Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Real-Time Multi-Track Mixing For Live Performance

Joaquín Jiménez-Sauma

Supervisor: Frederic Font

August 2018

Universitat Pompeu Fabra Barcelona

Table of Contents

Dedication

(Optional, if used placed on a right page next to an empty left page)

Acknowledgments

(Optional, if used placed on a right page next to an empty left page)

1. Abstract

Finding the balance of sounds in a multitrack recording is always a time-consuming process performed by experienced professionals. A poor mix makes it difficult to let the sounds stand out or enhance their presence, also creating a recording where clarity is not perceived. Auditory Masking of tracks is a common problem that affects the presence of instruments in the mix, making some elements indistinguishable and not audible.

This thesis analyses previous research in the field of automatic mixing and investigates methods to avoid Auditory Masking in the multitrack performance of music. A set of tools are developed during this process to help reduce Auditory Masking by implementing different state-of-the-art techniques as well as a standard measurement of Auditory Masking. Unlike similar approaches in state of the art, we designed the unmasking tools in this thesis to work in real-time.

The intention of previous research in the field of automatic mixing was to improve mixing for studio recordings. This thesis aims to apply this knowledge to the case of real-time performance where different considerations apply.

We developed three different real-time versions of unmasking tools between two channels as a result of this research. Each version implements different techniques with similar objectives but different advantages. These tools we perform both quantitative and qualitative evaluations to find which one provides the best results.

2. Introduction

2. 1 Research Question

Can we create a reliable system to help a live performer of real-time computer music reduce the amount of auditory masking in a multitrack mix in order to focus on the creative aspect of his performance? What are the obstacles, disadvantages or benefits of trying to create a definitive tool for auditory unmasking of real-time audio?

2.2 Motivation

Auditory Masking in music is a common problem in real-time performance where a performer needs the assistance of a live mixing engineer to help the performer keep the clarity and presence of all sounds involved in the musical performance.

In the field of electronic musicians who perform live, the set up is in most cases, composed of small, or portable hardware and a computer. This configuration or resources available do not always allow the inclusion of an audio engineer in the performance. Mixing of multitrack music is also an artistic process. Some artists prefer to keep their mixing decisions saved as part of their tool and use them every time they perform.

We need to separate the technical side of the performance setup from the artistic and creative tasks. There are great opportunities and challenges to create production tools that help the performer concentrate on their creativity. In this thesis, in particular, we propose to explore the creation of mixing tools that help the performer unmask, and in turn, gain, clarity in the mix of sounds.

The areas of Intelligent Music Production and Intelligent Mixing are entirely new, and even with current achievements on this area of research along with many proposed solutions for mixing a multitrack recording have been proposed for studio sessions, the case for a real-time tool is not clear yet. No definitive tools or commercial apps are available for real-time performers at the time of writing this thesis.

The importance of a great mix in real-time performance lies in the fact that it is one of the last steps in the audio processing chain. Mastering is the last step in the chain, but real-time mastering is a complex task, and in the case of an automated tool, it requires a high amount of knowledge, training, prediction and intelligence.

2.3 Objectives

In the process of implementing a real-time mixing tool we are doing a study of previous research on this subject and I will use this results as a starting point for the development of a reliable tool that helps unmask one or more channels in a multitrack mix.

The results will be evaluated considering its relevance in the particular case of a real-time tool, and will be implemented as Max for Live patches. This tools focuses on electronic musicians playing mostly electronic music using Ableton Live.

The particular objectives during this research will be oriented on processing at a considerable amount of CPU time (to avoid limiting the performer in the use of his tools), a reduced amount of latency, and ease of use. The latter follows the idea of having a tool that lets the performer focus on creativity more than technical aspects of audio, this tool needs to present a reduced number of controls to stop the performer to worry about the settings of the unmasking tools.

2.4 Structure Of This Thesis

Introduction
Presentation of the research question, motivation, scope and objectives of this thesis.

Essential Theory
This chapter goes through the important theory that is the basis for the implementation.

State Of The Art
This section describes previous research and shows the relevant works used as the basis of this thesis, describing the reasons to use each one and the way it will contribute to the tools developed in this work.

Proposed Tool

We discuss definitions regarding the challenges and strategies for the development of this tool. We explain the differences between versions and research used in each case. We also present restrictions in its use and an introduction to the development tool.

Implementation

We give a walkthrough of the most critical aspects of the development and strategies used to achieve the objectives of this tool.

Using the tool

We show what the tool does and how to interpret the results returned by their processes.

Evaluation

In this chapter, we present the methodology to evaluate the results of using the three tools developed during this research.

Conclusions and Future Work

This chapter sums up the work and experiments done during this research and evaluates the results. Also, comment on the work that we can do to improve this research, directions it may take and possible spin-offs.

3 Essential Theory

3.1 Multitrack Mixing

Multitrack Mixing occurs after the recording of separate channels of a musical piece, typically one instrument per channel. Mixing engineers perform this process on a mixing desk or Digital Audio Workstation (DAW) inside a computer. It aims to balance the sounds in a recording, representing an optimisation problem where there is not a single right answer, and many goals may be the purpose or end state. The goals range from adapting the recording for a specific audience, performance location, live performance, or to bring emotional impact. Mixing can also be a performance act, as the mixing manipulation during tape concerts, electroacoustic concerts or dub mixing.

Mixing engineers know how to perform the mix of a multitrack recording, they know how to create great mixes, but they might not know how to explain them [19], this means, there is not a strict set of rules to follow when creating the mix of a multitrack recording.

Typically, the multitrack signal goes into a series of audio processes like gain, panning, equalisation, dynamic range and time-domain processing. This process may not include all the steps mentioned, neither in the same order, it all depends on the mixing engineer decisions [22].

3.2 Artistic Mixing vs. Technical Mixing

A mixing engineer can create a mental mix of the recording just by listening to the raw multitrack thanks to his experience. This mix is the engineer’s vision, and he will likely change it thanks to the input from the producer or artist [20]. The same engineer can create a mix that makes all instruments sound clear, balanced and separately without disappearing due to auditory masking. Some mixing engineers think of this task as performance once that they achieve some basic balance settings [21]. This way of thinking is what separates the artistic from the technical mixing.

Basic balance settings mean a state of the mix when the engineer can start adding character or imprinting characteristics of a specific goal in the final mix.

This thesis will deal with the technical aspect of mixing, creating tools that can be used as an assistant in the process of mixing, allowing the performer to focus on the creative aspect of mixing after achieving some basic balance settings.

3.3 Frequency Masking

Frequency Masking is a phenomenon in which the perceived audibility of one sound is affected by the presence of another sound which has a similar spectral content [3]. The sounds in a mix compete with each other in multitrack recordings. The result of masking is a lack of clarity.

Figure 1 shows an example of a masker and masked sound. The masker can mask many sounds at once.

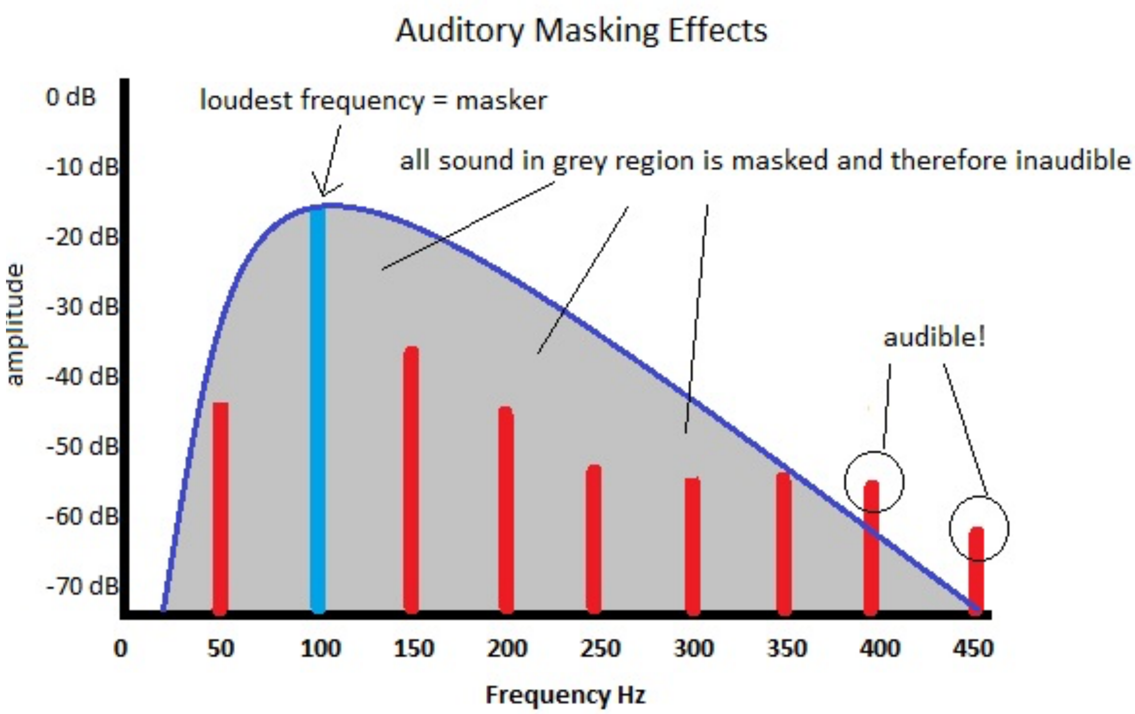

Figure 1. Auditory Masking in Frequency Domain. *(Image retrieved from www.audioholics.com)*

