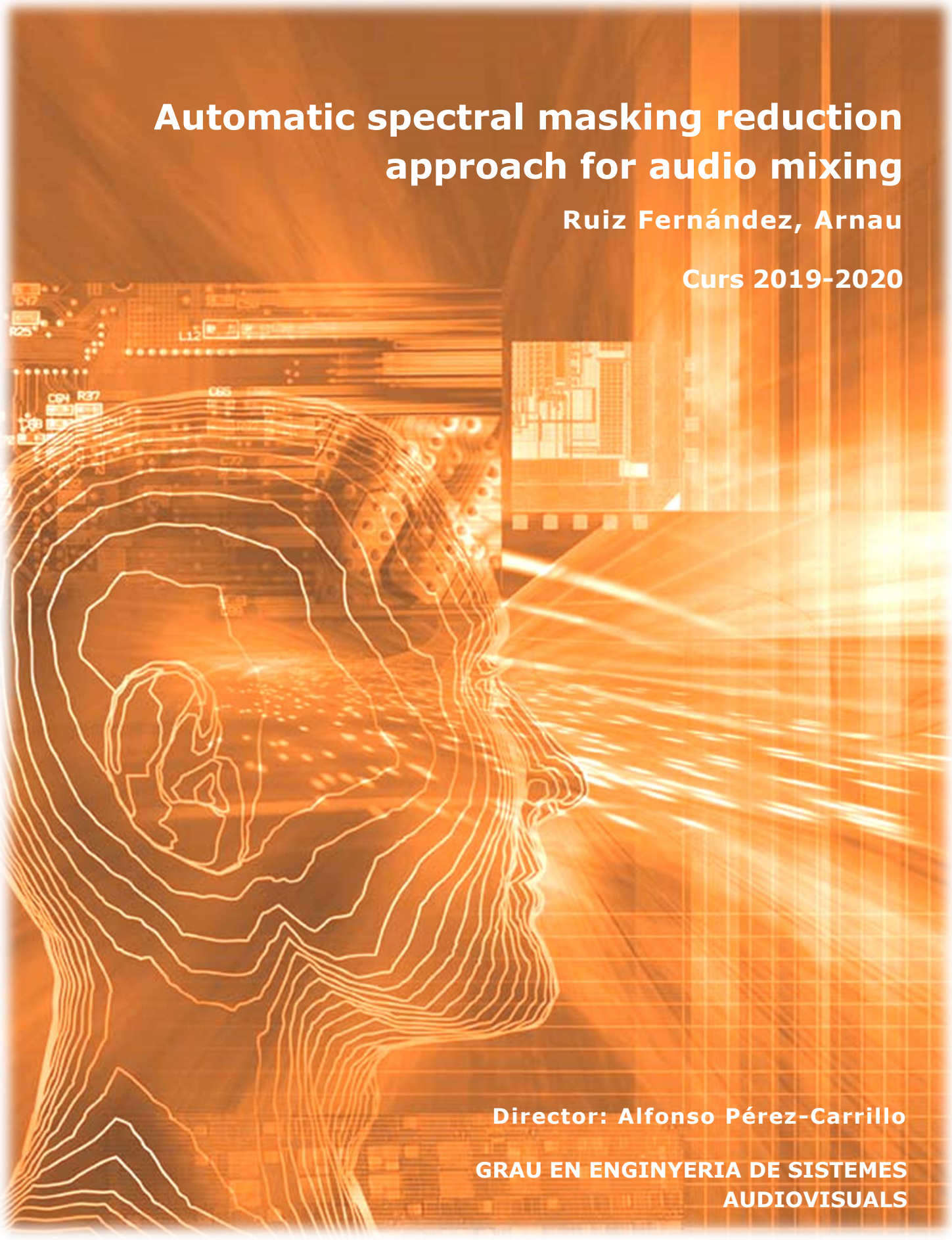
Automatic spectral masking reduction approach for audio mixing

Ruiz Fernández, Arnau

Curs 2019-2020



Director: Alfonso Pérez-Carrillo

GRAU EN ENGINYERIA DE SISTEMES AUDIOVISUALS

upf. Universitat Pompeu Fabra Barcelona
Escola Superior Politècnica

Treball de Fi de Grau

Automatic spectral masking reduction approach for audio mixing

Pompeu Fabra University

Arnau Ruiz Fernández

BACHELOR THESIS UPF / 2020

SUPERVISOR

Alfonso Pérez-Carrillo

Department Music Technology Group

To my love...

Abstract

Automatic mixing topic is becoming relevant nowadays due to the more affordable prices of the audio technologies. Anyone with a sound card and a computer can record from home, but it does not mean that they have the ability and the knowledge to perform a good mixture. This method allows us to increase automatically the clarity and understandability of each track of the song based on decreasing the frequency masking between instruments. Among all the processes implied in the mixing of a song, this method focuses in the equalization by minimizing frequency masking. The algorithm divides each track in ten frequency bands and ranks them using the magnitude as a reference. Then for each pair of tracks, it compares the rankings and the magnitudes on each frequency band to decide if masking is occurring. If that is the case, it reduces the magnitude on those bands using filters. Two offline versions have been implemented using Matlab, one fully-autonomous version and a semi-autonomous version, that requires some manual interaction. Also an online version has been implemented using JUCE, in order to create a VST plugin that can be run by a DAW software. The algorithm presents noticeable changes in the clarity of the instruments, even though sometimes needs some parameter adjustments to adapt it to the song necessities and context.

Resum

L'àrea de la mescla automàtica és cada cop més rellevant a causa de l'abaratiment de les tecnologies de gravació d'àudio. Qualsevol amb una targeta de so i un ordinador pot gravar des de casa, però no necessàriament implica que sàpiga mesclar música. Aquest mètode permet augmentar la claredat i distinció de les diferents pistes d'àudio de manera automàtica, basat en reduir l'emmascarament espectral. Entre els diferents processos d'una mescla, aquest mètode se centra en l'equalització per reduir l'emmascarament entre instruments. L'algoritme separa cada pista en deu bandes freqüencials i fa un rànquing de les bandes amb més magnitud. Es compara els rànquings i les magnituds de cada parell d'instruments per decidir si s'està produint emmascarament freqüencial. En aquest cas, es redueix la magnitud en aquella banda en l'instrument culpable de l'emmascarament. S'han creat dues versions "offline" utilitzant Matlab, un completament autònom i un semiautònom que requereix interacció manual. També s'ha creat un arxiu VST utilitzant JUCE per poder utilitzar l'algoritme en programes DAW. L'algoritme presenta canvis evidents en la claredat dels instruments, encara que necessita de cert ajustament dels paràmetres per ajustar-se correctament al context de la cançó.

