to be extracted. Hence, in order to understand how the temporal averaging affects the spectral content of the signal and the complexity of the system, we run the algorithm with different intervals. Specifically, we considered three cases: i) 1/8$^{\text{th}}$ note level averaging (32 frames across 4 bar excerpts), ii) 16th note averaging (64 time frames across 4 bar excerpts), and iii) using all the frames from the sinusoidal model without any averaging (1379 STFT bins for our parameterization).

Through our own informal inspection of the effects produced by the different configurations for temporal averaging, we found that, when using the beat averaging configuration (32 time frames) it results with a much noisier relationship if compared to the summarization at the 16th note level (64 time frames). In contrast, the result is smoothest without any temporal averaging, but it is moderately less correlated with the data. With respect to the harmonic dimension, the 16th note segmentation adequately captures the rate at which harmonic content changes in the signal, without losing too much fine detail through the temporal averaging process. Furthermore, for the chosen genre of electronic music of our data set, it's possible to assume that the harmonic structure remains constant over the duration of each 1/16th note (i.e., 125 ms). Therefore, to exact a balance between temporal resolution and computational complexity, we summarize the partials and amplitudes by taking the frame averaged over the duration of each 1/16th note. Thus, for each track we can define a set of frequencies and amplitudes, f_i and M_i, describing the harmonic content, where i indicate each 1/16th note time frame (up to 64).

4.5 Optimal pitch-shift computation

At the end of the analysis process our system is thus able to compute three matrices f, m, p representing the values of frequency, magnitude and phase for each of T1 and T2 input samples (Overview presented in Fig. 4.3). Furthermore, the system is also able to compute the residual content of one of the tracks, and store it for the final re-synthesis operation. Since one of the final purposes is to apply different processing operations for each of the two tracks, in the following we will refer to T2 for the track that will be the subject of the pitch-shifting operations, and to T1 for the signal that will be object of the modification process.

Once the phase analysis is completed, we have now all the components to start our investigation on the harmonic content. We recall that the three matrices f, m, p describing the harmonic content of each tracks were processed through a time averaging process (over 64 time frames). Hence, each of these matrices measures *nx64* where n is the chosen number of extracted partials (Fig. 4.4).

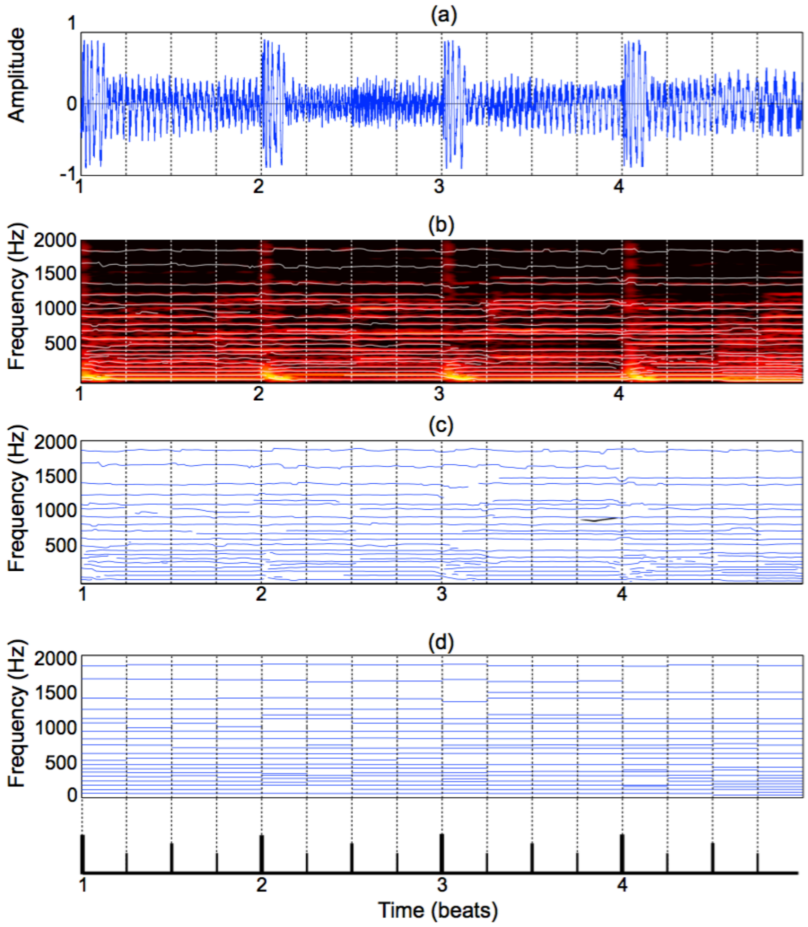


Figure 4.3: Overview of sinusoidal modelling and temporal averaging. (a) A one bar (i.e. 2 s) excerpt of an input audio signal sampled at 44.1kHz at 120 beats per minute. Sixteenth notes are overlaid as vertical dotted lines. (b) The spectrogram (frame size = 4096 samples, hop size = 256 samples, FFT size = 4096) which is the input to the sinusoidal model (with overlaid solid grey lines showing the raw tracks of the sinusoidal model). (c) The raw tracks of the sinusoidal model. (d) The sinusoidal tracks averaged over sixteenth note temporal frames, each of duration 125 ms. Image from [5]

There is another last step to consider before starting with the computation of the optimal pitch-shift ratio. The amplitudes values extracted from the sinusoidal analysis function of the SMS package are expressed in normal dB. In order to facilitate the computation of the roughness model over these set of partials, we want the amplitude values to be expressed in dB-spl. The reason is because with the decibel scale very low volumes correspond to the 0 dB, instead with the db-spl scale the low amplitudes values appear around -140 dB-spl. Consequently, we mathematically convert the amplitude value for the partials of each track in dB-spl. In formula we can express this as:

$$m(t)\left[dB_{spl}\right] = 20\times log_{10}\left(\frac{A_m}{A_{ref}}\right); \qquad (4.4)$$

where *m(t)* is the vector describing the instantaneous amplitude values for the 20 partials of a time frame *t*, $A_{ref} = (20\times10^{-6})$ is the reference amplitude level, and A_m is the linear amplitude level calculated as $A_m = 10^{\left(\frac{m(i)}{20}\right)}$.

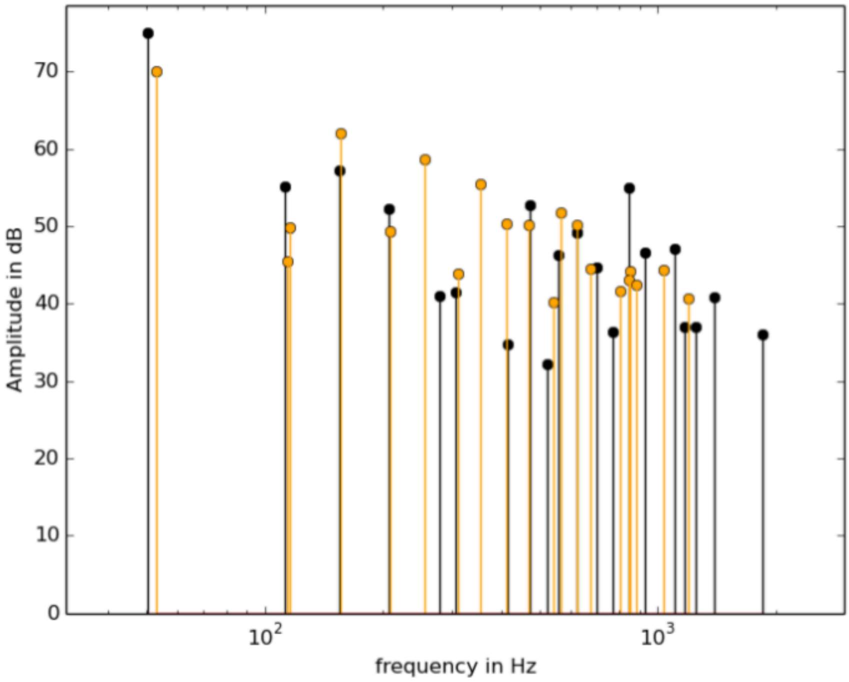


Figure 4.4: The partials of two excerpts for one temporal frame. Each spectrum holds 20 sinusoidal partials and is visualized in its frequency in Hz and amplitude in dB.

Once we have the amplitudes values on a decibel scale we can focus on the computation of the pitch shift ratio that maximizes the consonance between the two input tracks T1 and T2. The computation of the optimal pitch-shift follows directly from the model used in the consonance based mixing method. As a consequence, with respect to the roughness model highlighted in Chapter 3, this model provides us with a measurement of roughness computed over the whole length of the audio samples, which represent the amount of consonance/dissonance generated by the mixing of the two input tracks.

Furthermore, the model computes in the same way a measure of roughness for a set of possible pitch-shifted version of T2. In total it analyses S = 97 possible pitch-shifts which cover the range of ± 6 semitones in 1/8th semitone steps (i.e., 48 downward and 48 upward shifts around a single "No Shift" option). As a result, from the implementation of this model, we obtain as output what we referred to as the roughness matrix, for which a visual representation is given in Figure 4.5.

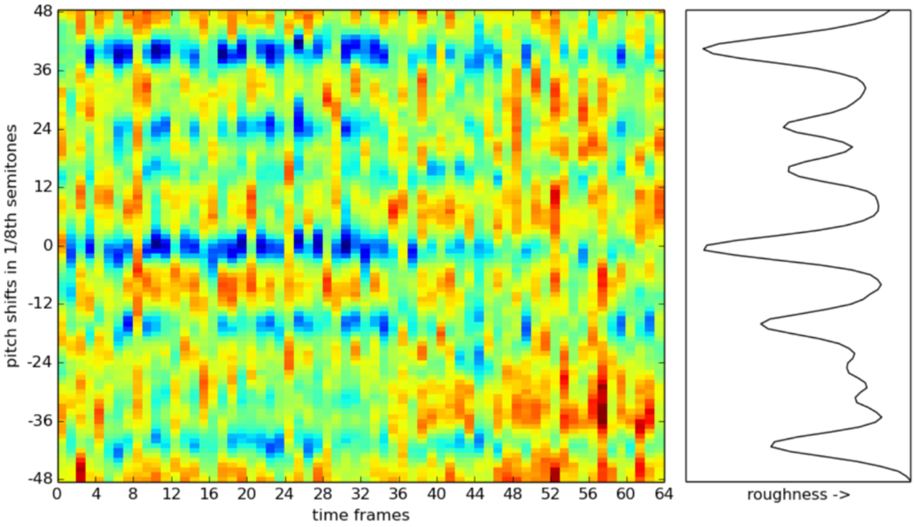


Figure 4.5: Visualization of roughness, D [s], over 64 frames for the full range of pitch-shifts. Blue regions indicate lower roughness, while red indicates higher roughness. The subplot on the right shows the overall roughness value, with respect to corresponding raw of the matrix (i.e. possible pitch-shift).

From the roughness matrix we can select the pitch-shift that results in maximizing the consonance of the final mix. This ratio, in the following, will be referred to the *optimal pitch-shift* and it is obtained by selecting from the set of investigated pitch-shift, the one that results to generate the minimum value of overall roughness.

4.6 Dissonance suppression methods

After the optimal pitch-shift computation, the next step of our mixing process, which is the main core of our work, is the modification phase. In order to understand how to develop a method suitable for the purpose of DJ mixing, which could lead us to an improvement of the quality of the resulted mix, we implement different types of strategies. The main idea related to our approach for modify the harmonic content, is to apply specific suppression operations on the spectra of T1 (the track which was fixed in the pitch-shift computation), and then synthesize a modified version of it which will be mixed with the pitch-shifted version of T2. Our first approach consists in identify and then suppress the time frames where, by looking at the whole time length of the music mix, the roughness measurement evoked by combining the tracks together produce a significant high value (i.e. the time frames which results to be measured as more dissonant in the final mix). In the following, we referred to this attempt to modify the music signal as temporal masking.