Studies about the effect of auditory masking propose to quantify this phenomenon. In this thesis, we are going to use the metric proposed by Aichinger and Sontacchi [7], called the Masked-to-Unmasked Ratio (MUR). We present the formula to compute this metric is presented in equation 3 in section 4.2.

3.4 Real-Time Performance On A Computer

We understand the concept of real-time performance on a Computer as a single performer using software and hardware to generate sounds while manipulating timbres using audio mixers, effects, and MIDI controllers. This setup allows performers to assume a role that borrows practices from conductors, mixing engineers, DJs, and instrumentalists [24]. The concept of live-sequencing exists and has been explored since the 1940s and constitutes the norm amongst live electronic music performers.

For this performance to be strictly considered Real-Time, it should guarantee a response in defined time constraints and without perceivable delay. During the performance, a continuous process outputs a signal, and the time needed to process each sample must be shorter than the sampling period.

3.5 Basic Equaliser Theory

The full description of an equaliser explains its functions in the sense of a device that allows to emphasise or attenuate the volume of specified frequencies. During the mix, we use equalisation in different ways to correct problems that were created during the recording session or from incompatibility among instruments. On the creative side, we use equalisation in order to produce original effects [26].

The parametric equaliser is the most powerful and flexible of the equaliser types. It allows the mixing engineer or operator to emphasise or attenuate an arbitrary location in the audio spectrum.

An equaliser of a one-third octave is a conventional design. This setup means each octave of the spectrum contains three bands, for example, starting at 1000 Hz, the following frequencies to be attenuated or emphasised will be a series starting from 1260 Hz, 1587 Hz, 2000 Hz, until the end of the spectrum. [27]. Frequency bands spacing determines the number of bands and the requirement to cover the entire audible spectrum. Octave graphic equalisers, for example, usually have ten bands, ranging from about 31 Hz at the lowest to 16 kHz at the highest. Third-octave designs usually have 31 bands ranging from 25 Hz to 20 kHz. The ISO standardises these frequencies, and it is documented in [13]. This type of equaliser is often found in hardware units because the gain sliders are easy to control. Rarely this kind of equalisers is used on individual channels for mixing because those are far too inaccurate for a typical mix where the settings remain static during the whole recording.

Note that individual bands in an equaliser will not only amplify the range for the frequency it is labelled, for example, 100 Hz - 1kHz. It will affect the frequencies around it as well.

Using an equaliser to unmask a signal with a fixed setting will decrease the dynamic impact of the masker, especially in cases where the masker and masked interact in a strong rhythmic way. In this case, a form of a dynamic equaliser will allow the masker to retain its full tonal balance when the masked is absent; this is where modern equalisers implemented as plugins or computational tools are essential to avoid killing musicality in the recording.

3.6 Spectral Processing and FFT Equalisers

Spectral processing or sound processing in the frequency domain is a fundamental technique to achieve sound synthesis, transformations and processing in Real-Time situations. Real-Time spectral processing is relatively new in the world of music computing. Languages like Max/MSP or Pure Data have made it possible along with new developments in computer processing. Artists, composers and researchers use these tools widely to process in Real-Time. Both enable work in the spectral domain via FFT analysis/re-synthesis [28].

Audio signals are composed of a multitude of different sound components; in order to process the audio of a given signal, we use and analyze building blocks to compose it. In the case that these building blocks consist of complex-valued sinusoidal functions, such a process is also called Fourier analysis. The Fourier transform maps a time-dependent signal to a frequency-dependent function which reveals the spectrum of frequency components that compose the original signal. Short-Time Fourier Transform (STFT) allows us to compute the frequency-based representation of audio by determining the sinusoidal magnitude and phase content of local sections of a signal as it changes over time. The STFT shows which frequencies are “contained” in the signal, also at which points in time these frequencies appear. We can efficiently compute the discrete Fourier transform (DFT) using the fast Fourier transform (FFT) to yield a discrete set of Fourier coefficients that are indexed by time and frequency parameters and is used to compute signals that are not perfectly periodic. The correct physical interpretation of these parameters in terms of units such as seconds and Hertz depends on the sampling rate, the window size, and the hop size used in the STFT computation.

Figure 2 shows a brief example of a signal transformed from time-domain to frequency-domain by taking small parts of the time signal and processing each one to compute the frequency information per window. During each window, it is possible to compute values and descriptors in the frequency domain. Also, to modify this information to reconstruct a new version of the signal.

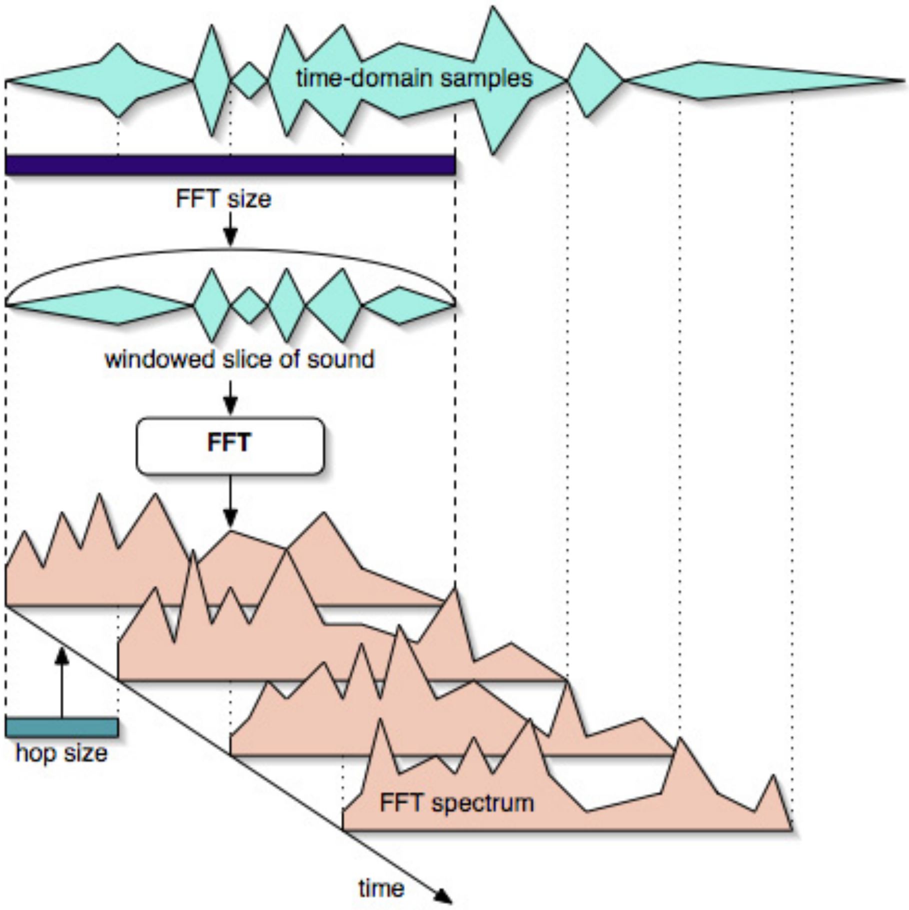

Figure 2. Diagram of the Short Term Fourier Transform (STFT). *(Image retrieved from https://cycling74.com/2006/11/02/the-phase-vocoder-—part-i/)*

The STFT of a signal consists of the Fourier transform of overlapping windowed blocks of the signal. Windowed means applying a function of each block of information so that only a specific section of the signal is nonzero. The shape of the window turns out to be necessary because the windowing produces a result where each bin of the transform (each component in the Fourier series), includes some energy from other bins nearby. The window function has the same result as a filter impulse response in frequency bins.