Summary

List of Figures

List of Tables

Chapter 1

INTRODUCTION

One common challenge when mixing a song is the masking reduction, which is a psycho-acoustic effect that happens when the presence of a signal makes inaudible or partially inaudible another signal. When masking is caused by two signals happening at the same time it is called *spectral masking* or *simultaneous masking*. It is frequency-independent, meaning that not all frequencies are being masked equally. Different audio algorithms help to reduce spectral masking (e.g. compression, panning, levelling or equalizing). This project approaches mixing by masking reduction in frequency domain as an equalizer.

1.1 Motivation

Back in the days, only professional studios had the required equipment to record music but now, thanks to the cheapening and digitization of audio technology, almost any musician can record at home by just buying a sound interface and plugging in a microphone or an electric instrument. In addition, there are a large variety of free tools to edit and record audio. It is very common to find musicians that record their musical ideas, but they do not have the knowledge to mix it properly. This is one of the reasons why these days, automatic mixing is becoming relevant. Furthermore, automatic mixing speeds up the mixing process for recordings that are not intended to be final, for example, the audio that a songwriter creates as a guide for the other musicians, so they can learn the song easier and faster. Also, as J. Reiss [12] described, it can be useful for live music performances.

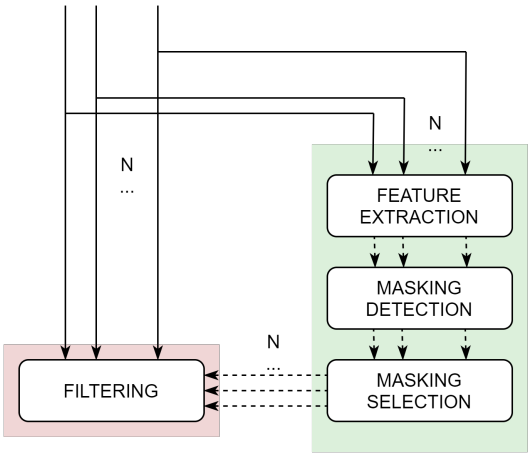


Figure 1.1: Block diagram of the system. The analysis block is marked in green, the processing block in red. Solid lines are for audio, dashed lines are for feature data. Adapted from J. Reiss [1].

Another motivation behind this project came while I was in high school and I discovered audio mixing. This encouraged me to start my engineering studies and to carry on this project applying the audiovisual concepts I have learned among these years.

1.2 Objective

The goal of this project is to detect the spectral areas where masking is happening and reduce them on the signals causing the masking.

To achieve this objective I adapted the algorithm described by J. Reiss [1], which is an autonomous multi-track masking reduction algorithm with n inputs and n outputs as shown at figure 1.1. In the **analysis** stage, the system reads the n inputs, extracts the features, and compares the tracks features to detect the masking between tracks. Once the masking has been evaluated and selected on the analysis stage, the system applies the proper filtering on each track in the **processing** stage. J. Reiss [1] performs a mix-down on the output of the system, but in my approach, I will keep the n tracks separated on the output, so further processing can be done on the mixing.

The objectives are the following:

- Build an offline version using Matlab.

- Build an online version that can be loaded in a DAW software using the VST format and JUCE to create the plugin. It has to be optimized so it can be run in real-time.

- The output must have less masking, increasing the understandability and clarity of the tracks.

1.3 Background

The first time automatic mixing was mentioned was in 1975 by Dan Dugan. He described an automatic multi-microphone gain and gate system [13][14]. It started as a way to help engineers to manage all the microphone volumes, avoiding feedback and noise from unused microphones on live performances by muting or reducing the volume of those microphones. In 2007 E. P. Gonzalez and J. Reiss [15] made use of modern digital signal processors and the digital consoles to create an automatic stereo panner. Since then, other automatic approaches have been made on the different mixing areas as compression [16][17][18][19], artificial reverberation [20][21], levelling [22] and other approaches that combine different algorithms to create a more complete and complex automatic mixer [23].

Other related works to the present project are those about automatic equalization. Perez-Gonzalez and J. Reiss [24] worked on the first autonomous equalizer based on a cross-adaptive method, which assumes that an optimal mixing is when the "loudness per band tends to the overall average loudness of the signal" [24]. Based on this solution. authors could reduce the overall masking by just decreasing the probability of masking to occur. However, following this approach, does not necessarily mean that each track has a proper equalization to enhance the instrument.

Other approaches on equalization are described in [25][26], where the main objective is tone enhancement instead of masking reduction, i.e., improving and enhancing the overall spectrum of the master track[1] using a pitch tracker and an spectrum-matching algorithm [26]. Further work on the area of masking reduction is described by D. Ronan [27], which combines equalization and dynamic compression to achieve a better masking reduction. It also adds the sub-grouping

[1]The master track term is commonly used to describe the mix down of all the tracks.

idea that allows creating different equalization and compression presets based on the instrument type.

1.4 Outline

This thesis contains six chapters followed by the bibliography. It contains the following chapters:

1. Chapter 1 introduces the topic and context of the thesis. It starts defining the automatic mixing problem and the project motivation followed by the objectives. Some related works are presented on the "Background" section and an overall overview of the thesis is presented.

2. Chapter 2 explain some of the theoretical concepts related with auditory masking. It starts with the anatomy of the human auditory system followed by the human frequency processing model. It shows some methods to simulate the ear sound processing, including its non-linearities. Then the relation of this model with the auditory masking is shown.

3. Chapter 3 explain some of the audio treatments that a professional audio engineer performs when mixing a song. It also gives context on the application of the thesis project on audio mixing and details on how the equalization process works. In addition, it explains other processes as "Balance", "Panorama", "Audio effects", "Dynamics" and "Interest", and the further process of "Mastering".

4. In chapter 4 is shown the algorithm structure, detailing each one of the stages, starting with the audio analysis and finishing with the filtering. It also presents some variations of the algorithm made for different approaches, an online VST implementation and two offline Matlab implementations.