4.6.1 Temporal masking

In order to identify the regions of the spectra that produce a significant contribution in generating dissonance, our first step, is to identify where, during the time evolution of the track, these dissonant regions occur. To this end, it was necessary to deeply understand all the information that the roughness model can provide us. In fact, if before we used it to find the consonance maximizing pitch-shift between two tracks, we will now explore its computation in order to find which time frames, with respect to the computed optimal pitch-shift alignment, will produce the most dissonant values. As we already saw in Chapter 3, the roughness measurement that describe a music mix is computed by summing over the whole length of the signal the roughness values calculated locally within each time frame. As a consequence of that, by exploring the computation of the roughness measurement, we are able to extract a value, which represent the roughness generated by the combination of the two tracks in each single time frame.

Our designed method investigates the roughness matrix generated during the pitch-shift ratio computation (Fig. 4.5). By looking at the raw corresponding to the computed optimal pitch-shift, we are able to extract the roughness calculated in each single time frame (Fig. 4.6)**.** By having this information, we are now able to analyse the temporal evolution of the roughness measure, which, as we would expect due to the nature of music, has a non-linear behaviour with respect to time evolution of the signal (Fig. 4.6).

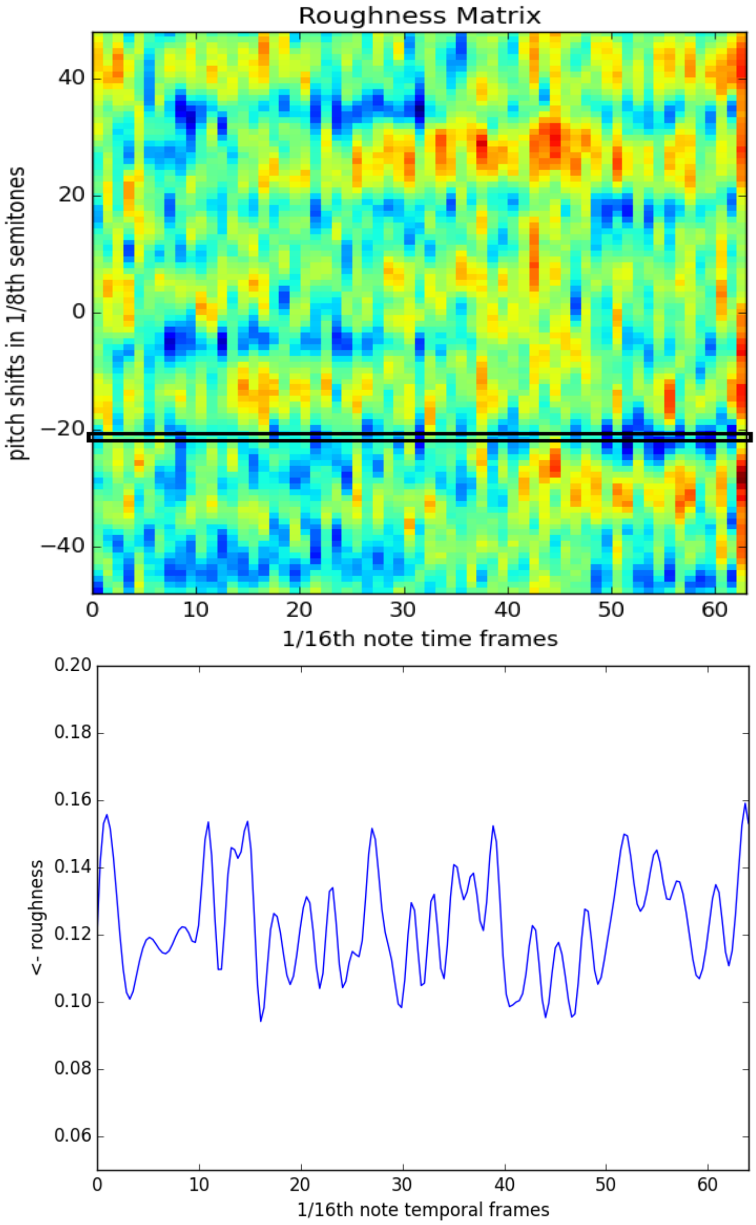


Figure 4.6: Upper plot: Roughness matrix generated from the mix of two training samples. In the square is pointed out the raw corresponding to the optimal pitch-shift. In the lower plot the corresponding visualization of the roughness degree evolution over time for the training mix under the optimal-pitch shift.

Now that we know how to explore the roughness value over its evolution in time, the next step is to select the time frames in which the roughness measurement is maximized. In order to do that and to understand which time frames will be the subject of the suppressing operations, we selected all the time frames in which the roughness measurement overcome a chosen value (Fig. 4.7).

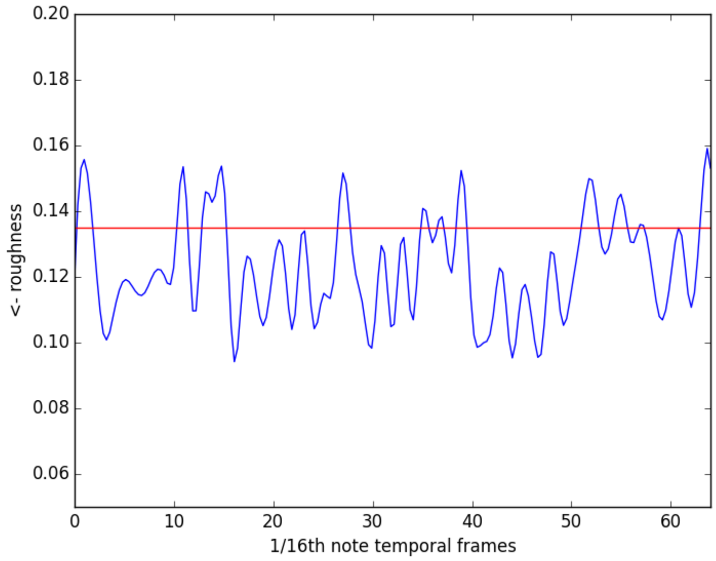


Figure 4.7: Example of designed strategy. Apply a threshold to identify the most dissonant time frames.

Our first approach to set this threshold value was based on the idea that it could have been possible to choose a constant value to define a specific boundary to consider roughness nor consonance or dissonance. However, this kind of implementation was not possible. In fact, due to the complexity of music signal, the roughness measurement is strictly dependent on the nature of the input track. This means that for each possible combination of input tracks, the model produces a unique roughness curve (i.e. the roughness evolution across time), and thus, it is not possible to compute a constant value that would fit as global threshold (Fig. 4.8). Consequently, our implemented criteria, is based on the statistics evaluation overall the roughness measurements. Specifically, the algorithm, by looking at all the roughness values generated for each time frame, computes a percentile value, which with respect to a chosen percentage, acts as the threshold to apply on the roughness time evolution.

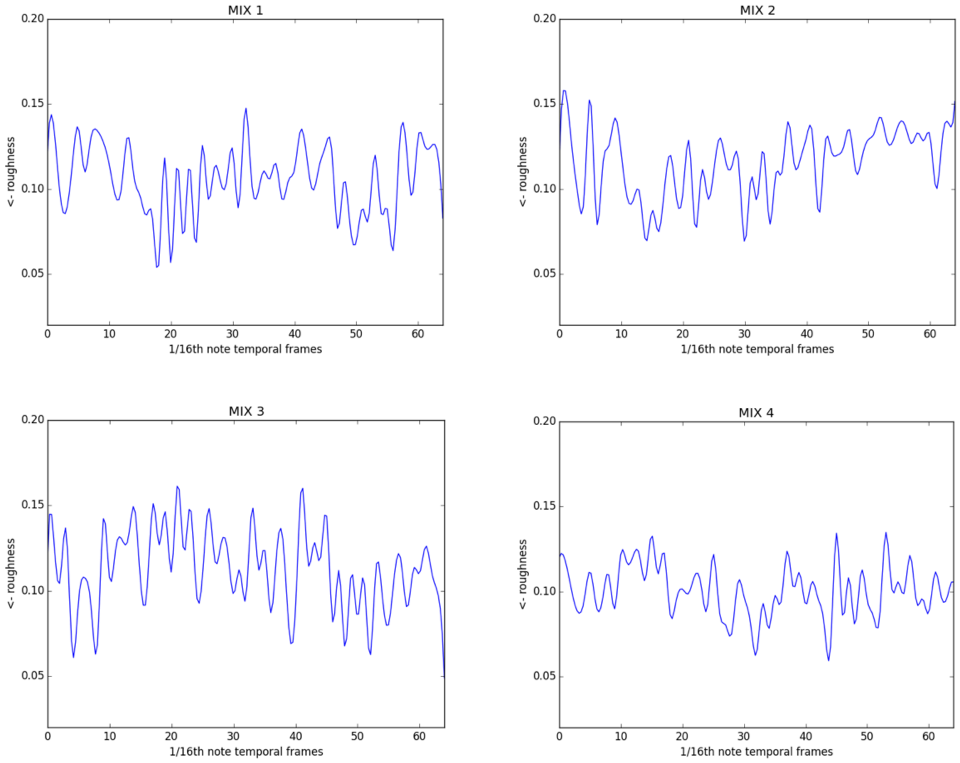


Figure 4.8: Different roughness evaluations of 4 various mixes. From the differences in each curve it is easy to understand that a constant value for a threshold would affect in different way each signal. Thus, the threshold to be applied is strictly dependent on the nature of the input signals.

In Figure 4.9 we can see different threshold values applied on the roughness evolution of two mixed samples. From this figure, it is easy to understand that by increasing or decreasing the percentage value, we can choose how many time frames we want to be modified in the signal. In fact, once the algorithm identifies the set of time frames to be modified, the next step is to apply a suppression operation on the original spectra (i.e. not averaged) of the music samples. Consequently, our next task, was to find a way to eliminate the contribution of these time frames into the final mix.

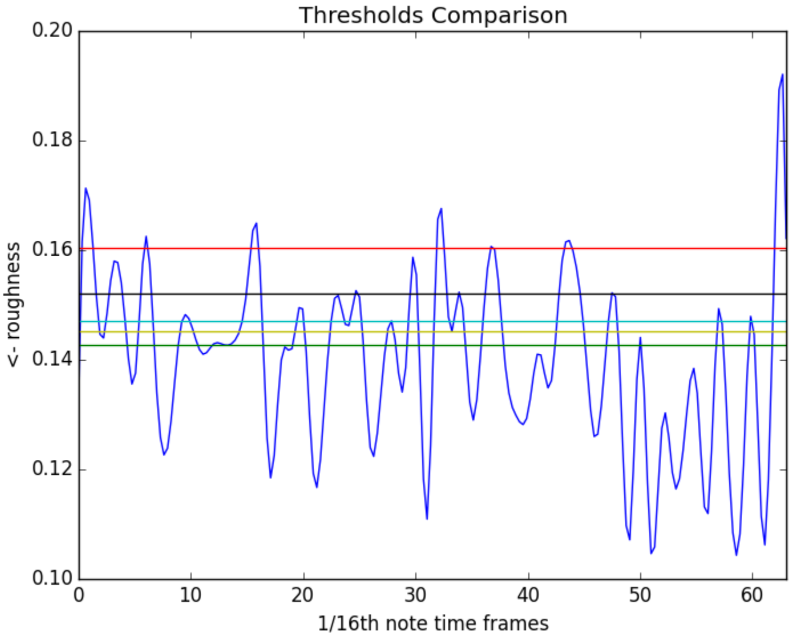


Figure 4.9: Different thresholds applied on the roughness measurement over time for a training sample. Percentile corresponding to 90 (red), 80 (black), 70(light blue), 60 (yellow), 50 (green) percentage of the overall roughness degree.

The suppression method, by taking advantage of the sinusoidal model, works in a straightforward way. Specifically, the algorithm, for each of the selected time frames, decreases the amplitudes values of 40db for all the 20 partials that describe it. This means that the algorithm is masking the music signal over its temporal evolution. The fact that by applying this method the synthesized version of T1 appears to be truncated in its continuity over time, which as a consequence results in unpleasant listening experience (Fig. 4.10) suggests the need for a more sophisticated approach.