The FFT size determines the size, in audio samples, of the overlapped windows that the transform converts to and from the spectral domain. The window size must be a power of 2. The hop size (number of samples between each successive FFT window) is equal to the FFT size divided by an overlap factor (e.g. if the frame size is 512 and the overlap is 4, then the hop size is 128 samples). The value must be a power of 2 and defaults to 2. The recommended value is 4 for most applications.

The FFT is an algorithm for computing the Fourier Transform with fewer multiplications than would otherwise be required provided that the number of bins is a power of 2.

In this thesis, we will use the process of computing the frequency domain of a signal with STFT to create an FFT Equaliser. This device has no mechanical or electronic counterpart. It can be viewed as a graphic equaliser with thousands of bands or only as equalisation by drawing curves. In any case, it provides precise control of frequency response. There is an essential difference between FFT and more traditional equalisers: FFT curves are often shown with equal frequency spacing across the window, whereas graphic equalisers have equal octave spacing [31]. For lower frequencies, the FFT equaliser presents a characteristic called "poor bass resolution". One way to compensate for this is to use a large FFT Size.

Graphic equalisers work in the time domain and FFT equalisers in the frequency domain.

This thesis will use spectral processing with Max/MSP because it performs all the computation and transformations to relate time domain continuous musical signals into the frequency domain, allowing for Real-Time windowing, manipulation of partials, frequency bins, amplitude in a multithreading environment inside a DAW, in this case, Ableton Live.

3.7 Adaptive Digital Audio Effects

The implementation of the tools in this thesis follows the definition of Adaptive Digital Audio Effects (A-DAFX). These audio processes are effects driven by parameters or features that the effect extracts from the sound itself [23]. The principle of this effect's class is to provide a changing control that adapts to dynamic signals as opposed to DAFx class, which is static.

A-DAFX sound processors combine the theory of sound transformation and adaptive control. This kind of process is fundamental for real-time mixing and equalisation because the tools in this thesis apply different levels and configurations of effects based on the dynamic characteristics of the input signals. The interest in the development of tools like this is to provide possibilities for re-interpreting musical sentences, changing timbres or presence of musical events with, strong, immediate or fine changes in the properties of the sounds [23].

3.8 Multithread Programming

In multithread programming, a system runs as a collection of concurrent tasks. Using concurrent tasks rather than using a single task for all has advantages, like support for separation of tasks where the programmer designs, implements, tests, and verifies each one as a separate entity. Another advantage is having concurrent tasks, where many processes run at the same time instead of a single sequential task [29].

Music and digital audio are a parallel structure, creating a set of non-communicating streams of tasks called voices, lines, or tracks. The designers of most DAWs create their tools around this concept. Examples include the way Ableton Live can use multiple cores and the multi-threading option in Max/MSP poly~ abstraction [30].

This thesis will use multithread programming to implement audio processing tasks like Short Time Fourier Transform in Real-Time using Max/MSP and Ableton Live.

3.9 Analysis Synthesis Techniques

In recent years several Digital Signal Processing algorithms have been developed to take an input signal and produce an output signal that is either identical to the input or a modified version of it. FFT is the basis of these implementations, where the programmers assume that this process can represent the input signal as a mathematical formula with time-varying parameters, and the synthesis is simply the output of the model. The benefits of these techniques rely on the derived values for this analysis. The parameters can be modified to change the perceptual significance and musical utility of the result.

The FFT used for these techniques models the input as a sum of sine waves and the parameters to be determined by analysis are the time-varying amplitude and frequency for each sine wave. These sine waves are not required to be harmonically related, so this model is appropriate for a wide variety of musical signals. While this method works well for harmonic sounds, a model that sums a series of sine waves cannot represent very well other percussive sounds (e.g., clicks) and certain signal-plus-noise sound combinations. The process can still synthesise these sounds, but attempting to modify them can produce unexpected results [32].

This thesis will use Analysis Synthesis Techniques to decompose audio signals, modify the components and produce a new output adapted to the goals of the mixing process. We detail these processes in section 6.

4 State Of The Art

4.1 Intelligent Music Production

Music production has been democratised in recent years, allowing anyone who has access to a computer to produce professional results. Musicians, singers and producers can perform the whole production process “in the box”. However, this process needs professional audio engineering knowledge during these tasks. Access to professional tools at home, bedroom or small studio does not guarantee professional results [1].

4.1.1 Intelligent Music Production vs. Human Engineers

Many products exist in the market today that employ forms of intelligence used exclusively by humans in the past to solve actual music production problems. Besides some rare exceptions, most of these products do not try to fully automate sound engineering tasks or replace humans involved in the music-making process. These products take advantage of computational power and precision to make the process of music-making faster, fun and forward-thinking. Human skills like passion, inspiration, communication are no competition for the perfect algorithm [2].

A human mixing engineer creating a mix would need to have control of multiple fader gains simultaneously. When masking is present, the mixing process becomes an iterative optimisation problem. With an automated process, the control of the fader gains, or interaction with the unmasking system offers advantages over the human process [25].

Amateur or even professional artists can take advantage of Intelligent Music Production tools to concentrate on the artistic and creative process creating more expressive performances thanks to advancements in technology. Exploring the power and flexibility of computers can make the producers expression venture away from the rigid structure of the traditional toolset of music production [2].

4.2 Automatic Mixing

Automatic mixing of multitrack music remains an unsolved problem that has not exhausted its research directions. Pioneers in this field believe these applications are still in its infancy [1].

The term "Automatic Mixing" appeared thanks to Dan Dougan, who invented a device that automatically reduces the strength of a microphone's while it is not receiving any signal. This device, the "automixer", was created for settings where no sound operator is present, in order to avoid feedback and noise in the signal. Dan Dugan showed his first "Adaptive Threshold Automatic Microphone Mixing System" in 1974 at the 49th Audio Engineering Society (AES) [14].

Between 2007 and 2010, Enrique Perez-Gonzalez proposed automatic methods to adjust level, stereo panning, unmasking and delay correction, giving birth to the field of Automatic Mixing [1].

Further research has generated different proposals for automatic multitrack music mixing. The solutions proposed fall into these categories: Adjusting levels, panning, EQ, Compression and reverb. It is important to mention that not all proposals focus on real-time mixing.

In the market, some tools are available for this task like Neutron by Izotope, Vocal Rider by Waves, and Smart EQ by Sonible. From these tools, only the latter performs real-time processing after previous offline training.

4.2.1 Audio Engineering Best Practices For Unmasking

The research results in [15] compare three of the most commonly used techniques used by mixing experts to unmask sounds. Mirrored Equalisation, Frequency Spectrum Sharing, and Stereo Panning.

- **Mirrored Equalisation**: To unmask one channel in a specific frequency, mixing engineers boost the corresponding frequency on the masked signal and at the same time, attenuate the same frequency on the masker. The research by [5] suggests that this approach may not work because it is more reliable to attenuate the masked frequency regions instead of boosting the unmasked frequency regions. Subjective tests show little preference for this solution.

- **Frequency Spectrum Sharing**: This approach uses the spectral centroid calculation for each track and uses this information to determine which track is high pass filtered and which is low pass filtered. Results show that listeners in the tests were indifferent of the application of this unmasking method.
- **Stereo Panning**: Adjusting the pan position to reduce frequency masking proved to be the best method for unmasking. However, this solution may be unpractical because it may be against artistic decisions; for example, we may not want to pan the kick drum or bass to one side of the recording.
- **Sidechaining**: This technique compensates the amplitudes of masker and masked signals by reducing the overall amplitude of the masker signal every time the masked reaches a certain threshold. This approach produces a highly dynamic manipulation of the masker signal that depends on the behaviour of the masked signal.

Many applications for automatic mixing exist now, from autonomous to assistants and workflow enhancing tools. This solutions vary and have different degrees of user control. We can classify these solutions as real-time and offline.

4.2 Previous Research

This section will refer and comment specific research made in the previous years that developed knowledge used as a starting point in this thesis. The research publications in this section deal with the unmasking of audio signals. This list will also help us to compare the solutions against each other, in order to find the ones that are more suitable for the case of Real-Time processing.

Perez-Gonzalez & Reiss 2008: Improved Control for Selective Minimisation of Masking Using Inter-channel Dependency Effects [16].