5. In chapter 5 is explained the procedure and metrics used for the evaluation. As this algorithm is applied into a psycho-acoustical area, both subjective and objective evaluation are used to evaluate the efficiency. Then the results are shown for each one of the implementations showing the effects of the parameters.

6. In chapter 6 I discuss the results given in chapter 5 to came up with a conclusion and a reflection about the performance of the algorithm together with some improvements and further investigation.

Chapter 2

MASKING PRINCIPLES

The main aim of this work is to implement an algorithm that can reduce the masking on a sound mixture, but we first have to understand how masking occurs and how can we model and measure it. To understand properly the masking effect we also have to understand how the human auditory system works, but focusing on the frequency processing.

When a sound can't be perceived because of the presence of another sound is called *masking*. The sound that causes the masking is called *masker*, and the one masked is called *maskee*. There are two types of masking:

- *Frequency masking* when the two signals occur at the same time and there is an overlap in the excited frequencies. Also called simultaneous masking or spectral masking.

- *Temporal masking* happens when two short sound occurs very close in time and one of them is inaudible due to the presence of the other. Also called non-simultaneous masking, it can be caused in two directions, first, the masker is emitted and then the maskee (forward masking), or inverted, being the maskee first and then the masker (backward masking), as shown in figure 2.1.

The one we want to reduce is the frequency masking, which is the most common in a multi-track mixture.

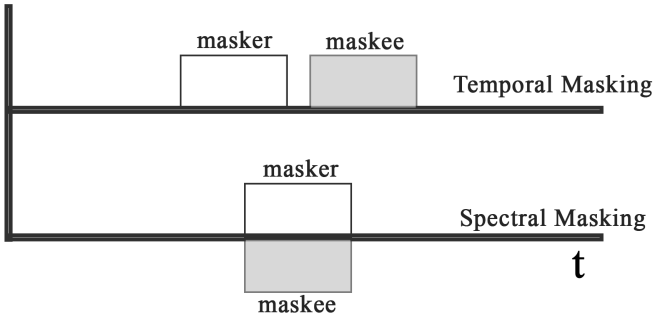


Figure 2.1: Graphic representation of the two classes of masking, temporal and spectral masking respectively. The horizontal axis is represented the time. When the signals are not simultaneous and one of them is masked by the other is called "Temporal Masking", while if the signals are simultaneous it is called "Spectral Masking".

2.1 Anatomy of the human auditory system

Some human auditory system models proposed, like the one proposed by B. Moore [10], tries to replicate the way the human auditory system processes the sound, so understanding the anatomy of the human ear is essential. It is composed by the *peripheral system*, the responsible of transforming the incoming pressure waves into electrical impulses, and the *central nervous system*, which analyze all the information from the peripheral system to distinguish between the different sound sources and understand the information and context of each sound, e.g, the words of a speech, distinguish music, background noise, an animal, etc...

The peripheral system

The ear consists of the outer ear, the middle ear, and the inner ear, as shown in figure 2.2. Each one of these parts plays a relevant role in the sound sense, but I will focus on the inner ear, where most of the signal processing is made.

The outer ear is responsible for the transduction of the sound into the ear, i.e., to adapt the external sound to the auditory canal. It consists of the pinna (or auricle, the visible part of the ear) and the external auditory canal. The pinna is responsible for the geolocalisation of the sound by the non-symmetrical geometry

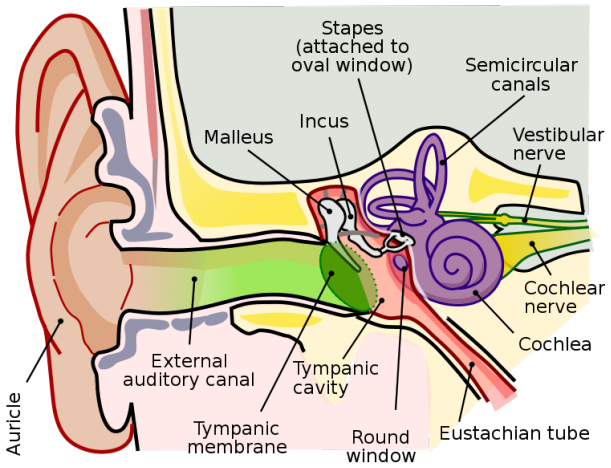


Figure 2.2: Human ear structure. Green colour represents the outer ear, red represents the middle ear and purple colour represents the inner ear. Image from Chittka L, Brockmann A (2005) Perception Space [2].

respect the anterior and posterior directions of the head. The pinna together with the auditory canal acts as an impedance matcher and as a resonator. At the end of the auditory canal, the eardrum (or tympanic membrane) is located, which is a membrane that vibrates as the pressure waves come in, acting as a microphone does, connecting with the middle ear.

The middle ear transfers the mechanical waves coming from the outer ear to the cochlea, which is filled with a fluid where the sound will travel. The tympanic membrane is connected to three concatenated bones, starting with the malleus, then the incus, and finally the stapes, that are connected to the inner ear. The middle ear has to ensure the proper transmission of the waves, and to do so, it contains small muscles that prevent possible damage dealt by loud sounds. It also has a conduct that stabilizes the pressure inside the middle ear and the outer ear, so the eardrum is in balance and the proper position.

The inner ear is responsible to do the spectral fragmentation of the sound, but it also contains the semicircular canals, responsible for the rotation sense on the body. The cochlea performs the spectral detection and is a spiral-shaped cavity with a conduct filled with a fluid that starts at the other side of the stapes (the oval window) and finishes at the round window, as shown at figure 2.3 and 2.4. This conduct is divided into two parts by the Helicotrema: the Scala Vestibuli (or vestibular conduct) from the oval window to the helicotrema, and the Scala Timpani (or basilar conduct) from the helicotrema to the round window. Also,

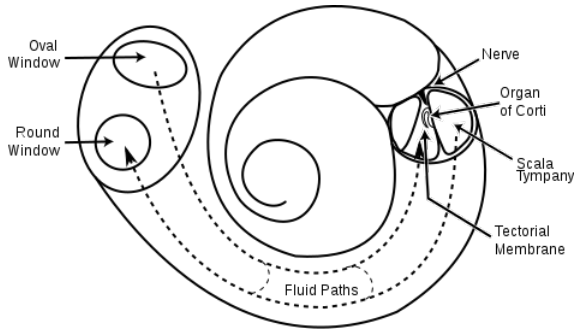


Figure 2.3: Structure of the human cochlea. The sound is transmitted through a conduct starting at the oval window that crosses all the cochlea in both directions finishing at the round window. Image from Wikimedia Commons [3].

between both conducts, there's a third conduct called scala media.