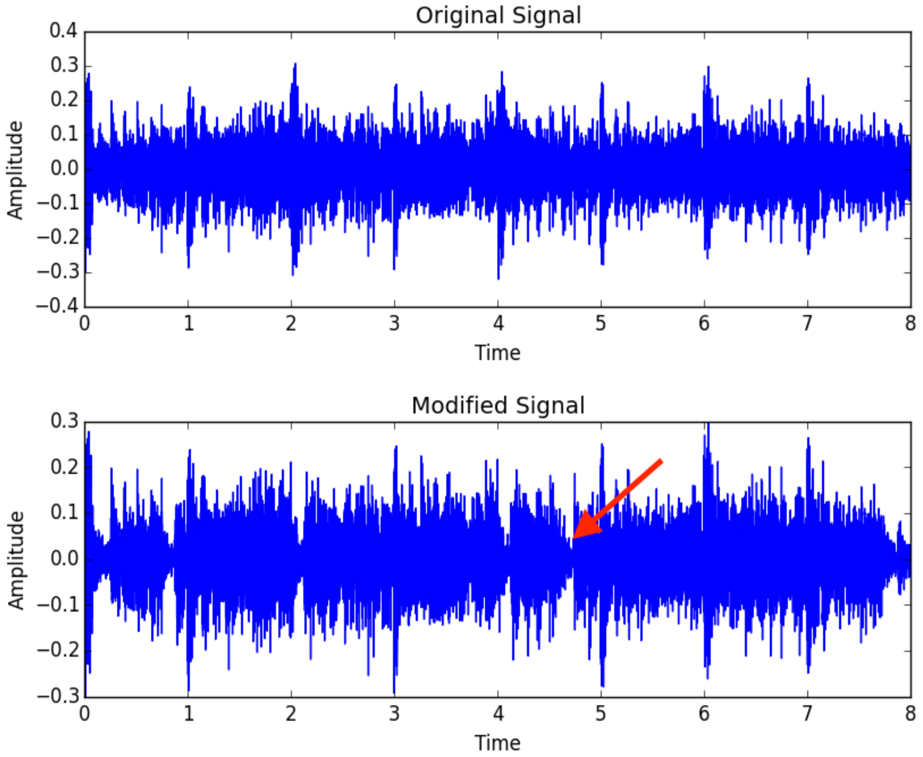


Figure 4.10: Waveforms corresponding to the original signal and the modified one. It's easy to see that in the modified signal the waveform results to be truncated along its time evolution (an example is outlined by the red indicator).

4.6.2 Partials suppression

With the temporal masking method, we have seen how, by exploring the roughness model, we are able to identify the time frames that produce a significant dissonant effect on the final mix. In order to expand the ways in which we can modify the spectral content to improve the consonance of the final mix, we investigate deeply onto the computation of the roughness measurement. Specifically, the idea is to identify the particular set of partials, within each time frame, which consistently raises the roughness values (i.e. produces the most dissonance effect). To this end, we explore how the roughness computation is calculated within each single time frames. In other words, we are looking at the roughness value which is generated by the collision of each of the 20 partials of T1, with all the pitch shifted partials describing T2, for every of the 64 time frames.

In order to do that, we first have to compute a pitch-shifted version of the averaged partials describing T2. To this end, our implementation follows directly from the frequency shift used in the roughness model (eq. 3.8). We define the set of 20 partials describing T2 in a time frame *t* as $f_2(t)$, and mathematically apply the pitch-shift as:

$$f_{2,i}(t) = 2^{log_2(f_{2,i})+\frac{s-48}{96}} \quad i = 0, \dots, n; \tag{4.5}$$

where *i* ranges over the selected number of partials *n* (20 the default implementation), and *s* is the optimal pitch-shift ratio expressed as the corresponding number of 1/8th semitone steps (i.e., the raw index of the roughness matrix, [0, 96])

Once we have the pitch-shifted frequency values of the partials of T2, we are now able to investigate within each time frame, the roughness generated by the co-occurrence of the single partials from T1 and T2. To this end, for each of the 64 time frames *t*, we merge the partials describing T1, $f_1(t)$, with the pitch shifted partials $f_{2,s}(t)$ of T2, and we do the same with the amplitude values $m_1(t)$ and $m_2(t)$. As a results we obtain due vectors:

$$f(t) = [f_1(t), f_{2,s}(t)] \quad \text{and} \quad m(t) = [m_1(t), m_2(t)]. \qquad (4.6)(4.7)$$

We then calculate the roughness measurement *g(y)*, according to equations (eq. 3.4) in chapter 3 with the merged frequencies and amplitudes as input. Where *y* is the frequency interval between two partials f_j, f_k expressed in critical bandwidth of the mean frequency $\bar{f}$:

$$\bar{f} = \frac{f_j+f_k}{2} \quad \text{and} \quad y = \frac{|f_j-f_k|}{CBW(\bar{f})}. \qquad (4.8)(4.9)$$

The roughness values computed for every pair of partials *j, k* are then weighted by the corresponding amplitude values to obtain the final value describing the roughness measurement as:

$$d_{j,k} = g(y_{j,k})\frac{m_j + m_k}{2}. \tag{4.10}$$

By looking at degrees of roughness produced by this computation, we can generate a roughness matrix for a single time frame, which describes the interaction in terms of roughness evaluation of all the partials (of T1 and T2). Figure 4.11 illustrates the computed roughness matrix. In this way we can observe the interactions of the roughness-creating partials between the two tracks in a given frame.

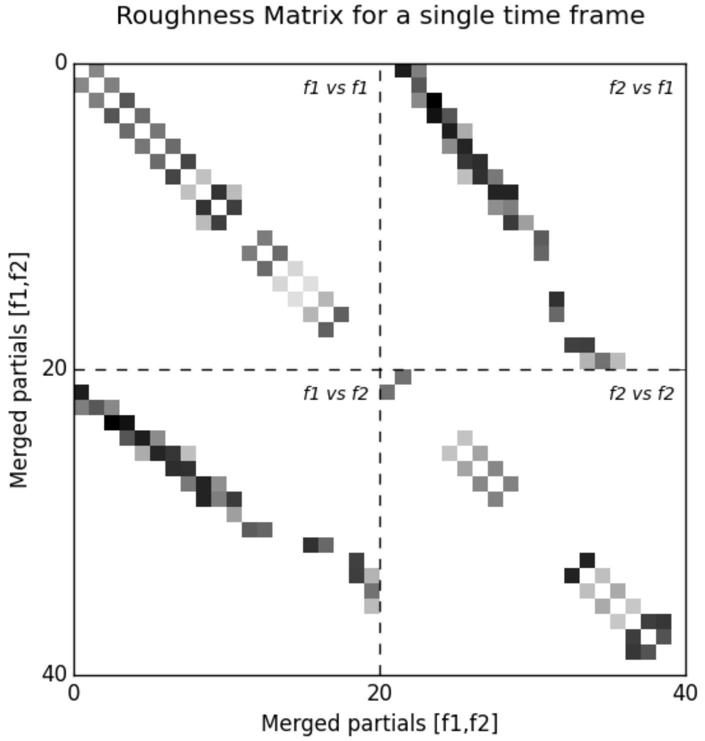


Figure 4.11: Visualization of the roughness matrix from Eq. 4.10: for the frequencies f 1 for one temporal frame of T1 and f 2 for the same frame of T2. Darker shades indicate higher roughness. The frequencies are sorted in ascending order per track to illustrate the internal roughness of T1 and T2 as well as the "cross-roughness" between them.

As stated above, our aim in this modification method is to identify from each time frame of T1 the set of partials that give a high contribution in increasing the roughness measurement. To this end, by exploring the roughness matrix of singular time fames, we are now able to select those partials of T1 presenting higher degree of roughness when combined with those of T2. The selection criteria come directly from the one we used in the temporal masking. By looking at all the positive values of the roughness matrix, we extract a percentile value, which represents the threshold for our selection. This means that the partials of T1, which when combined with those of T2, result in a roughness matrix with a value higher than the chosen threshold, are consequently selected to be suppressed. Once the algorithm has completed the selection process, in order to eliminate the contribution of the selected partials in the final mix, it applies a suppression method that relies on the same method used in the temporal masking. Hence, for each of the selected partials, it suppresses the corresponding amplitude by decreasing it of 30db.

This modification method, as we expected, results in decreasing the roughness degree computed over the whole length of the final mix. By comparing the roughness curve generated from the two tracks before the modification phase (T1 original sample, T2 pitch-shifted) (Fig. 4.12), with the roughness curve calculated with the modified and re-synthesized version of the fixed track (T1 modified, T2 pitch-shifted), we can see that the overall degree of roughness decreased.

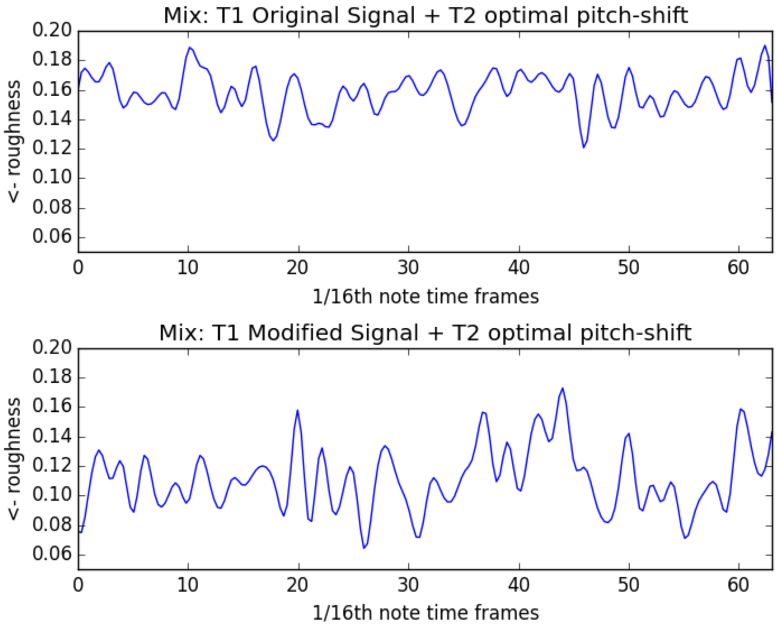


Figure 4.12: Different roughness evaluation pre and post modification. From the comparison of the two plots it is possible to see how, the roughness evolution curve describing the same mix, appear to be diminished, and consequently the overall roughness value is decreased

Even if applying this method results in decreasing the roughness degree generated in the final mix, we believed that it was possible to improve the modification process, by expanding the searching criteria. In fact, by running an informal experimentation on our partials suppression method, we identified a consistent limitation in the roughness computation. Specifically, we recognize how, the described roughness model, cannot provide us with the computation of the degree of roughness for two partials that are separated by an interval that is higher than an octave. As a matter of fact, the degree of roughness *g(y)* computed by the equation (eq. 3.4) results in non-zero values only when the condition $y<1.2$ is satisfied. But from our investigation we saw that, if the interval between the two investigated partials is higher than an octave, the relative y critical bandwidth interval will always have a value higher than 1.2. It is possible to notice that also from the graph representing the curve proposed by Plomp and Levelt (Fig. 4.13), the roughness curve tend to zero for high value of the critical bandwidth under observation.

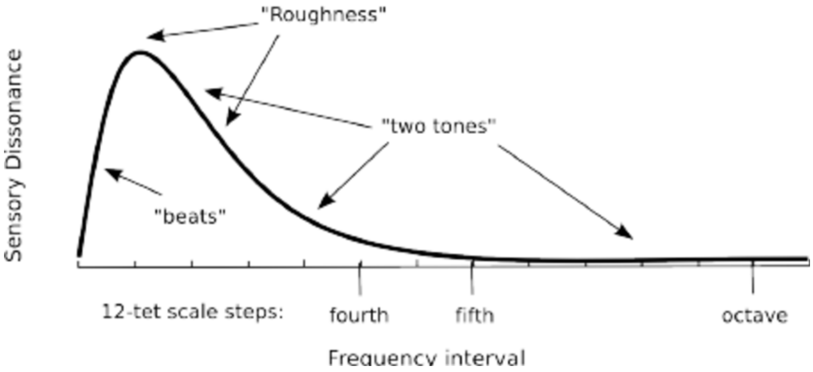


Figure 4.13: Roughness curve measured with respect to the music intervals domain. From this visualization it is possible to see that the function describing this curve tent to zero value for frequency interval higher than an octave. [Image from www.researchgate.net]

As a consequence of this, combined with the fact that, as we saw in chapter 3, every piece of music consists of different tones sounding simultaneously and each of them possibly describing a harmonic pattern, we realize that we needed to improve our partials suppression method. To this end, we choose to expand the selection process by combining it with a searching criterion that investigate on matching the already selected partials with harmonics-corresponding partial in a chosen range of adjacent octaves.