The author develops a tool that permits the enhancement of a source in respect to the rest of the mix by selectivity unmasking its spectral content from spectrally related channels. It proposes one of the first documented masking metrics, the Spectral Masking (SM), computes the amount of overlap between the source and the rest of the mix. We compute this measure using the formula:

$$SM = (FFT(Ch_m))^2 - (FFT(mix - Ch_m))^2 \tag{1}$$

Where Ch_m represents the channel m, the one we want to measure masking against signal *mix*, which represent the sum of all channels. This metric depends on the size of the FFT used, and the author recommends not to use windowing. When $SM > 0$, the channel Ch_m is unmasked, otherwise is masked by the rest of the mix [16].

This research focuses on volume adjustments of channels, and it implements cross-adaptive channel enhancement to perform a selective minimisation to intensify a user-selected channel by ensuring it is spectrally unmasked from the rest of the mix.

Perez-Gonzalez & Reiss 2008: These authors developed automatic equalisation of multi-channel audio using cross-adaptive methods [17]. This paper is part of the series of papers published during the beginning of research in automatic mixing. The difference with the previous one lies in the fact that it proposes a method to apply equalisation of channels using five-band equalisers. Tests for this approach report inaccurate results on low frequencies and future work suggest the use of more equalisation bands.

Vega & Janer 2010: Quantifying Masking in Multi-Track Recordings [8]. In this research, the authors propose the Masking Coefficient or MC, a measurement of masking amount between two or more channels. This measure is a number between 0 and 1 that corresponds to the amount of effective excitation overlap between two sounds. The following equation expresses the masking coefficient between two excitations:

$$MC = \frac{|\, e_1(t,b) - e_2(t,b)\,|}{60dB} \tag{2}$$

Where e_1 and e_2 are the matrices containing the excitograms of each sound in decibels.

The process of computing the *MC*, involves spectral processing and this may be a slow process that causes latency and it’s not intended for real-time tools.

Aichinger, et al. 2011: Describing the transparency of mixdowns: The Masked-to-Unmasked-Ratio (MUR) [7].

This masking measurement focuses on loudness and was tested perceptually through listening tests. It is computed as shown equation 3.

$$MUR = \frac{max(N_{masked}, 0.003)}{max(N_{unmasked}, 0.003)} \tag{3}$$

Where N_{masked} represents the overall loudness of a signal when the masker is present, and $N_{unmasked}$ describes the overall total loudness of the same signal, when the masker is imaginary assumed not to be present. This metric is independent of frequency.

This metric is the most used and cited across most automatic mixing research reports, including the ones mentioned in this section.

Hafezi & Reiss 2015: Autonomous Multitrack Equalisation Based on Masking Reduction: In this research, the authors design a simplified measure of masking based on test practices in sound engineering. They implement offline and real-time versions of unmasking models [5].

Offline versions depend on history and tendencies of the mixed tracks. The equalisation setting remains constant over time and uses FFT to obtain the magnitude of each frequency region in a track.

In the real-time version of this unmasking tool, the authors implemented an ISO standard octave-band (10 bands) processing on a frame-by-frame basis. Each filter computes the RMS of the amplitude of the signal, which represents the magnitude of that band.

One of the most important conclusions of this research is that, to improve performance, we may need to incorporate additional knowledge from psychoacoustics.

Robert Koria 2016: Real-Time Adaptive Audio Mixing System Using Inter-Spectral Dependencies [10].

This research proposes a method for real-time unmasking. The author describes and implements a model using Matlab, but just for test purposes. This implementation does not run in real-time and uses adaptive multi-band compressors that analyse the spectral information between tracks.

To simplify the decision on which channel is the masked one, the author proposes a hierarchical cascade system where the sequence of the channels dictates the priority of attenuations applied (Figure 3) and is the user who defines this sequence. The second channel should be unmasked (or attenuated) from the first, and then, the second channel unmasks the third, and so on. This system helps to reduce the number of calculations during the process, making it more suitable for real-time situations. Results from this research show that attenuations occur mostly below 400 Hz.

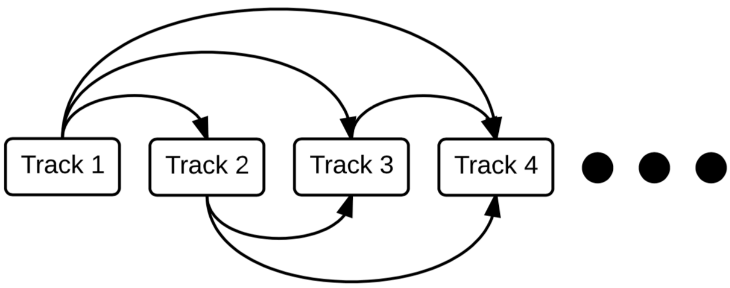

Figure 3. Track attenuation dependencies in a hierarchical cascade system [10].

De Man & Reiss 2017: Ten years of automatic mixing [1]. A literature review about the first ten years of automatic mixing. It sums up the most important milestones and state of the art in this field. Cites the most important publications and research work during the period 2007-2017 and hints that Machine Learning will be the best approach for this kind of tasks.

In regards to this particular research, it does not make a distinction between real-time or offline automatic mixing.

Ronan et al. 2018: Automatic Minimisation of Masking in Multitrack Audio using Subgroups [12]. Using subgroups is a technique employed by audio engineers, mostly the ones using digital tools because it saves processing power and simplifies the task of processing sessions with a large number of channels. This research tests whether or not using subgroups is beneficial or not to automatic mixing systems.

The solution proposed in this research uses a compressor with static settings for dynamic range during the whole duration of the entire track. It also incorporated knowledge from psychoacoustics by implementing a masking threshold as a function of frequency, following the MPEG psychoacoustic model.

The results were successful in reducing auditory masking between channels, but when compared against human mixes, the latter proved to create a better perception and clarity of the sounds involved in the mix.

5 Proposed Tool

In this thesis, we will use the research from Chapter 4 to implement tools to help unmask the mix of two channels. Each implementation will use a different technique and will only unmask two channels (masker and masked). In further work, we can extend these tools to unmask a channel from mixes involving more channels. These tools will be compared in Chapter 8 to decide the approach that produces better perception od unmasking between two channels.

The tools developed in this research do not intend to minimise the work of a professional audio engineer. As mentioned in section 4.1.1, this tool will work as an assistant to the audio engineer and will allow him to focus on the creative part of the mixing process. Also, this thesis is looking to develop tools for the live performer of electronic music on a computer, given this approach, this tool must be straightforward to use and with an interface that avoids complexity in the performance. We will minimise the number of controls to let the user focus on artistic performance.

5.1 The Challenges Of A Real-Time Mixing Tool

The real-time tools developed in this thesis perform dynamic processes on the input signals, and cannot use algorithmic or iterative methods to unmask. Also, it will not be able to process time domain masking.

Many research efforts on the field of mixing multitrack recordings use Machine Learning, Deep Learning or Artificial Intelligence techniques to achieve the goal of a balanced and unmasked mix. This processes typically pre-process the whole recording or a representative segment of it. Others, like the ones proposed in [16] and [17] perform adjustments after listening to the recording for a considerable amount of time. These solutions are then, oriented to studio work.

In real-time tools, the time allowed to make decisions is limited, and depends on the solution or tools used, but has to remain almost instantaneous and avoid producing high latency. The research on [32] and [33] has studied the amount of latency allowed for musicians. However, they do not report information regarding electronic music performance, where the performer is interacting with a computer system with possible delay compensation from the system, buffering and pre-recorded segments. The amount of latency depends on the performance but becomes critical if it contains live input from physical, analogue instruments or microphones. So far, this thesis will apply for performances where the computer contains all elements inside.

Latency is also influenced by the processing power of the computer, in a way that the CPU of the computer hard limits the processing power allowed for performance, and while the process of unmasking the signal executes, it needs to permit the calculations needed for the sound generating elements, audio processes, MIDI and operating system tasks.

5.2 Platform Used

Max For Live running inside Ableton Live is the platform that we are going to use for this research and development. It provides the tools to analyse Real-Time information about this channel, implement band-pass equalisers and STFT, as well as to create and design the interface that will allow the performer to interact with it.

Ableton Live is a complete DAW that we use to mix, process and record live inputs like voice or instruments, to run virtual instruments and audio processes in Real-Time or as part of a live performance. These characteristics will allow running unmasking tools in a real-world scenario.