On figure 2.3 is shown a transversal section of the cochlea, easier to see how the conducts are distributed. The *organ of corti* is responsible for the sound sense, that transforms mechanical movement into electrical impulses that the brain can process and give us all the information about that incoming sound. Notice that that recieved signal contains several sound sources, but using the differences between the output of each one the ears and the spectral segmentation, it can recognize almost every individual source. The spectral segmentation is performed by the *basilar* membrane by changing its height within the cochlea, see figure 2.4. At the base of the cochlea, the basilar membrane is thinner than the helicotrema, performing that way a "tuning" process that makes the high-frequency components vibrate at the earlier stages of the basilar membrane and the low frequency vibrate

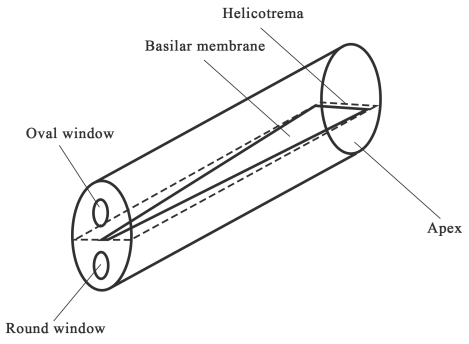


Figure 2.4: Cochlea showed as it was straight, showing how the basilar membranes get bigger as the sound travels through the cochlea.

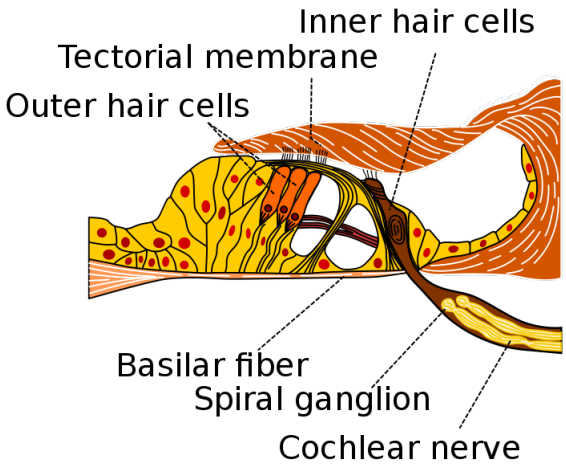


Figure 2.5: Frontal view of the organ of corti. Image from Wikimedia Commons [4].

at the end of the basilar membrane. The frequency at any point of the basilar membrane that gives maximum amplitude is called *characteristic frequency*. Then, the wave is transmitted by the stapes to the oval window and it travels along the *scala vestibuli* and *scala tympani*, ending at the *round window*, where it is dumped.

As mentioned before, the organ of Corti is attached to the basilar membrane (BM), so when the BM vibrates the organ of Corti does too. The organ of Corti contains the nerve cells, called *hair cells*, this translates the vibration into electrical impulses that are sent to the brain through the *fiber cells*. The electrical impulses are generated thanks to the *stereocillias*, small filaments attached to the hair cells and the tectorial membrane (see 2.5). The stereocillias move when the organ of Corti vibrates, allowing to the hair cells to create impulses depending on the amount of movement of it. Mention that the hair cell maximum response at a certain point of the BM correspond to the characteristic frequency of the basilar membrane where it is attached. This characteristic is called *tonotopic organisation* [28].

2.2 Frequency processing models

The whole hair cell system of the cochlea creates a spectral analysis of the incoming sound, but it does not have an infinite resolution. The *frequency sensitivity* or *frequency resolution* is the minimum frequency difference between two sinusoids

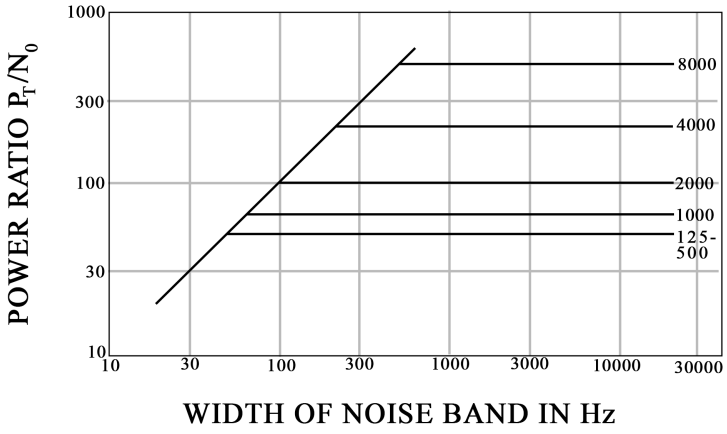


Figure 2.6: Power ratio versus the width of the noise. The horizontal lines show how the critical frequency stabilises depending of the frequency of the sinusoid, higher the frequency higher the critical bandwidth. From Fletcher [5]

that the human can distinguish.

Fletcher first mentioned the concept of *auditory filter* on 1933 [29] and redefined it on 1940 [5] and he explained the frequency resolution concept by modelling the human auditory system with rectangular shaped pass-band filters with same centre frequency than the characteristic frequency of the basilar membrane where the cells are located. Fletcher experimented to measure the bandwidth of these band-pass filters [5], using a sinusoid with constant frequency and adding a band-passed noise centred at the frequency of the sinusoid, keeping constant the power density. He then plotted the minimum required intensity of the sinusoid to be detected respect the noise power versus the band-passed noise bandwidth, as shown in figure 2.6. In this graph two clear stages are shown: first the power ratio increases linearly and then at a certain frequency of the sinusoid (called *critical frequency*) the power ratio curve gets flat. The critical frequency gets higher as the sinusoid frequency increase.

This model is based on some assumptions that are not completely true and is not taking into account some non-linearities, for example, the detection of the frequencies does not fall on just one auditory filter but the detection is expanded to the neighbor filters, as explained by B. Moore [10].