4.6.3 Harmonic matching

The selection criteria of the frequency suppression method, allow us to identify within each time frame the partials that increase in a consistent way the dissonance of the final mix. Although, we have seen that the roughness model is not able to identify the dissonance generated by the combination of partials whose frequency value does not belong to the same octave. To this end, we expanded the selection criteria by searching the spectra for harmonic matching partials.

The matching criteria work as follow: for each partial of T1, within the same time frame, which was selected as most dissonant from the roughness computation (and thus, to be suppressed), the model analyses the frequency spectra and searches for partials which correspond to a harmonic correspondence of the selected one. Hence, for each of the selected frequency values f_0, the algorithm analyzes the frequency content, both in higher and lower directions of the spectra, for a chosen number of octaves $\pm k$. To this end, we need to recall that, as we saw in chapter 3, the interval of an octave is, physically speaking, the doubling of f_0 for an octave up and, respectively, the halving for an octave down. The algorithm, by following this relation, starting from the selected frequency f_0, in order to find partials which correspond to harmonics correspondences, analyses the spectra by looking at an interval of +- half a semitone centred around the theoretical computation of the harmonic value for the investigated frequency. If it identifies some partials that are matching the harmonic value, and those were not selected from the prior partials selection criteria, it marks them as partials to be suppressed. This expansion of the partials suppression method allows us to incorporate concepts of harmonic theory with the psychoacoustic approach of the roughness model. In this sense we can suppress partials in a more "musical" way, since harmonically related partials are more likely to correspond to musical notes than simply suppressing unrelated partials.

4.7 Post-processing

Once the phase relative to computation of the modify operations is concluded, the last step of our system, in order to apply the theoretical results in practice, is the post processing phase. It mainly consists of two operations: the first task is to apply the estimated pitch-shift to the audio-signal of T2, and the second is to merge a synthetized version of the modified harmonic content of T1 with the corresponding extracted residual. The pitch-shifting computation of T2 was realized with the software utility "Rubberband". To merge the harmonic content of T1 with its residual we adopt the dual operation of the subtraction we used to extract it. Thus, we simply add the two signals together and as a result we had the final modified version of T1. To avoid loudness differences causing an effect on consonance perception that might have been artefacts of the pitch-shifting process, each audio excerpt was normalized to a reference loudness level using Ableton Live.

Chapter 5
Experimental results

This chapter is dedicated to the description of the experiments that we performed as well as the definition of the evaluation metrics and the specific experimental setups we used. The chapter is divided into two sections. In the first we will present the evaluation metrics and the tests used to measure the performances and the robustness of our spectra modifying methods. The second section is dedicated to the description of the evaluating listening experiments and to the discussion of the results.

5.1 Objective testing of the suppression methods

In this section, we will provide a description of the evaluation tests we ran on each of the different approaches presented in our system. In order to analyse the performance and the functionality of our presented strategies, we ran several tests with different types of signal. To this end, for each of the introduced methods, which we recall are: a) Temporal masking, b) Partials suppression and c) Partials suppressions and harmonic matching we provide two different examples of evaluation. Specifically, we tested the response of each approach for: 1) a pure tone signal, and 2) a complex tone signal. It's important to note that we ran this set of tests only on the suppression methods and not on the entire system. This means that for all of the following tests the roughness analysis and the suppression operation is processed only on the original input signals without taking into account any pitch-shifting operation.

For these types of informal experiments, we generated different signals based on the constraints defining the input samples of our system provided in Chapter 4 (i.e. 16 beat excerpts at 120 bpm). All the samples have been created with the already introduced DAW Ableton Live.

5.1.1 Temporal masking

As described in the previous chapter, this method aims to find and suppress in one of the input samples, the time frames that, by means of the analysis on the roughness computation, contribute the highest dissonance to the final mix.

Testing with pure tones

The first experiment was designed using only pure tone signals. The first input signal, which will not be the object of the modification (i.e. it acts as the track that would be pitch-shifted), consists of a pure sine wave corresponding to a C6 (frequency around 1046 Hz). The second is composed as the alternation of a C6# and a G6 every 4 beats, which correspond respectively to the minor second and perfect fifth of a C6 (i.e. most dissonant and most consonant intervals in one octave range). Figure 5.1 provide a visual representation of the spectrograms of the two input signals.

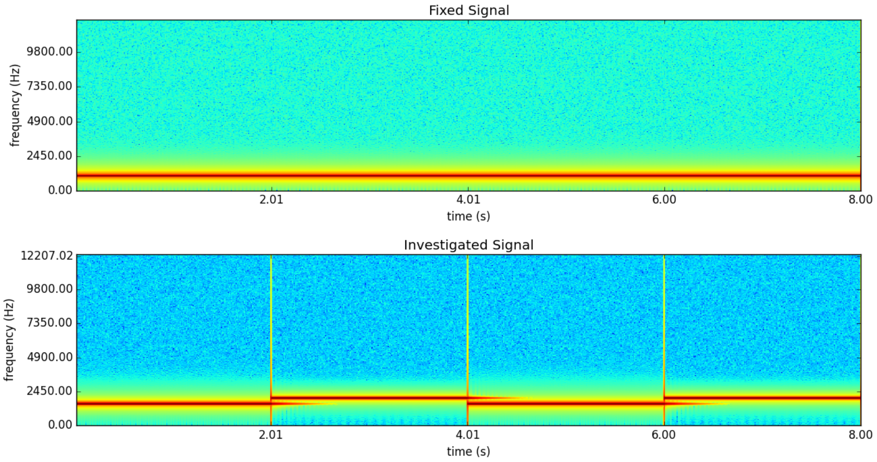


Figure 5.1: Spectrograms of the two input test samples. The upper plot shows the frequency content of the signal of a pure sine wave corresponding to the C. The second plot shows the spectrogram of the signal composed as the alternation of a C# and a G pure sine wave.

By setting these samples as input, we expect that the algorithm applies transform operations on the spectra of the investigated signal, specifically in those time frames where the C# appears (i.e. most dissonant time frames). As we can see from figure 5.2, which represents the spectrogram of the investigated signal after the modifications, the algorithm identified and suppressed 4 time frames (threshold value set at 95%) in the spectral regions which corresponded to the C# (most dissonant), and thus worked as expected. In order to suppress all the time instants where the C# is present we would need to drastically increase the number of frames to remove by reducing the threshold.

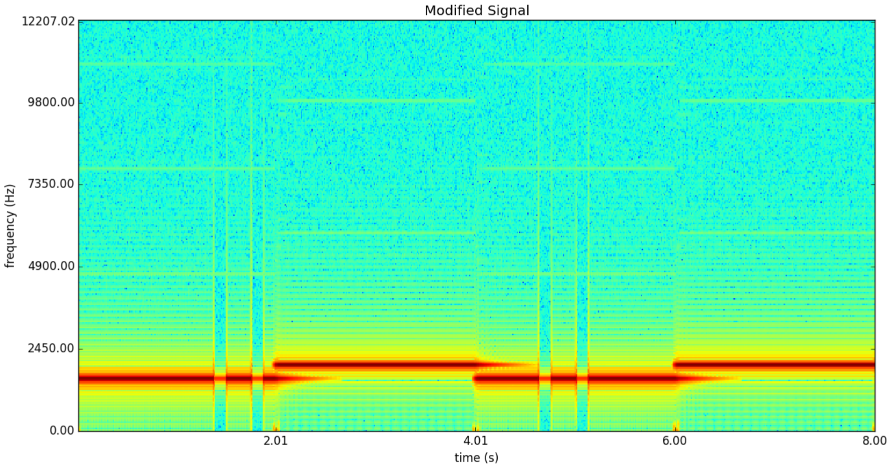


Figure 5.2: Spectrogram of the modified signal. From this visualization we can see how the method transform the spectra by suppressing the amplitude of the partials that belong in the time frames corresponding to the C# sine waveform.

Testing with complex tone

The second test was designed using complex signals; it follows directly from the evaluation criteria used in the case of pure tones. Hence, the first input signal, which will not be the object of the modifications (i.e. the track to be pitch-shifted), consists of the repeated playing of a C6 performed on a synthetized piano (fundamental frequency around 1046 Hz, with 8 harmonics) every 4 beats. The second signal is composed as the alternation of a B6 and a G6 every 4 beats, played through a synthetized pan flute (6 harmonics of the fundamental) which correspond respectively to the perfect fifth and major seventh of a C6 (i.e. most consonance and most dissonance intervals in the same octave). Figure 5.3 provide a visual representation of the spectrograms of the two input signals.

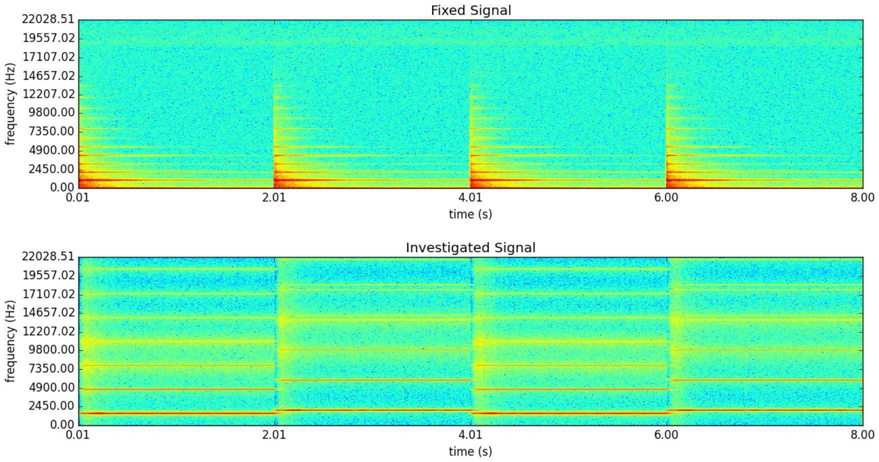


Figure 5.3: Spectrograms of the two input test samples. The upper plot shows the frequency content for the signal relative to the single C note played through a synthetized piano. The second plot shows the spectrogram of the signal composed as the alternation of the notes G and a B played through a synthetized flute.

By running the system with these input samples, we expect the algorithm to transform the spectra of the investigated signal in those time frames where the B note appears (i.e. most dissonant time frames). As we can see from figure 5.4, which represent the spectrogram of the investigated signal after the modifications, the algorithm identified and suppressed 4 time frames (threshold value set at 95%) in the spectral regions that corresponded to the B (most dissonant), and thus worked as expected. Again, as with the pure tones example, a lower threshold would be required to remove all of the B frames.

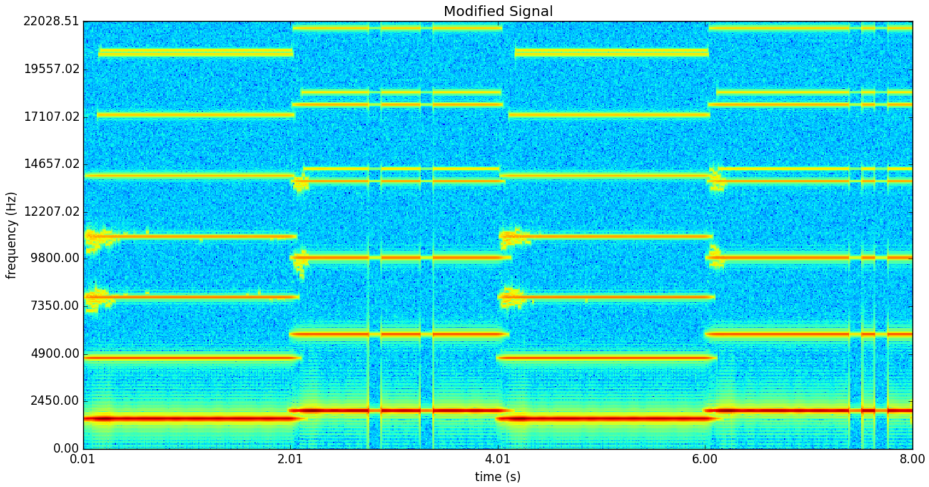


Figure 5.4: Spectrogram of the modified signal. From this visualization we can see how the method transform the spectra by suppressing the amplitude of the partials that belong in the time frames where the B note is played.