Version 8 of Max now enables us to implement the reception of all channels involved in just one patch, which makes it very practical to reduce the complexity and configuration of the tools implemented. Previous research had to develop a plugin for each channel in the mix.

5.3 Restrictions Of This Implementation

As mentioned before, and for testing purposes, the first version of the unmasking tools will process two signals, the masker and the masked. This configuration will give immediate results necessary to come with conclusions about its efficiency. We conduct and analyse listening tests in chapter 8.

It will be necessary to have the input of a performer or audio engineer to get professional results using these tools because this will only focus on unmasking, not in the whole mixing process. If we need some other processes like equalisation, change in amplitude, compression or distortion, it will leave that to the user.

6 Implementation

To test different findings in the process of this development, we created three different Intelligent Mixer devices (IM). These tools differ in the way we implement the unmasking process. The tools are named IM-WS (Whole Spectrum), IM-EQ (Equaliser) and IM-SEQ (Spectrum Equaliser).

All implementations receive two signals, the masker and the masked signal as input and produce a single output signal, the processed mix of given channels. Figure 4 describes the signal flow and processes involved in the unmasking of two signals used by all implementations of IM devices.

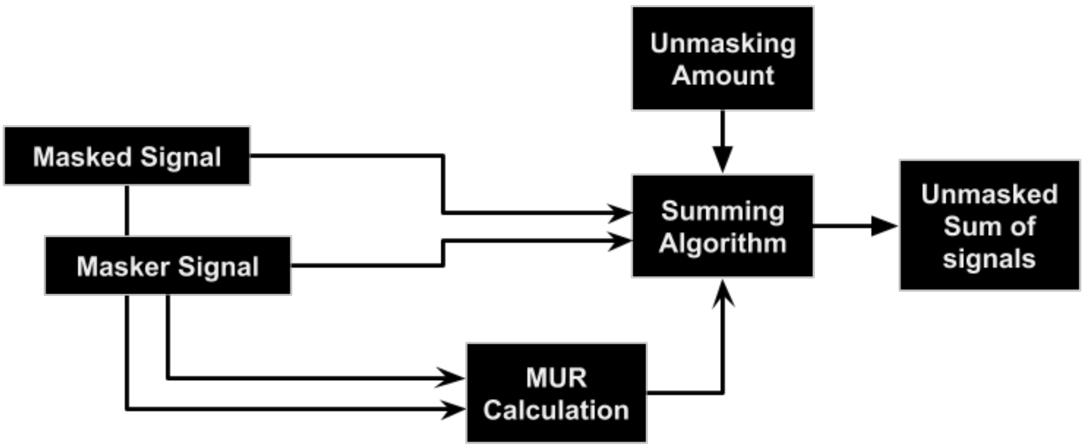

Figure 4. Block diagram of the unmasking process.

6.1 Implementation of IM-WS

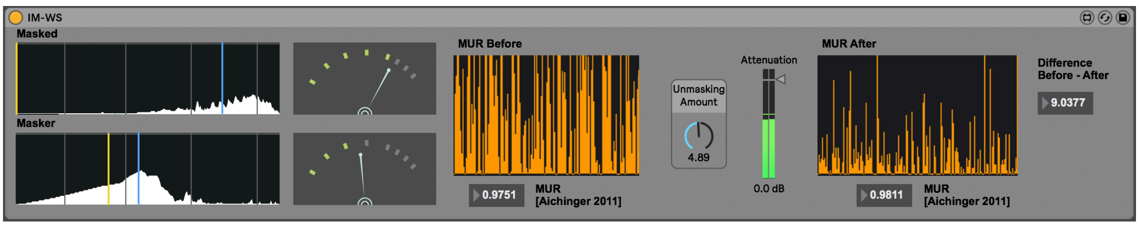

Figure 5. User interface of IM-WS.

This version of the unmasking tool computes MUR between masked and masked after short intervals of time. We adjusted these intervals at 23 milliseconds for the tests, but this value is subjective; we can alter it for CPU efficiency. At the moment, this parameter is not part of the user interface because it goes beyond the scope of this thesis. Figure 5 shows the interface of IM-WS, where the Attenuation slider shows the real-time attenuation of the masker signal.

The only parameter that needs user interaction is Unmasking Amount. This parameter will let the user decide how much the unmasking process should affect the output signal, and consequently, the mix of the channels involved.

This implementation performs unmasking by completing the method described in [16], where the authors apply full range magnitude adjustments instead of equalisation techniques. In this case, the masker channel attenuates in an amount given by the formula in equation 4.

$$A_t = UA_t * MUR_t \tag{4}$$

Where *A* is the amount of attenuation at time *t*, *UA* is the Unmasking Amount set by the user (in this implementation limited from 0 to 10) and *MUR* is the calculated ratio at time *t*.

This unmasking tool works similarly as a side chain compressor with fast attack and release, where the Unmasking Amount parameter gives the ratio.

6.2 Implementation of IM-EQ

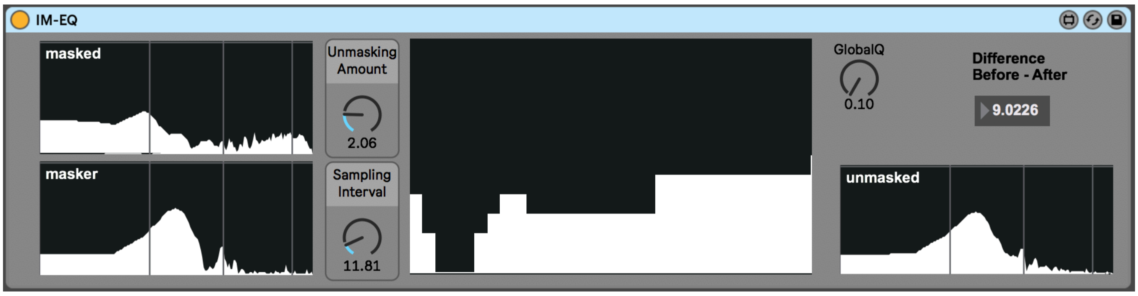

Figure 6. User interface of IM-EQ.

The research made by [17], uses a five-band equaliser since results were not quite as expected; it concludes that future work could use a higher number of equaliser bands. IM-EQ uses an ISO standard 31 band equaliser on the masker signal to create space in the spectrum that allows hearing the masked signal. The ISO 31 band equaliser [13], is a standard tool used widely in music production to shape the spectral envelope of an audio signal.

In this implementation, we analyse each signal, the masked and the masker at each of the 31 bands in separate processes, then MUR is computed per band to apply attenuation in the same way as IM-WS at each frequency band of the unmasking equaliser. This approach means 31 unmasking processes are running at the same time. Figure 6 shows the interface of IM-EQ, where the middle window shows 31 bars that represent the attenuation at each band of the masker signal.

Having 31 band independent processes impacts the percentage of CPU used for this task, and it is the equivalent to 31 times IM-WS. Also, this device does not work frame by frame, it reads the values of each band and then waits for the number of milliseconds specified in the Sample Interval control to read values at each band again. We use the time between readings to compute MUR at each band and to apply the corresponding attenuations. We could use this control to balance between accuracy and CPU usage. At longer interval times, less CPU usage, a small value of this parameter means more detailed unmasking.

This implementation may lack accuracy when reading, computing MUR and applying attenuations, because the system does not provide all filter values at the same time. Also computing MUR and applying attenuations may not happen at the same time on every filter. Another important fact is that at each time interval, the attenuation applied corresponds to the MUR computed in the previous interval, causing unexpected results for large values of Sampling Interval. During tests, short intervals (no more than 50 ms) have produced accurate results and proves the research made in [33] and [34]. Further research may be needed to find the maximum interval that produces accurate results, but this research goes beyond the scope of this thesis.

6.3 Implementation of IM-SEQ

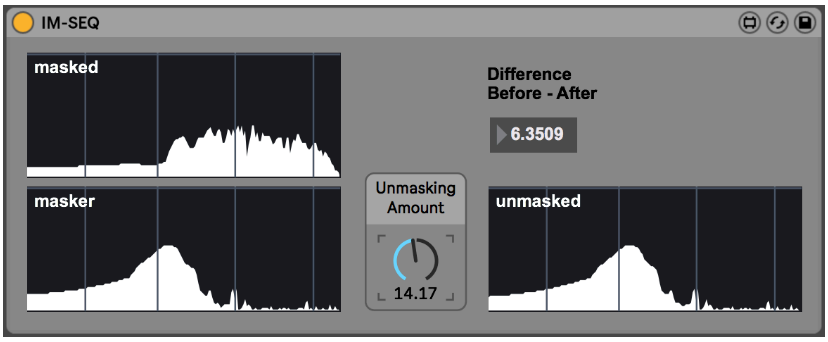

Figure 7. User interface of IM-SEQ.