2.3 Calculation of the critical bandwidth

On Fletcher's [5] model rectangular windows are assumed on the auditory filter and also that the masking threshold occurs when the noise power inside the auditory filter and the signal power is equal, i.e, when the signal to noise ratio K on the same auditory filter equals 1. Then we can write an expression to calculate the necessary power of the sinusoid power to be able to be heard at the presence of a band-filtered noise with power density N_0:

$$P_S = K \cdot \underbrace{\mathrm{CB} \cdot N_0}_{\text{Noise power}}. \tag{2.1}$$

We can isolate the critical band (CB) frequency and calculate it using the noise power density ($\mathrm{N_0}$) and the signal power ($\mathrm{P_S}$). More recent experiments [10] estimates that the real value of K is about 0.4 instead of 1.

We can rewrite the previous expression to fit any window shape. Then the noise power inside the auditory filter is

$$P_N = \int_0^\infty W(f)N(f)\mathrm{d}f, \tag{2.2}$$

with $W(f)$ being the window weight centred at the critical frequency of the auditory filter and $N(f)$ the noise power. Then, we can rewrite equation 2.1 as

$$P_S = K \int_0^\infty W(f)N(f)\ \mathrm{d}f. \tag{2.3}$$

Using this equation we can estimate the auditory filter shape and the curve of the masking threshold for a sinusoid versus the noise frequency with methods as the *Psychophysical tuning curves*, the *Notched Noise* method or the *Rippled-Noise* method.

2.3.1 Psychophysical tuning curves

The psychophysical tuning curves are a similar method to the previously Fletcher's mentioned method but now, instead of changing the bandwidth of the noise, we

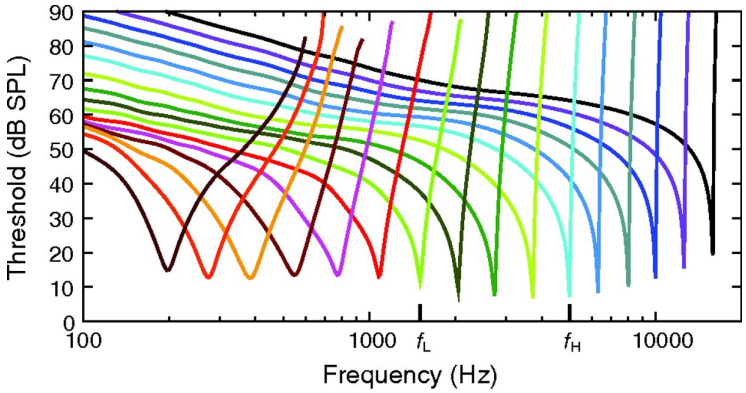


Figure 2.7: Each colour represents the threshold curve for a different frequency, and by combining them the tuning curve can be calculated. Figure from Reichenbach T, Hudspeth AJ [6].

have a fixed bandwidth and we find the masking threshold versus the noise centre frequency. If we repeat this experiment for different characteristic frequencies then we can have an overall curve of the threshold versus the noise centre frequency as shown in Figure 2.7.

Mention that this experiment is not taking into account an effect called *off-frequency* listening, which means that the auditory filter centre with the highest response to the signal does not necessarily correspond to the signal frequency and that also means the masking or frequency recognition does not recall only in one auditory filter, the combination of the different auditory filter can be useful on the frequency recognition.

2.3.2 The notched-Noise method

The previous method does not contemplate the off-frequency listening effect and for that reason, B. Moore [10] proposes a model that softens the effect of the off-frequency listening. A noise signal with a notch around the signal frequency is used instead, as shown in Figure 2.8, and then the signal power equation 2.3 is modified to fit the new noise. In this case, the noise power inside the auditory filter is the sum of the left and right part of the noise that is inside the filter, which means just splitting the integral by two:

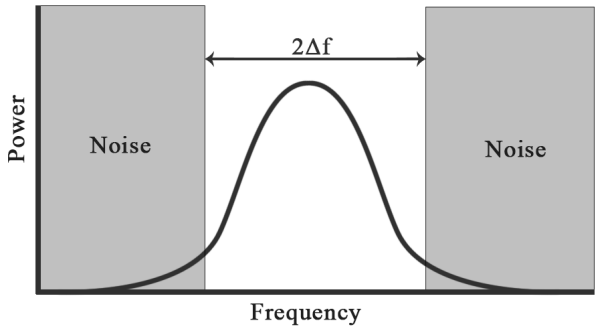


Figure 2.8: Notch-noise method. Noise with a notch centred at the centre of the auditory filter with notch width Δf.

$$P_S = K \int_0^{f_c - \Delta f} W(f) N(f)\, \mathrm{d}f + \int_{f_c + \Delta f}^{\infty} W(f) N(f)\, \mathrm{d}f, \tag{2.4}$$

where f_c is the signal frequency and Δf is half of the noise notch size. If we use a constant power noise, we could substitute the noise power $N(f)$ for the noise power density N_0 outside the integral:

$$P_S = K N_0 \int_0^{f_c - \Delta f} W(f)\, \mathrm{d}f + K N_0 \int_{f_c + \Delta f}^{\infty} W(f)\, \mathrm{d}f. \tag{2.5}$$

But the auditory filters seems to be non-linear (as discussed later on section 2.4), especially on high frequencies, and the last expression (2.5) assumes symmetry around the centre frequency. To be more precise on the auditory filter shape Patterson, Nimmo-Smith, Weber, and Milroy [30] suggested a filter function called *roex*. Using a new frequency variable $g = |f - f_c|/f_c$ then the filter shape $W(f)$ can be expressed as

$$W(g) = (1 + pg)e^{-pg}, \tag{2.6}$$

where p is a variable that defines both the slope and bandwidth of the filter. Then we can use a different p value for each side of $W(f)$, giving us a function that can predict quite well the shape of $W(f)$. Nevertheless, we have to take into account other non-linearities for example the fact that with higher levels the shape of the auditory filters compresses, so Patterson et al. [30] also propose a new variable r to solve this problem. Then the window shape can be written as