5.1.2 Partials suppression

The second and more consistent method we introduced, as we have seen in the previous chapter, aims to find and suppress in one of the input samples, the set of partials that, within each time frame and by means of the analysis on the roughness computation, contributes the most dissonance to the final mix.

In order to evaluate the performance and the functionality of this approach we ran the system with different tests on the input signal by following the evaluation metrics used to test the temporal masking. Hence we tested it first with pure tones and then with complex tones as inputs.

Testing with pure tones

As in the case for temporal masking evaluation, the first experiment was designed using only pure tone signals. The first input signal, which will not be the object of the modification (i.e. the track to be pitch-shifted), consists of a pure sine wave corresponding to a C6 (frequency around 1046 Hz). The second instead is composed as the superposition of a B6 and a G6 (frequency respectively 1975 and 1567 Hz), which as we have seen before, correspond respectively to the major seventh and perfect fifth of a C6 (i.e. most dissonant and most consonant intervals in one octave range). Figure 5.5 provides a visual representation of the spectrograms of the two input signals.

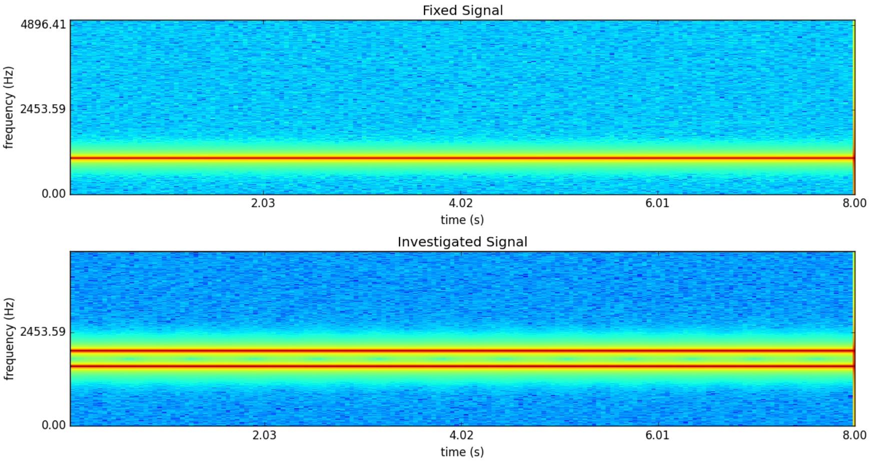


Figure 5.5: Spectrograms of the two input test samples. The upper plot shows the frequency content for the signal relative to the sine waveform describing a single C note. The second plot shows the spectrogram of the signal composed as the superposition of two sinusoids corresponding to the notes G and a B.

By setting these samples as input we expect the algorithm to apply a transform operation on the spectra of the investigated signal and specifically suppress the partials corresponding to the B (i.e. most dissonant partials). As we can see from figure 5.6, which represent the spectrogram of the investigated signal after the modifications, the algorithm has identified and suppressed the partial corresponding to the B note and thus worked as expected.

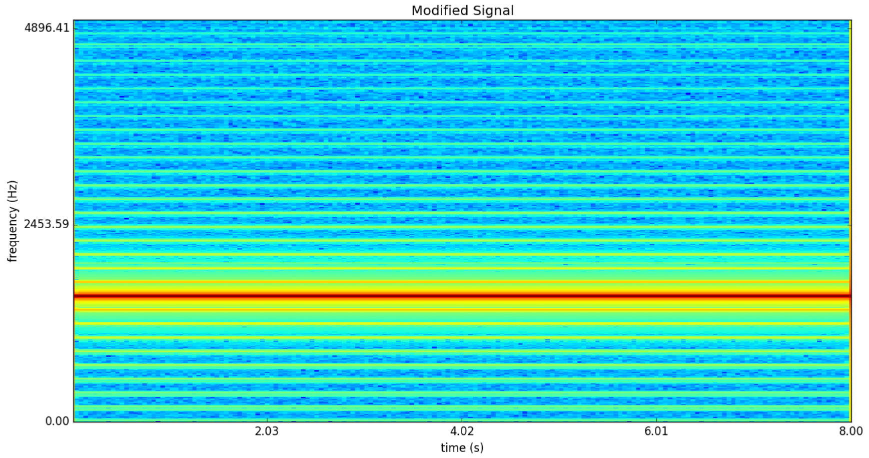


Figure 5.6: Spectrogram of the modified signal. From this visualization it is easy to see how the method transform the spectra by suppressing the sine wave corresponding to the B without altering the one describing the G.

This kind of test was very informative, not only because it provided a good feedback on the partials suppression method, but also because, by running it with different configurations of the inputs pure tones, it illustrates a significant limitation of the roughness model. We saw that the roughness model cannot provide us with a measure of the roughness, which occurs between two pure tones that belong to different octaves. In fact, by looking at the roughness curve proposed by Plomp and Levelt [37], we can see how it can only provide us the information on how two frequencies relate to each other in terms of roughness only if they belong to the same octave. This means that if we compare a pure tone (e.g. a sine waveform corresponding to a C6), with the second diminished of a higher octave (C7#), which should result in a high value of roughness, instead it provides us with a roughness value corresponding to zero. As a consequence, we can state that the implemented roughness model is not able to compute the roughness that occurs between two partials when these are separated by an interval of an octave or more.

Testing with complex tone

The second test, as we did for the temporal masking, was designed using complex signals, and it follows directly from the evaluation criteria used in the case of pure tones. To this end, the first input signal, which will not be the object of the modifications, consist of the continuous playing of a C5 performed on a keys synthesizer (fundamental frequency around 539 Hz, and 11 harmonics). The second signal instead consists of the continued playing of a G5 (fundamental frequency around 1005 Hz), using a second synthesizer, plus single sine waveform corresponding to the major seventh of the tonic, which as we saw before corresponds to a B5 (corresponding frequency value 988 Hz). Figure 5.7 provides a visual representation of the frequency spectra of the two input signals.

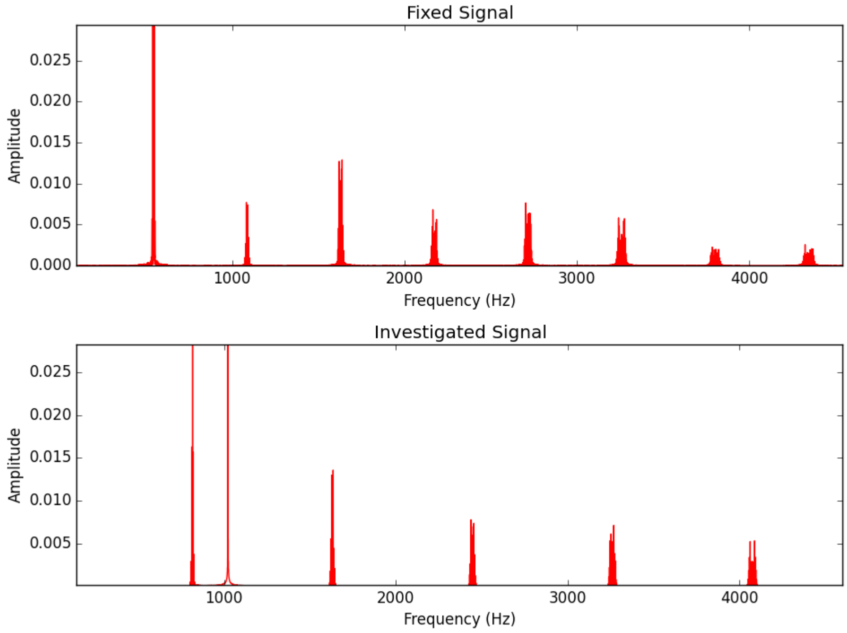


Figure 5.7: Plots of the frequency spectra for the two input samples. The upper plot represents a visualization of the frequency content describing the signal composed as the continuous playing of a C note. It is easy to recognize the first peak as the fundamental frequency and the following one, which correspond to the harmonic of the fundamental. In the lower plot, the second input sample, where the fundamental of the B is flanked by the pure sine signal relative to the B note.

By running the system with these input samples, we expect the algorithm to consistently transform the spectra of the investigated signal by suppressing specific components. Specifically, it should identify the partials corresponding to the B note (i.e. the partials which produce most dissonance). As we can see from figure 5.8, which represents the spectrogram of the investigated signal after the modifications, the frequency content corresponding to the B note was suppressed, and after the modification results in a lower amplitude value.

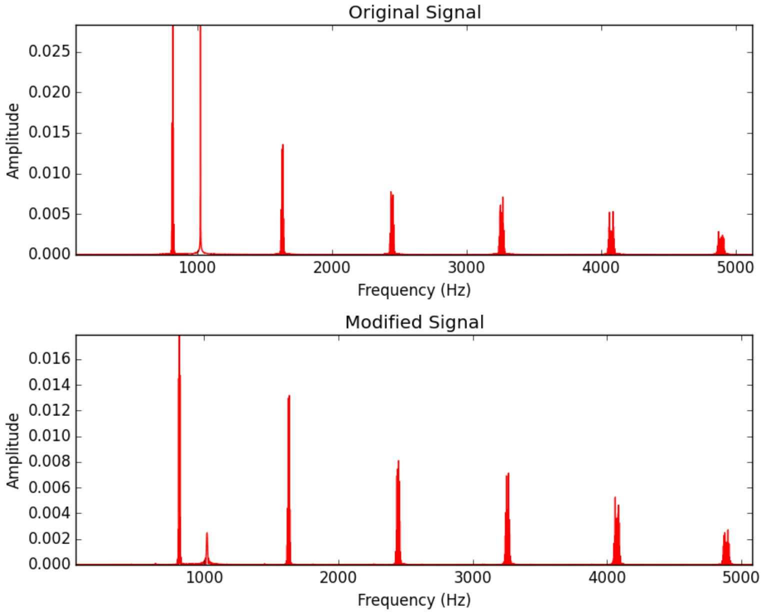


Figure 5.8: Plots of the frequency spectra for the modified signal. The upper plot represents a visualization of the signal before the modification phase instead the second plot figures the same signal after the spectra transformations. It is easy to recognize how the method identified and suppressed the B note. As we can visualize in this plot, the modification did not totally delete the harmonic content corresponding to the B, but only reduced its amplitude, as this is a suppression procedure.

5.1.3 Harmonic matching

The harmonic matching criterion was designed to overcome the limitation of the roughness model where it is limited to a single octave. Indeed, the matching principle aims to find and suppress the harmonic correspondences of the frequencies selected from our partials suppression method. In order to evaluate the performance and the functionality of this approach we ran the system by following the evaluation metrics used to test the partials suppression method. We tested it with pure tones signals as inputs.

Testing with pure tones

As in the previous cases, the first experiment was designed using only pure tone signals. The first input, which will not be the object of the modification, consists of a pure sine wave corresponding to a C6 (frequency around 1046 Hz). The second instead is composed as the superposition of a C6, a C6# and a C7# (frequency respectively 1108 and 2217 Hz). We already saw how the partials suppression method is able to identify the partial corresponding to the C6# as a component to be suppressed (second minor, i.e. dissonant interval). However, due to the limitation of the roughness model, it would not be able to identify the C7# (which should be considered dissonant as well). Figure 5.9 and 5.10 provide a visual representation of the spectrograms of the two input signals.

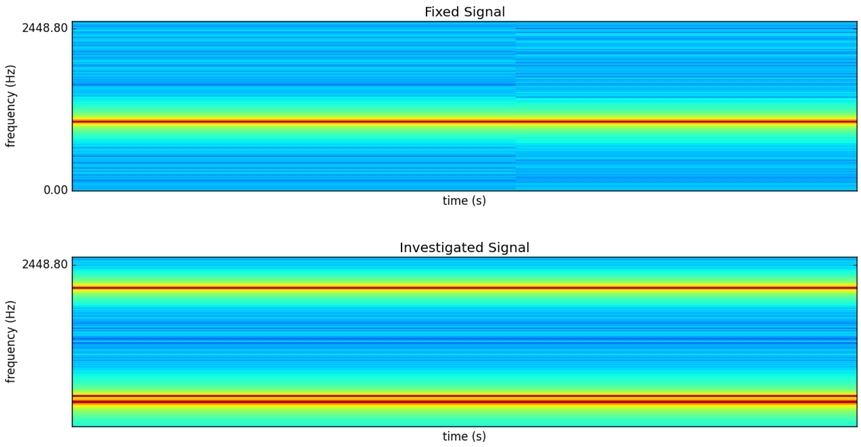


Figure 5.9: Spectrograms of the two input samples. The upper plot shows the frequency content for the signal relative to the sine waveform describing a single C note. The second plot shows the spectrogram of the signal composed as the superposition of three sinusoids corresponding to a C and two c sharp belonging to different adjacent octaves.