The last tool implements a spectrum equaliser with an FFT size of 4096. The advantage of using spectral processing is that it is easy to implement in Max for Live and each of the 2048 bins of this implementation can run as separate parallel threads at the same time, also that this implementation analyses the incoming audio in the frequency domain, where we are comparing spectral information directly. Each thread runs its process of computing MUR and can attenuate itself according to the masking amount set by the user. We created this implementation have more detail, synchrony and control in the unmasking process.

This version of the tool does not use 2048 bins for equalisation. From the work of Robert Koria [10], we have learned that frequencies above 400Hz do not need a significant amount of attenuation when unmasking, thus, in order to simplify the process and make it less CPU intensive, We are using the first 38 bins. The centre frequency for bin 38 is 409 Hz, which makes it enough for unmasking frequencies in the range mentioned before. Usually, in music, low frequencies contain the most dynamically active part of a recording.

When this device is running, it creates a concurrent set of 38 threads, where each one analyses its corresponding bin from the masker and masked signals. Once it has read the amplitude values of masker and masker, computes MUR and attenuation for the given bin and applies attenuation to the corresponding amplitude of the masker bin, all during the same FFT window. The process inside each bin is the same as the one described in section 6.1 for IM-WS.

Given the characteristics of FFT computing and resynthesis, this process can compute and apply attenuations to the data in the same window. With this approach, the attenuation applied corresponds to the data read, and it does not present latency issues such as the one mentioned in section 6.2 for IM-EQ.

This process uses spectral processing to unmask the masker signal and create a resynthesised version of the signal below 409 Hz. After computing MUR and its corresponding attenuations, in order to have the whole masker signal spectrum with attenuations included, it is necessary to apply the audio process described in equation 5.

$$Masker = Masker - FFT(Masker, 38) + FFT(MaskerAttenuated, 38) \quad (5)$$

Where *FFT(Masker, 38)* represents the resynthesised signal using the first 38 bins of the FFT representation of the original Masker signal. *FFT(Masker Attenuated, 38)* represents the resynthesised signal using the first 38 bins of the masker signal after the application of attenuations. At the end of this process, masker contains the unmasked signal.

At the end of this process, we resynthesize the signal below 409 Hz using inverse FFT. The reconstructed signal (unmasked masker) is part resynthesised and part original, so it may not produce the exact representation of the masker sound. After many tests, most sounds kept their timbral characteristics without distortion or noticeable artefacts.

7 How to use the tools

The unmasking tools have similar configurations, and in this chapter, We will explain how to configure these tools inside an Ableton Live session properly. All tools use the same settings and configurations unless specified.

7.1 Installation

These tools are available to download fromÂ https://github.com/jjsauma/msmc. We provide the tools as separate files, one tool per file. In order to use it, it is necessary to drag the file from the file manager application of the computer (Mac: Finder / Windows: File Explorer) to a channel inside the Ableton Live session. Also, it is possible to use more than one instance of the tools in the same session. The output from a tool can be used as the input of the next to create complex flows in the mixing process.

7.2 Where to allocate IM-Tools

When performing the mixing of a multitrack recording, as discussed in section 3.2, it is necessary to separate artistic decisions and settings from technical issues. The intention of the tools developed in this thesis is to be used in the technical part of a mix, placing it after all artistic settings and audio processes. For example, if a drum channel uses equalisation, compression and distortion, the IM tool must be used after distortion. Use the unmasking tools after all mixing processes in the effect chain, including volume setting of each channel; these tools are Post Fader devices.

IM tools can be set anywhere inside an Ableton Live session, but we recommend to place it into a dedicated channel. Figure 8 shows an example where we place IM-WS implementation in a channel named IM-WS. The channel must be configured as “In” in the Monitor options. In the same example, the channel “Drums” is the masked signal, and the “Bass” channel is the masker signal. Using routing options from Ableton Live, we send the masked signal to “track in” (or channels 1 and 2) of the IM device, sending the masker signal using channels 3 and 4 in the Audio To settings of the channel.

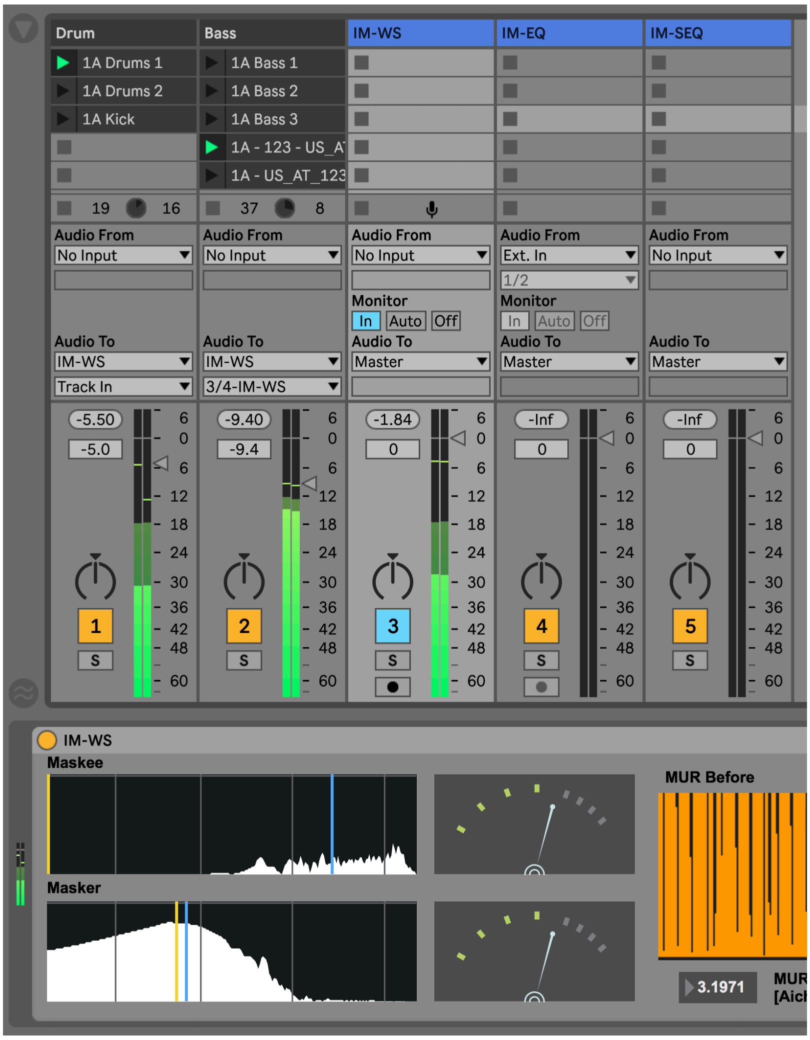

Figure 8. Where to place and how to route signals to IM Tools in an Ableton Live set.

7.3 Setting Parameters

Once we send the masker and masked signals into the IM Tool inputs, the displays will show the input signals on corresponding spectrometer windows and a dial control named “Unmasking Amount”. When we set this control to zero (the minimum value), the device is not applying any unmasking algorithm to the signals.

Setting Unmasking Amount to values greater than zero will modify the masker signal dynamically so we can hear the masked signal. The Unmask Amount control value is left to the user because it depends on taste. The Unmasking Amount will not be the same value for each recording, because spectral characteristics of different recordings are not the same. Also, the values among the different tools do not imply the same effect of unmasking, because different techniques are applied.

7.4 How to process more channels

If we need to unmask from more than one channel, the process must be done grouping channels in pairs sorting the masking channels by priority as proposed by [10]. This process is adapted for this research and described in Figure 9.