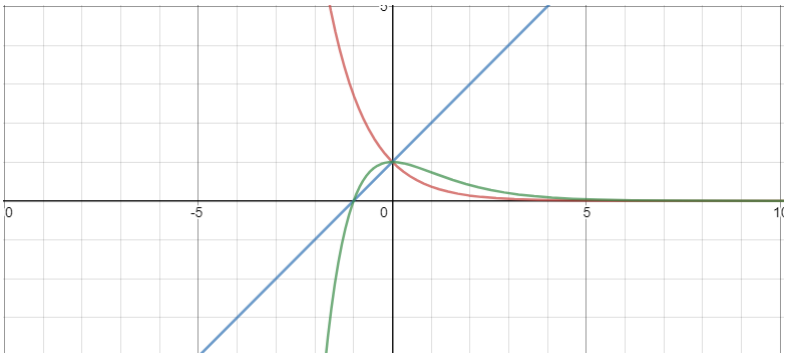


Figure 2.9: Red function is the exponential function, blue is the line function and green is the product of both functions.

$$W(g) = (1 - r)(1 + pg)e^{-pg} + r, \tag{2.7}$$

and in this case a same value of r on both sides seems to be enough [31]. One last function is given by [30] when a large variety of noise width is used and when the noise level is high because the previous function gives some deviation on the results. With this one, the error is reduced and it adds another exponential function with a variable t:

$$W(g) = (1 - r)(1 + pg)e^{-pg} + r(1 + tg)e^{-tg}. \tag{2.8}$$

To understand all these new variables let's just start with the first roex function proposed, the equation 2.6. We have two terms product, the first one $1 + pg$ is a line and the second one, e^{-pg}, is a negative exponential function, the functions can

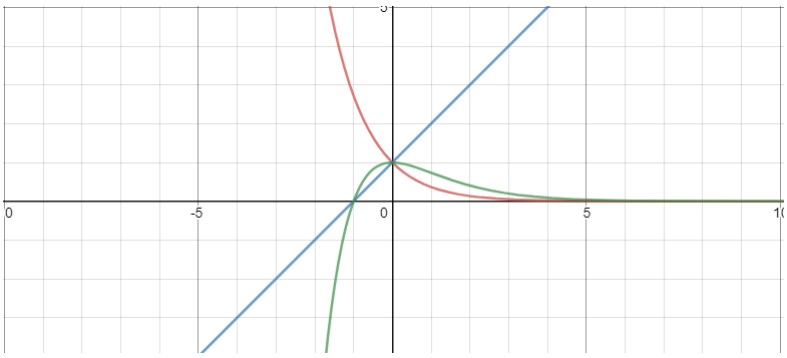


Figure 2.10: Red function is the exponential function, blue is the line function and green is the product of both functions.

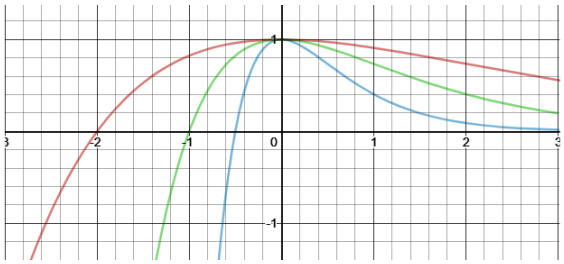


Figure 2.11: Equation 2.6 with $p = 0.5, 1, 2.$ represented with colour red, green and blue respectively. The higher p more rounded the curve.

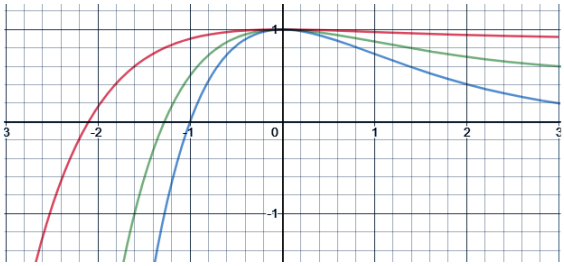


Figure 2.12: Equation 2.7 with $p = 1$ and $r = 0, 0.5, 0.9$ represented with colour red, green and blue respectively. With lower r the curve is flatter on the positive x axis but the slope on the negative x axis is similar, just starts to decrease on a lower x value.

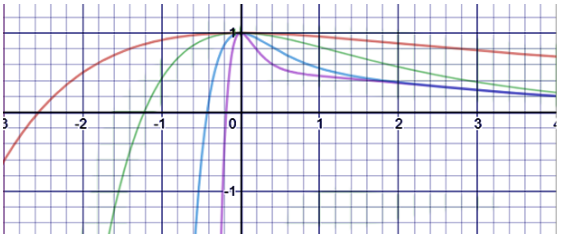


Figure 2.13: Equation 2.8 with $p = 0.5$, $r = 0.5$ and $t = 0, 1, 3, 5$ represented with colour red, green, blue and purple respectively. The parameter t is the same as p but for the secondary exponential function, and combining these two exponential functions with different "p" values you can control how fast the function reaches $y = 0$, see for example that for t=5 the function first decrese after $x = 0$ but then the slope is much horizontal.

be seen in Figure 2.10. As the negative exponential tends to $+\infty$ for $x < 0$, if the function is multiplied by a negative number it then tends to $-\infty$, that's what the line function is doing on the negative x. For the positives x, as the exponential tends to $y = 0$ the general function also does, but slower due to the effect of the line.

At figures 2.11, 2.12 and 2.13 the effect of the different parameters are shown. The first figure shows the first function proposed, $\mathrm{roex}(g, p)$. With small p value the function is wider, and bigger p value makes the function sharper. The second function $\mathrm{roex}(g, p, r)$ adds the parameter r, that is actually "horizontalising" the function, i.e. bigger r, more horizontal. The last one, $\mathrm{roex}(g, p, r, t)$ is actually the sum of two $\mathrm{roex}(g, p)$ functions, and you can control the mixture of both exponentials with the parameter r. With a 50% mixture ($r = 0.5$) and $p = 0.5$ (same function than the red curve at figure 2.11), increasing t creates a node on the right side that is coming closer to $x = 0$ as t increases. After that node, instead of getting closer to $y = 0$ in a negative exponential way it does more in a linear way.

2.3.3 Further considerations

When calculating the auditory filter shape we also have to take into account the effect of the outer and middle ear. So if the measurement experiment were made directly to the cochlea it would not be necessary to do any further consideration, but if the experiment is made by using headphones or loudspeakers, as the sound pass through the outer and middle ear you can not consider the emitted sound as the real input in the cochlea.