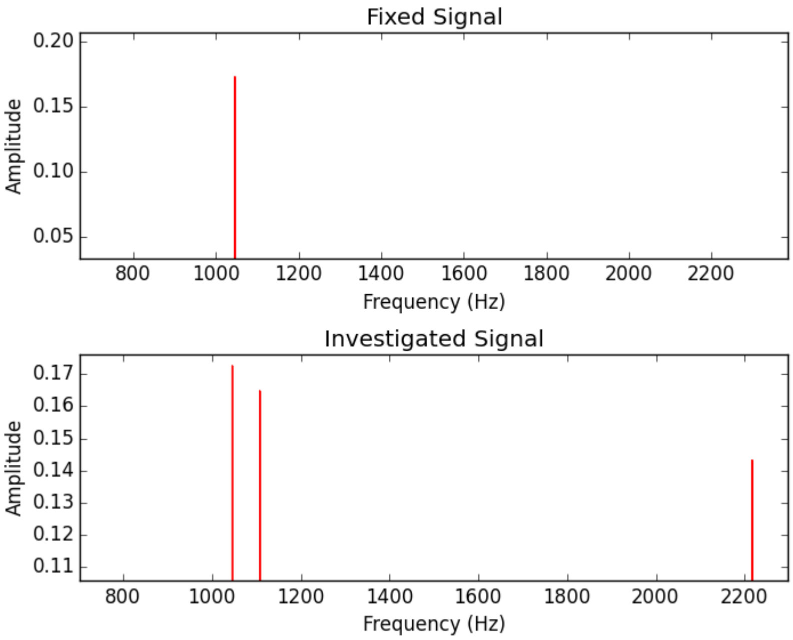


Figure 5.10: Plots of the frequency spectra for the two input samples. The upper plot represents a visualization of the frequency content describing the signal composed as the continuous playing of a C note. In the second one, the investigated input sample, where the C sine is the first frequency peak and the second and third are the two C# notes.

By running the system with these input samples, we expect the algorithm to be able to identify and suppress also the two partials corresponding to the C7#: the first one which belongs to the same octave of the C note and is selected using the partials suppression method, and the second one, which is from the higher octave and is only recognized by means of the harmonic matching method. As we can see from figure 5.11 and 5.12, which represent the spectrograms of the investigated signal after the modification, the resynthesized audio signal consists only of the partial corresponding to the C note: this means that the algorithm has identified and suppressed both partials describing the two C#, and thus has worked as expected.

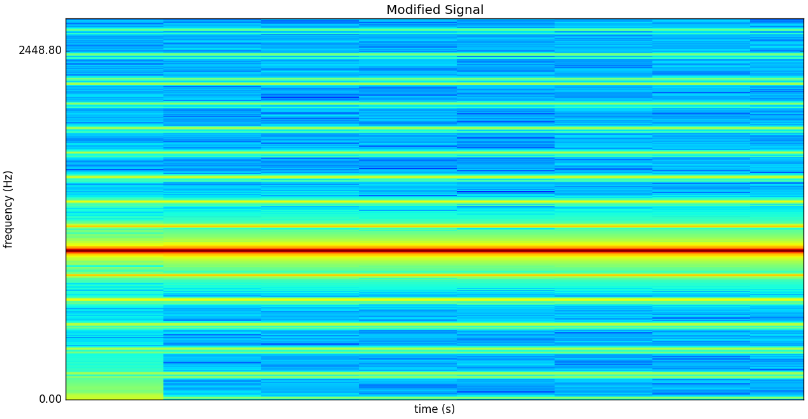


Figure 5.11: Spectrogram of the modified signal. From this visualization it is easy to see how the method transform the spectra by suppressing the sine waves corresponding to the C# notes without altering the one describing the C note.

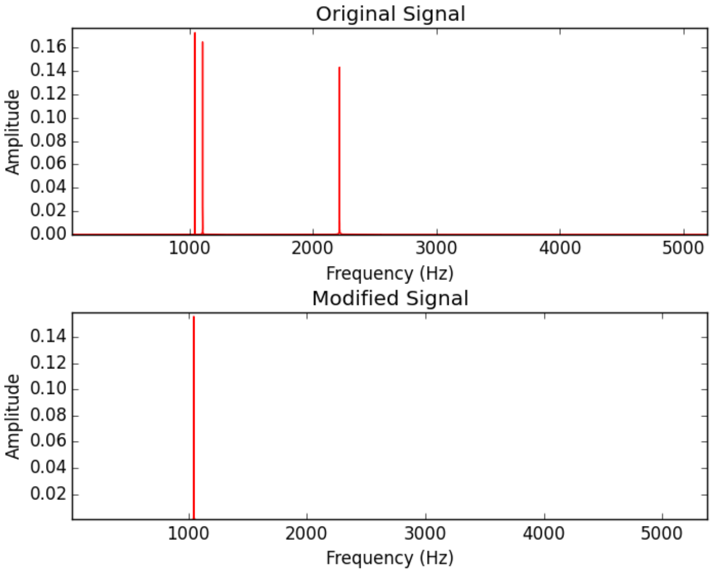


Figure 5.12: Plots of the frequency spectra for the modified signal. The upper plot represents the signal before the modification phase instead the second plot figure the same signal after the transformation. It is easy to recognize how the method identified and suppressed the two C# notes.

This set of tests was only the starting point of our evaluation process to illustrate the behaviour and reliability of the different masking approaches in very controlled conditions. Next, and in order to improve the complexity of our assessment, we tried to experiment on signals describing excerpts of real music.

5.2 Listening experiment

In order to obtain an objective evaluation of our mixing approach, we created a small scale listening experiment. Musically trained participants are asked to rate short mixes for their consonance and pleasantness. The training set of mixes was created according to different outputs of our system. In total we recruited 13 participants whose musical training is determined by them being: music students, practicing musicians, or active in DJing. When listening to each mix, the participants were asked to rate two properties: first, how consonant the mixes sounded, and second they are asked to rate pleasantness of the mixes. While the main goal was to assess the ability of our method to improve the consonance of the mix, the pleasantness question is included to see how closely participants equated consonance with personal taste for harmonic mixing.

5.2.1 Experiment setup

In order to achieve a rigorous evaluation on our methods, we defined two different sets of mixes to be the object of the listening experiment. The first one, where the tracks were subjected to a pitch-shifting operation by means of the described Consonance Based Mixing (CBM), and the second one, where we made no attempt to harmonically align the excerpts (i.e. no pitch-shift). For both of these two sets, populated by 5 different mixes, we created four different conditions:

A. **No masking**: we did not apply any of our spectra modifications methods to improve the harmonic align between the tracks.

B. **Time masking**: we ran the time-masking suppression method on one of the two input samples (the fixed one for the CBM case)

C. **Partials suppression**: we ran the partials suppression method on one of the tracks.

D. **Harmonic masking**: we modified the spectral content of one of the input tracks by combining the partials suppression method with the harmonics matching searching process.

The aim was to remove the same amount of information in each condition. To this end the choice of the parameters for our approach to the automatic harmonic DJ mixing was done in the following way:

- **Sinusoidal analysis**: The input samples were digitized by a sampling frequency of 44100 Hz; the parameters for the partials extraction (in order to respect the constraints we saw in Chapter 4 of this thesis) were set as: length of the window M=4096 samples with a Blackman window, number of STFT bins (coupled with the window length) N=4096, hop size H=256, and number of extracted partials n=20.

- **Temporal masking**: for the first suppression method we choose to select the time frames in which the roughness measurement overcome a threshold value corresponding to the percentile value computed for a percentage of the 65% (which resulted in selecting 23 of the 64 time frames or approximately one third).

- **Partials suppression**: for the second suppression method we set a threshold value corresponding to the percentile computed for a percentage of the 65% of the roughness measurement within the same time frame. This value resulted in selecting an averaged amount of 8 partials for each investigated time frame (i.e. around the 40% of the harmonic content).

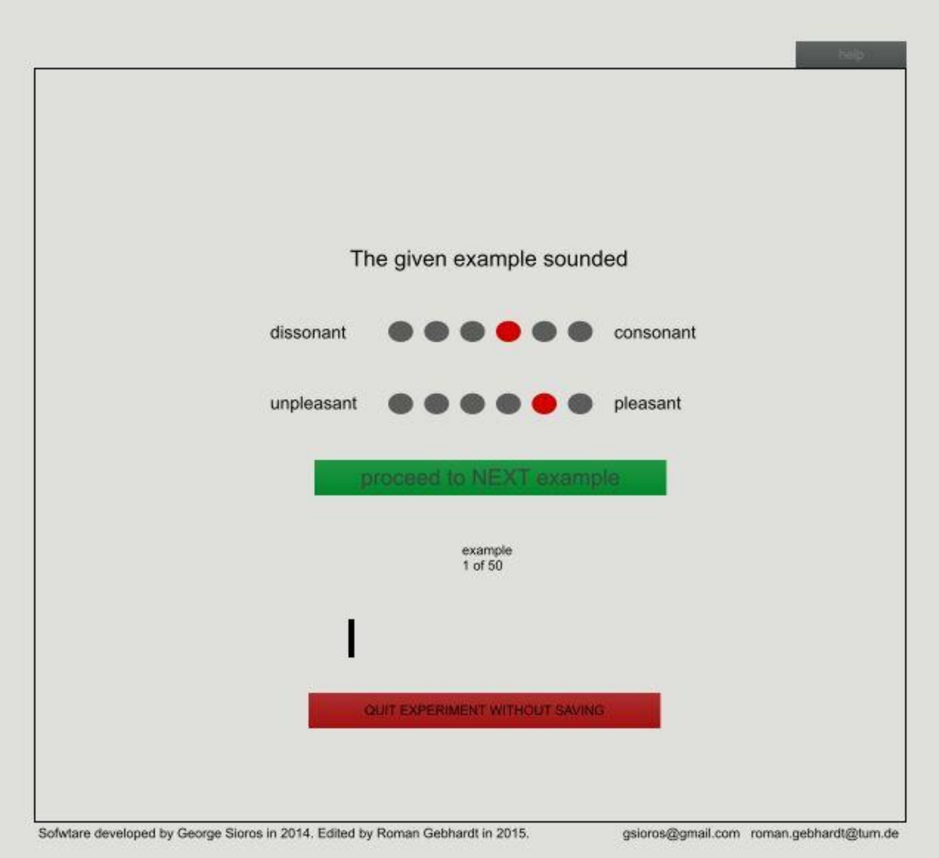
- **Harmonic masking**: to improve the selecting criteria of the partials suppression method, we implemented the harmonic matching by making it search for harmonic correspondences in four different octaves. Specifically, for a selected frequency value, it looks in the spectra at the two adjacent both higher and lower octaves.

- **Synthesis**: For the re-synthesis of the modified signals, we adopt the default value to compute the inverse STFT of 1024 bins using the SMS-tools of Serra.