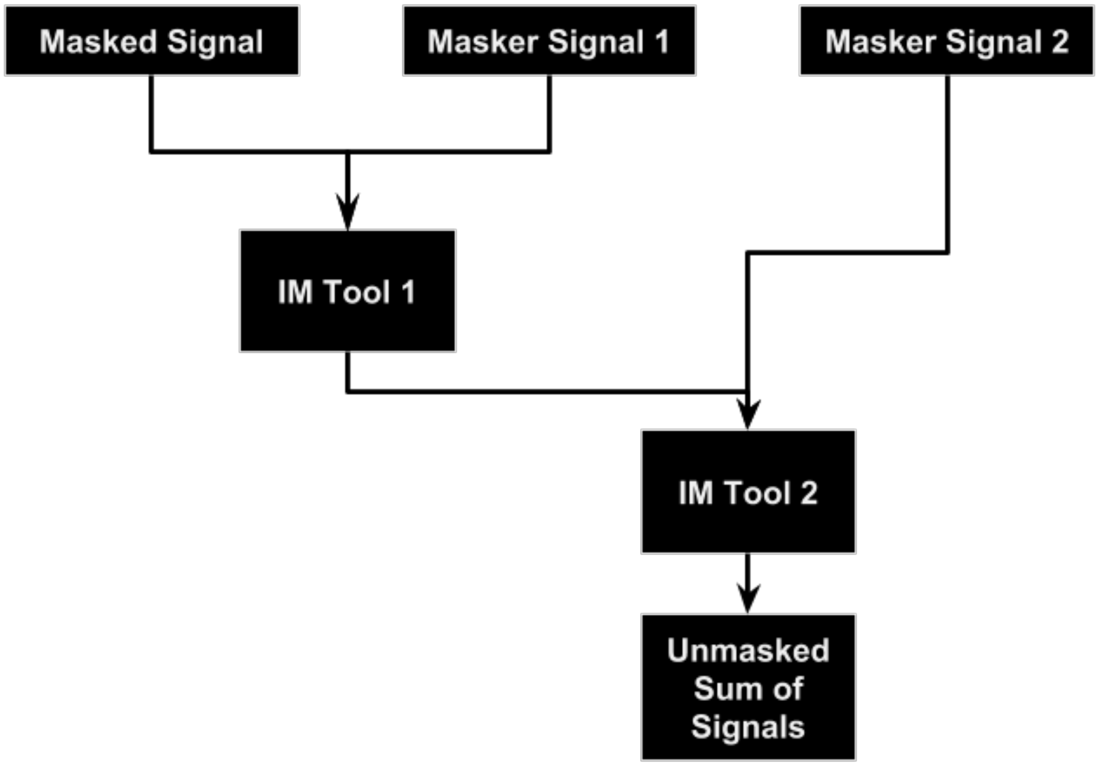

Figure 9. How to connect IM Tools to unmask from more than one channel.

8 Evaluation

We need to test the unmasking capabilities of the tools developed in this thesis to find out which one is adequate to reduce Auditory Masking in multitrack recordings.

Since we are measuring, and at the same time, computing MUR, the objective comparison between the tools is evident and easy to compute using this metric. On the other side, a subjective evaluation is necessary to find how each tool affects perception.

8.1 Objective Evaluation

Through this research, we are using MUR to measure masking quantitatively inside each IM Tool. MUR has proven to be an accurate measurement [1], and the effect is measured by computing the difference between MUR before the unmasking process, and MUR after the process. A comparison between the unmasking tools using this number may be difficult to interpret since the tools produce different results implementing different methods, and it is not a case of “more” or “less” unmasking that we measure with a number. The inherent measure for auditory masking, by definition, is perception. An objective evaluation of the outcome of IM Tools will not necessarily be accurate. For each tool, if we need to change the perceived unmasking, we need to change the Unmasking Amount value.

8.2 Subjective Evaluation

The final goal of this implementation and research is to create tools that allow the user to distinguish all sounds present in a mix. To have an opinion about how well the unmaking process works, we ran listening tests to measure the perception of human listeners.

Each user was presented with three pairs of sounds, in this case, a drum loop in one channel (the masked) and a bass sound on the other (the masker). Each pair is processed three times (one per tool). In total, each listener listens to 9 different mixes, grouped by the pair of sounds, named Mix 1, Mix 2, and Mix 3. After listening to the three different mixes for a pair of sounds, the user decides which mix sounds better and in his opinion, which presents more clarity and presence. The tool with the most votes is the one that performs better masking according to human perception.

In this test, we adjusted the values of Unmasking Amount at different values. We set each Unmasking Amount control to produce the same value of the difference between MUR before and after. This way, the different implementations can be considered equivalent.

8.3 Results

Listening tests were conducted online and gathered a total of 40 participants where they had to select the IM tool they think produces the better unmasked version of each of the three mixes. Figure 10 shows the of results for these tests.

The chart confirms an overall preference for IM-EQ, the method using 31 independent and dynamic equaliser bands. The second preferred method is IM-SEQ, which unmasks sounds using spectral processing. The least preferred method is IM-WS, using just one compressor to unmask a signal.

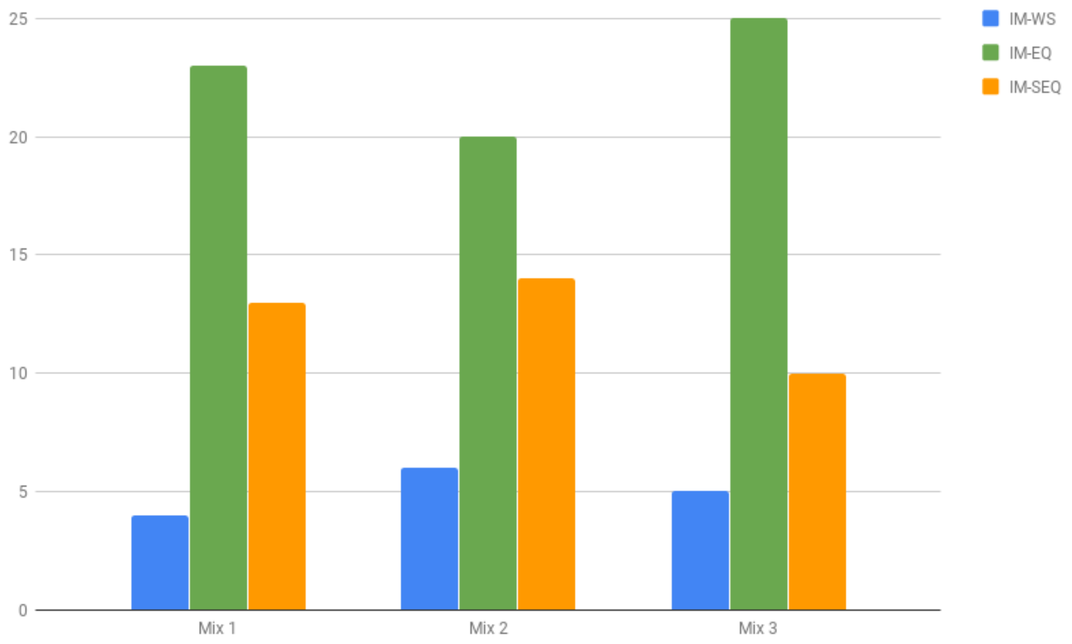

Figure 10. Results of listening tests. This chart show how many times an IM Tool was marked as better masking algorithm for each mix.

CPU usage is an essential factor when deciding which tool to use in a real-world situation. We ran tests on a MacBook Pro 2017 computer with a 2.9 GHz Intel Core i7 processor.

IM-WS produces de less CPU usage, and the rest of the tools produce results depending on their configuration. IM-EQ depends on the selected Sampling Interval. During tests, this tool utilised almost 50% of the available CPU using the minimum Sampling Interval. For higher values, this process was almost not noticeable. IM-SEQ uses around 10% of the available CPU. No settings are available to change in this tool, it is possible to reduce the FFT size to reduce CPU usage, but with a corresponding tradeoff in sound quality.

9 Conclusions and Future Work

9.1 Summary and Contribution

After a bibliographical review of intelligent mixing and auditory unmasking research (Chapter 4), a question about the feasibility of creating a definitive real-time unmasking device for mixing was proposed, along with research and documentation of the considerations for implementing such device. [1] Reflects on this research but concludes that no definitive solution has been proposed.

Previous research in the field led to creating three auditory unmasking devices implemented on Max for Live and running on Ableton Live with the intention of testing and using it on real time-performance of electronic music on a computer and still leaving CPU time available for music performance. Section 4.2 describes the contribution of previous research to this thesis and tools and how these results will help in the development and theoretical framework.

Chapter 6 describes the development of these devices, and documents to gain insights on the creation of real-time A-DAFX tools, along with equaliser theory for dynamic effects and spectral processing using real-time FFT computation.

In section 6.3, we propose and document a new method for auditory unmasking using spectral processing for real-time applications (which is usually CPU intensive). This method deals with the high processing percentage of other spectral implementations.