Another consideration is the actual eardrum impedance, so all together performs a cascade filter over the incident sound wave. Several measurements were made by Marko Hiipakka [32] showing how different can be the frequency response of the ear on every person. They created some artificial ear conducts and pinnas to understand the influence of the shape of the different ear parts.

2.4 Non-linearities of the auditory system

The auditory system due to its complexity has several non-linearities properties as the basilar membrane compression, combination tones, and two-tone supression[28]

or the non-linearities of the shape of the auditory filters [10] depending on the incoming sound level or frequency. These non-linearities play an important role in the masking problem.

2.4.1 Basilar membrane compression

The basilar membrane response does not act linearly when changing the input level. For a particular characteristic frequency on the basilar membrane, the maximum response should be when the signal is at the particular characteristic frequency, but that is not always true. For some incoming levels, the response at the characteristic frequency is not the maximum of the auditory filter, as shown in figure 2.14.

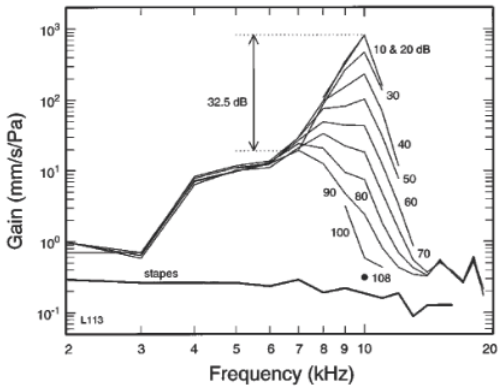


Figure 2.14: From [7]: response of an auditory filter of a chinchilla at characteristic frequency 10 kHz. Levels from 10 to 90 dB SPL.

2.4.2 Combination tones

The combination tones are artificially generated tones created by psychophysical processes when two tones are perceived simultaneously.

It is believed by Helmholtz [33] that the combination tones creation is due the distortion of the signal on the basilar membrane. For example and non-linear compressor with function $y(x) = \operatorname{sgn}(x) \log(1 + |x|)$ can produce a similar combination tone effect on the signal x (see figure 2.15). Helmholtz characterised this problem and solved the equation concluding that the first components generated

are $(f_2 - f_1)$, $(f_1 + f_2)$, $(2f_1)$ and $(2f_2)$. also secondary components $(2f_1 - f_2)$, $(2f_1 + f_2)$, $(2f_2 - f1)$, $(2f_2 + f1)$, $(3f_1)$ and $(3f_2)$. Not all components are equally present, $(2f_1 - f_2)$ or $(f2 - f1)$ are more noticeable.

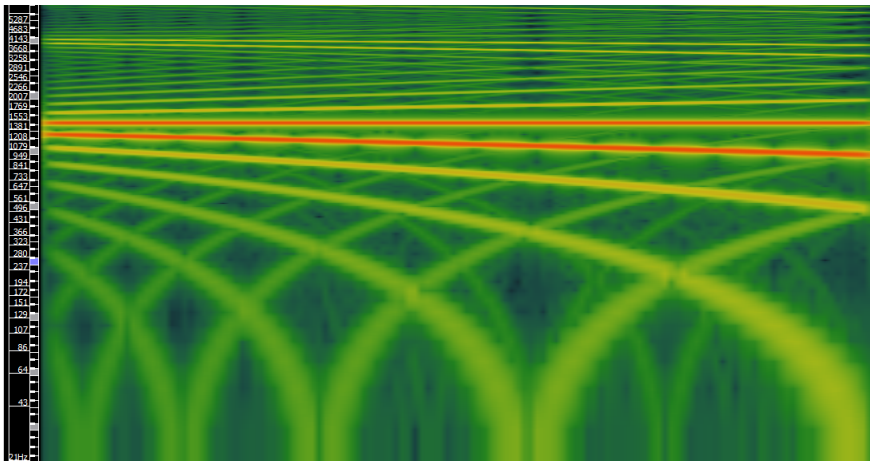


Figure 2.15: A 1500Hz tone and a falling tone starting at 1300Hz combined passing through a compressor characterised by the expression $y(x) = \text{sgn}(x)\log(1 + |x|)$. Additional tones appears that changes the frequencies proportionally to the falling tone frequency.

2.4.3 Two-tone suppression

Another non-linearity happens in the responses of the nerve fiber. It seems like there is a region of frequencies around the characteristic frequency on the basilar membrane that decreases the firing rate of the cells. Sachs and Kian [8] made an experiment with cats and plotted these areas in figure 2.16. This is highly related to masking because near tones to the CF can decrease the detection of that tone.

2.4.4 Variations of the shape of the auditory filters

As described before, the auditory filters do not have constant bandwidth, it depends on the frequency and the input level.

B. Moore [10] gave different approximations of the auditory filters bandwidths versus the frequency at figure 2.17. Based on the data from [34], [35], [36] and

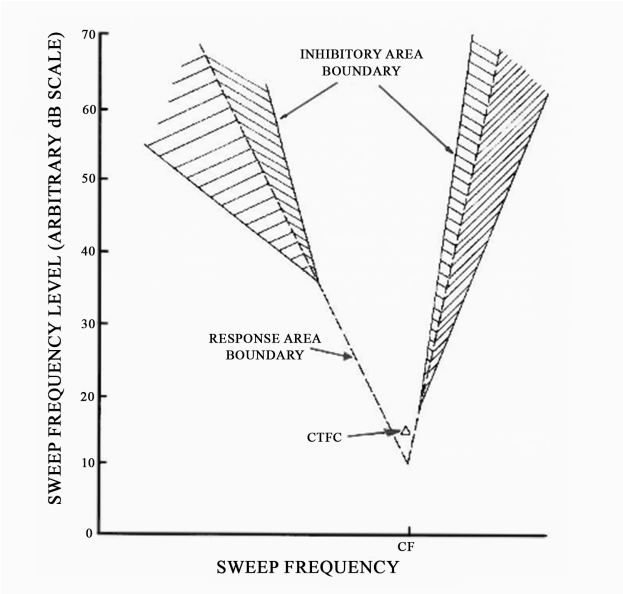


Figure 2.16: From [8]: Shaded areas represents these areas where a tone decrease the nerve fiber response. The line is the boundary of the auditory filter where it starts to have response for a tone.