For each of these modifying methods, the criteria to suppress the dissonance-maximizing regions of the spectra were implemented by decreasing the amplitudes of the selected components by 40 dB. Using a set of 20 excerpts, which respect the constraints described above, the 4 different conditions gave a set of 40 stimuli to be used in the experiment. To allow participants to have more opportunity to understand the mixes, each was looped once to make all the examples 16 s in duration. The order of the stimuli was randomized for each participant. After every single sound example, the ratings had to be given in order to proceed to the next example. To guarantee familiarity with the experiment format from the first stimuli on, a training phase preceded the main experiment and included similar mixes that had to be rated in the same way. Furthermore, the participants were introduced to the experiment with the following instructions:

> *“IMPORTANT: Please read all instructions carefully before starting the experiment.*
>
> *In the following experiment, you will be presented small audio examples and will be asked to rate them in terms of a) musical consonance b) how pleasant they sound to you on a scale from 1 to 6. We define musical consonance to mean:*
>
> *"Musical sounds are said to be consonant if they are perceived to 'sound well' to each other" - Richard Parncutt (1989).*
>
> *Each example lasts for 16 sec. In total there are 40 examples and the experiment should take around 20 minutes to complete. To perform the experiment, please follow this order:*
>
> *1. Click on the play button - note you can only hear each example once. 2. Rate the current audio snippet according to the scales described above.*
>
> *3. Click NEXT to proceed to the next example - this will immediately trigger playback of the next example.*
>
> *You will first start with a training phase, that is to allow you to familiarize yourself with the type of stimuli in the experiment and also to set the playback volume to a comfortable level. You can quit this phase whenever you feel comfortable and proceed to the main experiment.*
>
> *Your participation in this experiment is completely voluntary. If you wish to stop participating, you may do so at any time, but you may only participate in this experiment once.*
>
> *To begin the experiment please click the Proceed button below. By clicking this button, you confirm that you have read and understood the experiment instructions and give your consent for your rating data to be anonymously analysed when reporting experimental results. Please use headphones to perform the experiment.”*

For the performance of the test, both conditions were rated on a discrete six-point scale, using an existing Max/MSP-Patch for listening experiments (Fig. 5.13), designed by George Sioros and later used by Roman Gebhardt in [5].



The given example sounded

dissonant ○ ○ ○ ● ○ ○ consonant

unpleasant ○ ○ ○ ○ ● ○ pleasant

example
1 of 50

Sofwtare developed by George Sioros in 2014. Edited by Roman Gebhardt in 2015. gsioros@gmail.com roman.gebhardt@tum.de

Figure 5.13: Screenshot of the experiment interface designed in Max/Msp.

Regarding our hypotheses on the proposed conditions, we expected condition A (No suppression method) to be the least consonant for both the cases (CBM and no shift), most likely followed by B (Temporal masking). Of the remaining conditions that attempted to suppress the dissonance of the harmonic alignment, we expected the following order of consonance: C (partials suppression) followed by D (Partial suppression combined with harmonic matching).

As a matter of fact, by applying a pitch-shifting operation on the tracks solely by means of the roughness model (CBM case), can already provides a very good result in terms of pleasantness of listening to the final mix. Indeed, our ambition is very high: we are trying to make the good sounding mixes to sound even better.

5.2.2 Results

Inspection of Figure 5.14 and 5.15, which shows the average ratings per excerpt across all conditions respectively for consonance and pleasantness, highlight a wide range of ratings with some mixes considered very high in terms of consonance and pleasantness, while others were rated very low. The analysis of the experimental data shows very similar results for the average ratings for consonance and pleasantness. From the results we can see how they are very highly correlated, which supports the assumption that a high level of consonance, and vice versa a low level of dissonance, can be seen as a major factor for a good sounding mix, which is a fundamental theory of this approach.

For the set of mixes subjected to the Consonance Based Mix, in 3 out of 5 cases (mixes) at least one of the conditions where a suppression method was applied (B, C, D) was rated more consonant than condition A (no masking). In the case of the no-shift mixing this happened for all the mixes. The cases that did not show the expected result of conditions A being rated less consonant than the other conditions were Mix1 and Mix3. However, it is important to note that for the cases of Mix4, Mix5, Mix8 and Mix10, the ratings for all the conditions are strictly correlated and there is not a relevant difference between the ratings for condition A and those for the suppression methods.

By looking at Mix1 and Mix3 in the pitch-shifted set and at Mix8 and Mix10 for the no-shift, we can see how these cases all have high average ratings (respectively 3.3, 2.82, 2.9, 2.8, and overall average 2.96) for consonance for condition A (no dissonance suppression). This suggests that the mixing of the tracks already results in a very consonance sound (thus with low perceived dissonance) and therefore was always be understood as consonance, no matter what it was subjected to. The same type of argument can be applied to the data produced by Mix4 and Mix5 of the pitch-shifted mixes. Indeed, these mixes result in the highest average ratings for consonance in all the conditions (respectively 3.96 and 3.24). From these data we can state that our suppression methods do not improve the perceived consonance when the original mix was already perceived as very consonant. In other words, when the original no-modified mix is perceived as very consonant, there is little scope for improvement from our methods. This type of results asserts our initial assumption: our methods aim to improve the consonance of a mixing process that already results in producing very consonant mixes, and thus is a very ambitious task. A visualization of the consonance ratings is shown in Fig. 5.14.

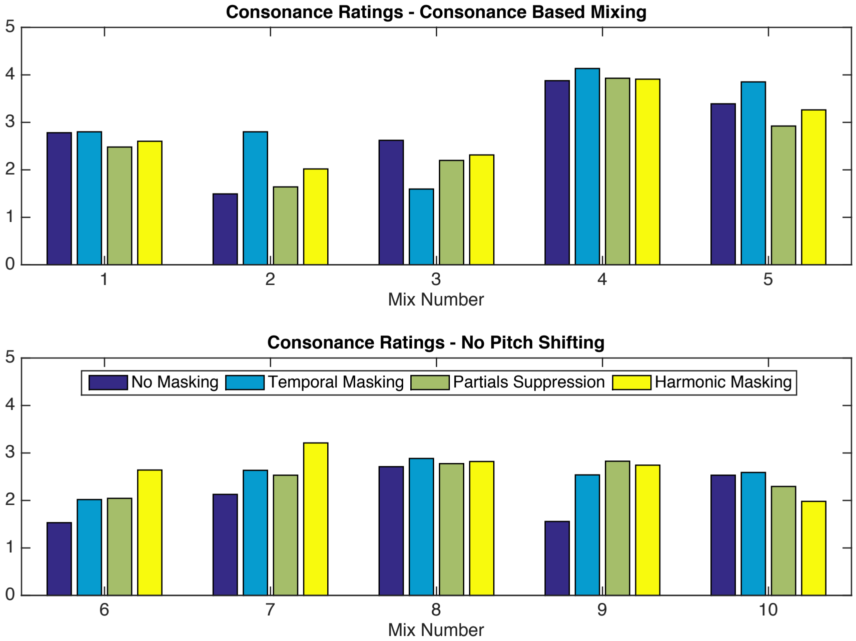


Figure 5.14: Average ratings from participants of the listening experiment for consonance. Please note that although mixes 1-5 and 6-10 are vertically aligned, there is no correspondence between them in terms of the source material, i.e. the content for all mixes are different.

Regarding pleasantness, we obtained the same kind of result for almost every mix. The same mixes that were rated as more consonant under the effect of the suppression methods resulted to be rated also as more pleasant. Furthermore, even if the consonance of the mixes was not perceived as improved, in two cases (Mix1 and Mix3) these were considered as more pleasant than the original not-modified mix. Consequently, this allow us to underline the tendency that minimizing the amount of roughness in the mixed music signal leads to more consonant and thus less dissonant results. A visualization of the pleasantness ratings is shown in Fig. 5.15.

Comparing conditions D with C shows that the addition of the harmonic matching model did improve the ratings for the perceived consonance when compared to those of the partials suppression method as we were expecting. In fact, the harmonic approach (D) was rated less consonant then the partials suppression (condition C) only in 2 mixes out of 10. However, in the remaining 8 mixes, it was preferred in terms of pleasantness only 5 times. This might suggest that in some cases the harmonic masking results in reducing the dissonance of the mix, but, due to the fact that it is suppressing a consistent amount of harmonic content, it also results in a less pleasant listening experience than the partial suppression method. However, the inclusion of the harmonic model does appear to provide alternative good mixes.

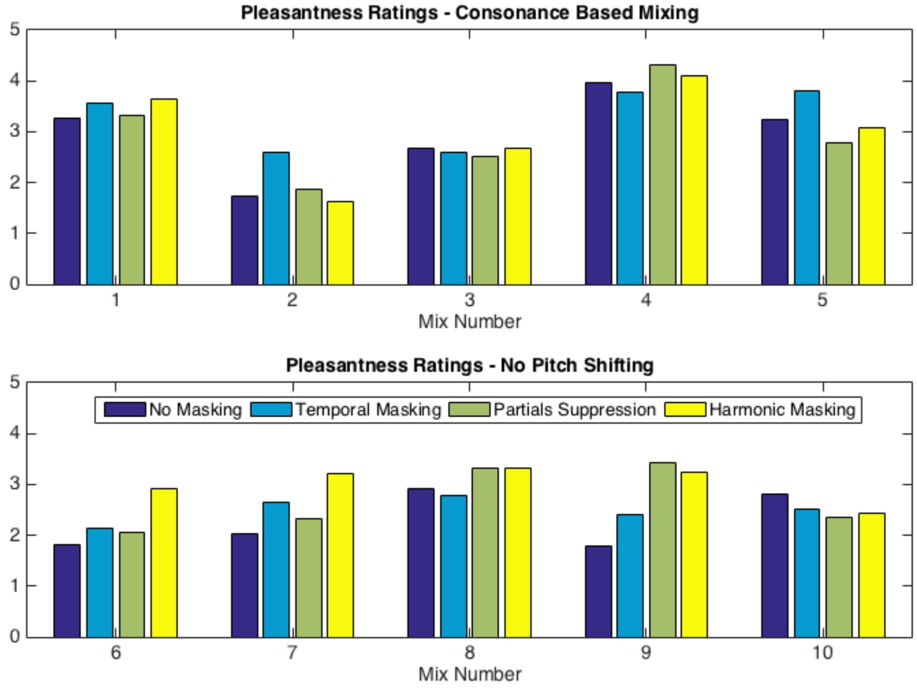


Figure 5.15: Average ratings from participants of the listening experiment for pleasantness

For the cases in which the dissonance suppression methods result in producing a more consonant mix then the not-modified one, the most interesting result is the fact that the developed time masking method was rated as more consonant than the mix resulted from the other conditions in 5 out of the 8 cases. This might suggest that by playing only the time frames from one track that contribute to consonance and suppressing the most dissonant ones, the system is not only mixing the tracks together but is actually creating new music. Specifically, the alternation of playing and silencing one of the two tracks lead to the creation of new harmonic rhythmic patterns in the mixing phase, which result to sound better than we would expect in the first place. This interpretation finds confirm by looking at what real DJs do in practice world, which by using a cross-fader mixer are able to switch simultaneously between the different tracks and create new rhythmic patterns.

Chapter 6
Conclusions and future developments

In this thesis we proposed a new approach for the automatic harmonic mixing of music. We introduce the idea that by modifying the music signal at the spectral level it is possible to achieve better results in terms of listening pleasantness of the mixes. To this end we proposed three different suppression methods that aim to identify and suppress the harmonic content resulting in dissonance in the final mix. The idea behind these methods is to create a model that is able to select and play only the *good parts* from a track while mixing it with a second one. We developed a system that estimated the sensory consonance between two tracks and used it to give both a pitch-shifted version of an input track for optimal consonance-based alignment and three different dissonance-reducing versions of the other track.

From the results of the small scale listening experiment we observed that the transformation of the input signals, by means of one of our introduced suppression methods, led to more consonant and more pleasant mixes of tracks than the simple combination without any adjustment. In comparison with the consonance based alignment (without dissonance suppression), which can be seen as the most recent state of the art, the presented methods improved the consonance of the mix and were preferred in most of the cases by the participants of the listening experiments. In general, we can state that our suppression method improves the consonance of the final mix when the original mix presents a high level of roughness. On the other hand, if the original mixed signal already presents very consonant relations we cannot see a significant change when applying the suppression methods. Since our methods aim at reducing the dissonance of the final mix, if the original mix does not contain highly dissonant components, our approaches are not able to improve the consonance of the investigated mix. In other words, there must be some inherent dissonance in the mix in order for our methods to be able to suppress it.

The assumption that roughness is a good descriptor for dissonance and vice versa its absence leads to consonant sound perception is strengthened by the results of this thesis and also applies in the context of music mixing. Mixed audio that highlighted a high degree of roughness was rated less consonant and less pleasant in almost every case compared to those with less roughness. In fact, roughness seems to play a predominant role in the definition of how two different pieces of music sound when mixed together. Even though, our harmonic matching method was developed to optimize the results of the partials suppression approach. By looking at the data, the sound examples that resulted from the application of the harmonic matching method were actually rated more consonant or pleasant than the ones produced solely by the partials suppression in the most of the cases. These results enhance our starting assumption that the incorporation of an aspect based on harmony is of critical importance in a system that aim at aligning music harmonically according to consonance.