Listening tests were run to experiment human perception of the unmasking tools (Chapter 8). The results show that the 31 band equaliser implementation (IM-EQ) is better perceived when used dynamically to unmask sounds in a mix. The second best result was achieved by the spectral implementation (IM-SEQ), but it is far from the results obtained with IM-EQ. When comparing IM-EQ and IM-SEQ for perception, we can conclude that an equaliser performs a more natural sounding for auditory unmasking, than an spectral equaliser even when its behaviour is not completely precise and synchronous. IM-SEQ produces an extremely precise blend of the masker and masker into a new signal that may include resynthesised sounds that can make the perceived sound an artificial mix.

On the CPU usage aspect, IM-EQ can be less expensive, but needs user interaction and understanding of how the Sampling Interval parameter affects the mix. This calls for specialised knowledge of the performer, which we are trying to avoid when developing this tool. Comparing with IM-SEQ, results suggest that multithreading adds CPU usage to the solution. Applying equalisation with an FFT is therefore generally very inefficient compared to comparable time-domain filters. I can conclude that IM-EQ produces the better balance between CPU usage and perception.

Previous research intended to create a method or solution for generic tasks, the research made in this thesis seems to prove that adding limitations in order to focus on a defined environment helps us create a device that can produce a reasonable solution.

9.2 Future Work

9.2.3 Multitrack device

Current devices can successfully unmask two channels, a device for unmasking more channels is possible to be developed instead of the suggested option of using many instances of IM Tools. This development may be able to optimise processing when analysing many channels at the same time. It will take more tests to decide how many channels can be processed by the device at the same time and still leave CPU time available for musical performance.

9.2.4 Using auditory system-like frequency scales

Using Mel or Bark Scales because it models the human auditory system behaviour may seem the following step in the development of a new version of IM-EQ. The implementation in this thesis allows for an easy reprogramming of a multi-band equaliser using these scales. A comparison between an ANSI S1.11 [31], Mel and Bark Scales would help in developing a more natural-sounding device for this task. Also, it will be easy to add a user interface control to select the frequency scale according to the needs of the performer.

9.2.5 Finding the optimal Sampling Interval level

We included the control of the Sampling Interval parameter to help save CPU usage while processing. More research is needed to determine the optimal interval and the cases where we can use fixed intervals according to the signals we are processing. Fast Sampling Interval will help to unmask rhythmic and dynamic signals, and a slow value will use less CPU when unmasking sustained and continuous sounds. This device can have profiles or use Machine Learning algorithms to find the best Sampling Interval setting according to the processed signal.

9.2.6 Mixing and Mastering

Spectral processing provides an accurate mix of the two channels, and the real-time method developed here may be adapted and tested for purposes of Frequency Spectrum Target Mixes, which is a current trend in digital mastering systems and production techniques [34], [35].

9.2.7 DJ-style mix

IM-SEQ has proved to be a blending tool that may have applications in DJ-style mixes, where the goal is to create a transparent blend of songs one after another. IM-SEQ blends two signals in a way that it avoids big amplitude jumps and sudden changes avoiding significant amplitude changes by continuously analyzing of their frequency spectrum.

9.3 Reproducibility

IM-Tools developed and tested during this thesis, as well as surveys and data, are stored in https://github.com/jjsauma/msmc. It includes Excel spreadsheets, Max for Live Patches and Ableton Live Set used for this research.

10 Bibliography

1. De Man, B., Reiss, J. Ten years of automatic mixing. Queen Mary University of London (2017).
2. Kokkinis, E. Artificial Intelligence in Music Making: is a Jobless Future Ahead of Us? (2017). Retrieved from: http://www.avidblogs.com/artificial-intelligence-music-making-jobless-future-ahead-us/
3. Moore, B. An introduction to the psychology of hearing. Academic Press, 6th Edition (2013).
4. Hermes, K. Towards measuring music mix quality: the factors contributing to the spectral clarity of single sounds. University of Surrey (2017).
5. Hafezi, S., Reiss, J. Autonomous Multitrack Equalisation Based on Masking Reduction. Centre for Digital Music. Queen Mary University of London, London, UK (2015).
6. Martinez Ramirez, M., Reiss, J. Ste, Audio Mixing as a Content-Based Transformation of Audio Features. Centre for Digital Music. Queen Mary University of London, London, UK (2017).
7. Aichinger, P., Sontacchi A., Schneider-Stickler B. Describing the Transparency of Mixdowns: The Masked-to-Unmasked-Ratio. 130th Audio Engineering Society Convention (2011).
8. Vega, S., Janer, J. Quantifying Masking in Multi-Track Recordings. Universitat Pompeu Fabra (2010).
9. Puckette, M. Apel, T., Zicarelli Real-time audio analysis tools for Pd and MSP. UCSD, Cycling ’74 (1998).
10. Koria, R. Real-Time Adaptive Audio Mixing System Using Inter-Spectral Dependencies. Linköping University. 2016.
11. D. Dugan, “Automatic microphone mixing,” J. Audio Eng. Soc., vol. 23, June 1975.
12. Ronan, D. Ma, Z. Nanamanra, P. Gunes, H. Reiss. J. Automatic Minimisation of Masking in Multitrack Audio using Subgroups (2018).
13. ANSI S1.11: Specification for Octave, Half-Octave, and Third Octave Band Filter Sets. American National Standards Institute (2004).
14. United States Patent 3,992,584. (1976).

15. Wakefield, J., Dewey C. An Investigation into the efficacy of methods commonly employed by mix engineers to reduce frequency masking in the mixing of multitrack musical recordings. University of Huddersfield (2015).
16. Perez-Gonzalez, E., Reiss, J. Improved Control Of Selective Minimisation of Masking Using Inter-Channel Dependency Effects. Centre for Digital Music, Queen Mary University of London (2008).
17. Vega, S., Janer J. Quantifying Masking In Multi-Track Recordings. Universitat Pompeu Fabra (2010).
18. Gibson, D. The Art Of Mixing. A Visual Guide to Recording Engineering, and Production. Mix Books. 2005.
19. Owsinski, B. The Mixing Engineer's Handbook 4th Edition. Bobby Owsinski Media Group. 2017.
20. White, P. Mixing Essentials. Sound On Sound Magazine (2006). Retrieved from: www.soundonsound.com/techniques/mixing-essentials.
21. Ma, Z. Intelligent Tools for Multitrack Frequency and Dynamics Processing. Queen Mary University of London (2016).
22. Verfaille, V., Arfib, D. A-DAFX: Adaptive Digital Audio Effects. CNRS - LMA (2001).
23. Baytas, M., Göksun, T., Özcan, O. The Perception of Live-sequenced Electronic Music via Hearing and Sight. NIME'16 (2016).
24. Terrell, M., Simson, A., Sandler, M. The Mathematics of Mixing. Queen Mary University of London (2014).
25. Principles of Equalization (2013). Retrieved from: www.izotope.com/en/blog/mixing/principles-of-equalization.html
26. Välimaki, Vesa., Reiss, J. All About Audio Equalisation: Solutions and Frontiers. Department of Signal Processing and Acoustics, Aalto University (2016).
27. Charles, J. A Tutorial On Special Sound Processing Using Max/MSP and Jitter. Computer Music Journal. MIT (2008).
28. Muller, H. Designing Multithreaded And Multicore Audio Systems. XMOX Ltd. Bristol, UK (2011).
29. Real-Time Musical Applications on an Experimental Operating System for Multi-Core Processors. University of California, Berkeley (2011).
30. Elsea, P. The Art And Technique of Electroacoustic Music. A-R Editions (2013).
31. Dolson, D. The Phase Vocoder: A Tutorial. Computer Music Journal, Vol 10, No. 4. (Winter, 1986).
32. Schuett, N. The Effects of Latency on Ensemble Performance. Stanford University (2002).

33. C. Bartlette, D. Headlam, M. Bocko, and G. Velikic. Effect of Network Latency on Interactive Musical Performance. Music Perception. Volume 24, Issue 1 (2006).
34. Elowsson, A., Friberg, A. Long-term Average Spectrum in Popular Music and its Relation to the Level of the Percussion. KTH Royal Institute of Technology, School of Computer Science and Communication, Speech, Music and Hearing. 2017.
35. Duarte Pestana, P., Ma, Z., Reiss, D., Barbosa, A. Black, D. Spectral Characteristics of Popular Commercial Recordings 1950-2010. Catholic University of Oporto, Queen Mary University of London, 2013.