With regard to future work, a first important point is to test the experimental data for statistical significance. The mean ratings in the described experiment offer an impression by inspection, but a more elaborate significance test needs to be done, based on the collection of rating data from a larger group of participants. Furthermore, as it has been a first trial to suppress dissonance by means of a concept of consonance analysis for music material, an interesting object of investigation will be to find a weight for the optimal amount of dissonant harmonic information to be suppressed. Due to the limitations of our presented methods – such as the fact that its results strictly depend on the parameters configuration, a possible future development to improve the quality of the resulting mixes would be to explore the computation of the suppression thresholds which we defined by means of a percentile computation (and thus relatively expandable). In a practical scenario, it's very challenging to choose the right configuration for a complex set of parameters across different pieces of music, such as: number of sinusoids, percentile, suppression factor, temporal averaging. Thus probably a more complex test is needed for each of these parameterizations.

Even though, the theoretic model of this thesis could already be used in some further practical applications. An interesting idea for example would be to apply the suppression methods not only to one of the input tracks, but on both signals, which would definitely lead to a highly modified version of the input signals consisting of a sequence of short (spectrally) consonance-optimized audio pieces. A more compositional approach on the other hand would be to explore the time masking methods, and understand how the resulted mix affects the rhythm structure of the combined mix. Another question then would be, if this sequence would still represent a consonant melody, since the temporal consideration of dissonance was not part of this work. A more complex possible development would be to bring the user into the loop and to come out of the concept of automatic mixing of music and develop an interactive music mixing system. By following this idea, the user would be able to listen to each possibility and decide which of the implemented suppression method he/she wants to apply (temporal, frequency, harmonic). Furthermore, a significant improvement would be to bring also into real-time computation the amount of dissonant content that the system can suppress. This would allow the user to play with the dissonance/consonance of the sounding mix as much as a real time music descriptor, and this could be even parameterized into a hardware implementation as much as we could do with a complex filter. The user would then be able to try to iteratively adapt parameters according to his/her personal taste.

In summary, the results of this thesis pointed to the assumption that if we remove the dissonance generated in the mixing process, the two pieces of music will improve the way they can match with each other. Dissonance however, appears to represent a complex phenomenon, which is not easy both to measure and identify. As a consequence, further work is required to completely understand its impact in particularly sounding mixes, and to define how to implement a more precise dissonance-deleting algorithm. However, the suppression of dissonance has shown itself to be a promising tool for enhanced content based mixing of music.

Appendix
Lists of music titles

We present now all the songs from a selection of recent electronic music which has been used to test the performance of the dissonance suppression approaches.

Track No.	Artist	Title
1a	Person of interest	Plotting with a double deuce
1b	R-A-G	Black rain (Analouge mix)
2a	KWC 92	Night drive
2b	Stephen Lopkin	The Haggis Trap
3a	Anton Pieete	Waiting
3b	Legowelt	Elementz of houz music
4a	Orson Weels	Leaving
4b	Donato Dozzy & Tin Man	Test 7
5a	Aroy Dee	Blossom
5b	Barnt	Under his own name but also sir
6a	Massimiliano Pagliari	JP4-808-P5-106-DEP5
6b	Lauer	Highdimes
7a	Locked groove	Dream within a dream
7b	Luke Hess	Break Through
8a	Roman Flügel	Wilkie
8b	Liit	Islando
9a	Tin Man	No new violence
9b	Julius Steinhoff	The cloud song
10a	Levon Vincent	The beginning
10b	Voiski	Wax fashion

Bibliography

[1] T. Hirai, H. Doi and S. Morishima, “MusicMixer: Computer-Aided DJ System based on an Automatic Song Mixing,” *ACE '15,* 2015.

[2] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez, F. Gouyon, P. Herrera, S. Jordà, O. Paytuvi, G. Peeters, J. Schülter, H. Vinet and G. Widmer, “Roadmap for Music Information ReSearch Creative Commons BY-NC-ND license,” 2013.

[3] Benward and Saker, Music: In Theory and Practice, Seventh ed., vol. I, Mc Graw-Hill, 2003, p. 178.

[4] M. E. P. Davies, P. Hamel, K. Yoshii and M. Goto, “AutoMashUpper: Automatic Creation of Multi-Song Music Mashups,” *IEEE/ACM Transaction on audio speech, and languageprocessing,* vol. 22, no. 12, pp. 1726 - 1737, 2014.

[5] R. Gebhardt, M. E. P. Davies and B. Seeber, “Harmoni mixing based on roughness and pitch commonality,” *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15),* pp. 185-192, 2015.

[6] X. Serra and J. Smith, “Spectral Modeling Synthesis: a Sound Analysis/Synthesis Based on a Deterministic Plus Stochastic Decomposition,” *Computer Music Journal,* vol. 14, pp. 12-24, 1990.

[7] B. Brewster and F. Broughton, Last Night a DJ Save My Night: The History of

the Disc Jockey, Second ed., London: Headline, 2006, pp. 54-82.

[8] S. Webber, DJ Skills: The essential guide to Mixing and Scratching, Elsevier, 2008, pp. 46-73.

[9] J. Driedger, M. Müller and S. Ewert, Improving time-scale modification of music signal using harmonic-percussive separation., vol. 21, IEEE Signal Processing Letters, 2014, pp. 105-109.

[10] W. Verhelst and M. Roeland, An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech, Minneapolis, MN: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1993, pp. 27-30.

[11] J. Flanagan and R. Golden, Phase Vocoder, vol. 45, Bell Syst. Tech., 1966, pp. 1493-1509.

[12] J. Laroche and M. Dolson, Improved phase vocoder time-scale modification of audio, vol. 7, IEEE Transactions on Speech and Audio Processing , 1999 , pp. 323-332.

[13] M. R. Portnoff, Implementation of the digital phase vocoder using the fast Fourier transform, vol. 24, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1976, pp. 243 - 248.

[14] A. Duncan and D. Rossum, Fundamentals of pitch shifting, P. 2714, Ed., Los Angeles: Proc. 85th Convention of the Audio Engineering Society, 1988.

[15] A. Haghparast, H. Penttinen and V. Välimäki, Real-time pitch-shifting of musical signals by a time- varying factor using normalized filtered correlation time- scale modification, Bordeaux: Proc. of the DAFx'07 - 10th International Conference on Digital Audio Effects, 2007, p. 7–14.

[16] N. Tokui, Massh! A Web-based collective music mashup system, Proc. 3rd Int. Conf. Digital Interactive Media Entertainment and Arts, 2008, p. 526–527.

[17] G. Griffin, Y. E. Kim and D. Turnbull, Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups, Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP), 2010, p. 437–440.

[18] H. Ishizaki, K. Hoashi and Y. Takishima, Full-automatic DJ mixing with optimal tempo adjustment based on measurement function of user discomfort, Proc. 10th Int. Soc. Music Inf. Retrieval Conf., 2009, p. 135–140.

[19] E. Chew, Towards a Mathematical Model of Tonality, Cambridge, MA: Doctoral Dissertation, Department of Operations Research, Massachusetts Institute of Technology,, 2000.

[20] O. Izmirli, "Template based key finding from audio," *ICMC,* p. 211–214, 2005.

[21] D. Huron and R. Parncutt, "An Improved Model of Tonality Perception Incorporating Pitch Salience and Echoic Memory," *Psychomusicology,* vol. 12, no. 2, pp. 154-171, 1993.

[22] E. Gómez and P. Herrera, "Estimating the Tonality of Polyphonic Audio Files: Cognitive versus Machine Learning Modelling Strategies," *Proceedings of the Fifth International Conference on Music Information Retrieval,* p. 92–95, 2004.

[23] C. Krumhansl, "Cognitive Foundations of Musical Pitch," *Oxford University Press,* 1990.

[24] C. Chuan and E. Chew, "Polyphonic Audio Key Finding Using the Spiral Array CEG Algorithm," *IEEE International Conference on Multimedia and Expo, ICME,* pp. 21-24, 2005.

[25] R. Parncutt, Harmony: A psychoacoustical approach, Berlin: Springer, 1989.

[26] W. Apel, The Harvard Dictionary of Music, Second ed., Cambridge: Harvard Universitary Press., 1970.

[27] H. Jensen, "A theoretical work of late seventeenth century Muscovy: Nikolai Diletskii's "Grammatika" and the earliest circle of fifths," *Journal of the*

American musicological society, vol. 45, no. 2, pp. 305-331, 1992.

[28] S. Pauws, "Musical Key Extraction from Audio," *Proceedings of the Fifth International Conference on Music Information Retrieval,* pp. 96-99, 2004.

[29] N. Hu, R. B. Danneberg and G. Tzanetakis, Polyphonic Audio Matching and Alignment for Music Retrieval, New Paltz, NY: Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003, pp. 185 - 188.

[30] A. Sheh and D. P. W. Ellis, Chord Segmentation and Recognition using EM-Trained Hidden Markov Models, Baltimore, Maryland: Proceedings of the International Conference on Music Information Retrieval, 2003, pp. 183-189.

[31] T. Fujishima, Realtime chord recognition of musical sound: a system using common lisp music, Proc. of ICMC, 1999, p. 464–467.

[32] K. Lee, Automatic Chord Recognition Using Enhanced Pitch Class Profile, New Orleans: Proceedings of International Computer Music Conference, 2006.

[33] J. Brown, Calculation of a constant q spectral transform, vol. 89, Journal of the Acoustical Society of America, 1991, p. 425–434.

[34] T. Jehan, "Creating Music by Listening," PhD thesis, Massachusetts Inst. Technology, Cambridge, 2005.

[35] E. Terhardt, The concept of musical consonance: A link between music and psychoacoustics, vol. 1, Music Perception: An Interdisciplinary Journal, 1984, p. 276–295.

[36] W. Hutchinson and L. Knopoff, The acoustic component of western consonance, vol. 7, Interface, 1978, p. 1–29.

[37] R. Plompt and W. J. M. Levelt, Tonal consonance and critical bandwidth, vol. 38, Journal of the Acoustical Society of America, 1965, p. 548–560.

[38] R. Parncutt, Parncutt's implementation of Hutchinson & Knopoff (1978), Available at http://uni-graz.at/ parncutt/rough1doc.html.

[39] R. Parncutt and H. Strausburger, Applying psychoacoustics in composition: "harmonic" progressions of "non-harmonic" sonorities, vol. 32, Journal of the Acoustical Society of America, 1994, p. 1–42.

[40] L. Hoffman-Engl, Virtual pitch and pitch salience in contemporary composing, Rio de Janeiro: Proceedings of VI Brazilian Symposium on Computer Music, 1999.

[41] R. McAulay and T. Quatieri, "Speech Analysis/Synthesis based on Sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 34, no. 4, pp. 744-754, 1986.

[42] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," *Proceedings of the 3rd International Conference on Music Information Retrieval,* p. pp. 150–156, October 2002.

[43] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing,* vol. 2, p. 765–768, May 2000.

[44] H. Ye and S. Young, "High quality voice morphing," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 1, p. 9–12, May 2004.

[45] B. B. Hubbard, The world according to wavelets, Wellesley, Massachusetts: AK Peters, 1996, p. 76.

[46] M. Müller, Fundamentals of Music Processing, Springer, 2015, pp. 53-57.

[47] A. Kanagasundaram, D. Dean and S. Sridharan, Jfa based speaker recognition using delta-phase and mfcc features, SST 2012 14th Australasian International Conference on Speech Science and Technology, 2012.

[48] F. Harris, On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, vol. 66, Proc. IEEE, 1978, pp. 51 - 83.

[49] A. Klapuri and M. Davy, Introduction to Music Transcription, vol. 1, Springer, Ed., New York, NY: Signal Processing Methods for Music Transcription, 2006, p. 8.

[50] W. M. Hartmann, Signals, Sound, and Sensation, Springer, 1997, pp. 145, 284, 287.

[51] W. A. Yost, Attention, Perception and Psychophysics, vol. 71, 2009, pp. 1701-1715.

[52] J. G. Roederer, The Physics and Psychophysics of Music, Springer, 1995.

[53] Helmholtz, On The Sensation Of a Tone, Dover, 1863.

[54] B. Moore and B. Glassberg, “Suggested formulae for calculating auditory-filter bandwidths and excitation patterns,” *Journal of the Acoustical Society of America,* vol. 74, p. 750–753, 1983.