[37] two expressions are proposed to describe the auditory bandwidth variation versus the frequency:

$$\mathrm{ERB} = 6.23f^2 + 93f + 28.5, \tag{2.9}$$

$$\mathrm{ERB} = 24.7(4.37f + 1). \tag{2.10}$$

An equation to know the filter number based on the equation 2.10 is

$$\mathrm{ERBnumber} = 21.4\log(4.37f + 1) \tag{2.11}$$

The shape of the auditory filters also changes by level variation, as shown in figure 2.18. Another effect of the non-linearity makes the filter shape to be flatted below the CF and step above the CF.

As mentioned in section 2.3.2 the function *roex* can be used in some of the variations to fit the filter shape depending on the frequency and the level.

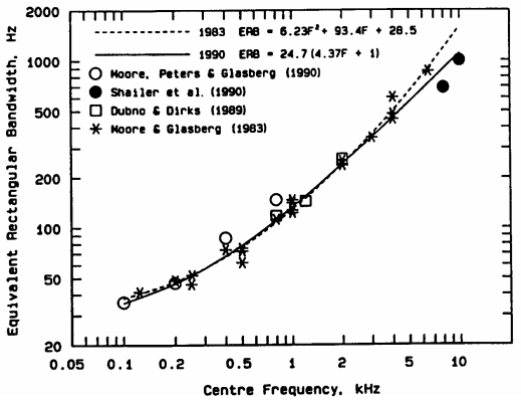


Figure 2.17: From Glasberg (1990)[9]: Data from four experiments marked with border circle, filled circles, squares and asterisks. Two proposed ERB functions are shown with discontinuous line and straight line.

2.5 Measuring masking

When a tone audibility is decreased or inhibited by the presence of another sound is called auditory masking. We refer as a maskee to the tone to be detected and as a masker the sound that creates the masking. We focus on the spectral frequency (or simultaneous frequency) because the present project aims to reduce the spectral frequency, the most common on music mixtures.

The **absolute threshold** is the minimum level on dB of a sound to be detected by the human auditory system in a silent ambient.

- The **minimum audible field** (MAF) is the curve of the absolute threshold versus the frequency measured by using loudspeakers at an anechoic chamber, taking as a level the theoretical level at the centre of the head of the listener.
- The **minimum audible pressure** (MAP) is the curve of the absolute threshold versus the frequency, measured by using headphones, taking as input level the level at the eardrum. This level can be measured by a small microphone or calculated using calibrated headphones and some acoustic formulas.

The minimum level in decibels necessary to hear a sound on the presence of another is called *masking threshold*, as described in previous sections. On the

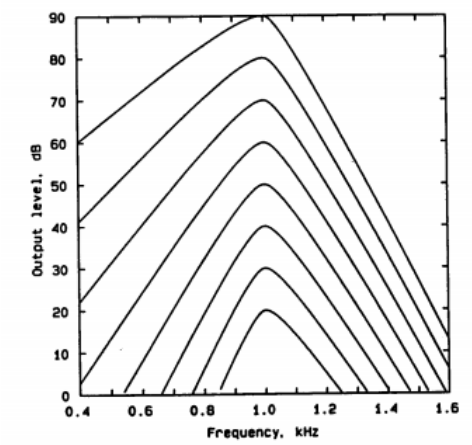


Figure 2.18: From [10]: 1kHz auditory filter output for inputs from 20 to 90 db SPL/ERB. While for intermediate leves the shape is almost symmetrical for lower and higher levels the shape, specially on the lower part, changes its slope, higher the input level smaller the slope.

other hand, computing the minimum necessary level to hear a sound when it is isolated is called *absolute threshold* or *threshold in quiet*, an example is shown in figure 2.19.

In previous sections, the experiments explained were intended to measure the auditory filter shape by moving or changing the masker over a fixed maskee, but to understand how a masker signal affects to other auditory filters the experiment has to be perform inversed, i.e. a masker on a fixed frequency and a variable maskee. This is known as *masking pattern* but it is commonly understood as the *excitation pattern* of the masker, i.e. the level of excitation is producing the signal (in this case the masker) on each characteristic frequency.

When representing the excitation patterns instead of using frequency a new variable is used that represents the perceived frequency, i.e. the distance or location in the basilar membrane. This new variable $z(f)$ is defined as

$$z(f) = 13 \arctan(\frac{0.76f}{1\text{kHz}}) + 3.5 \arctan\left((\frac{f}{7.5}\text{kHz})^2\right) \tag{2.12}$$

The previously mentioned critical band experiments show the critical bandwidth corresponds approximately 1.3mm and the basilar membrane is approximately 32mm, then there are $32\text{mm}/1.3\text{mm} \approx 25$ filters, but normally only 24 are used because the last one plays almost an irrelevant role. These filters are

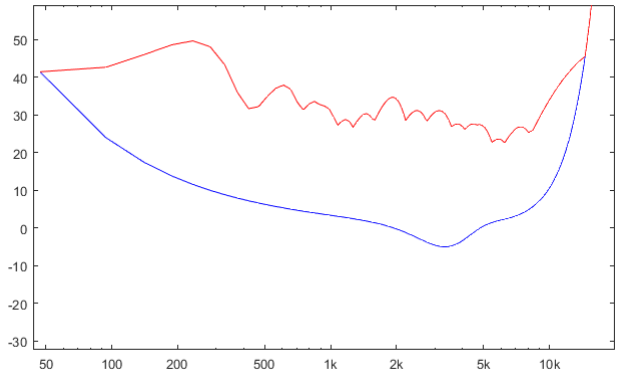


Figure 2.19: In blue colour: threshold in quiet curve (or absolute threshold) approximation given by the expression $3.64 \cdot (f_{kHz})^{-0.8} - 6.5e^{-0.6(f_{kHz}-3.3)^2} + 10^{-3} \cdot (f_{kHz})^4$. Red colour: example of a masking threshold from a signal. The signal-to-mask ratio is the difference between both.

shown at table 2.1

The most significant property of the bark scale is that the excitation patterns becomes almost linear, see figure 2.20, but the level linearity can not be assumed in this model, see figure 2.21. The present project evaluation is based on this model, but as what is being evaluated is the improvement of the masking on the same track the level variation is small, and therefore we can assume no error related to the level non-linearity.

When a masker is introduced in the auditory system it produces an excitation pattern over the system, which usually is assumed to be -3dB or -6dB when the masker is a noise or -14dB when it is a tone (the relation between the masker

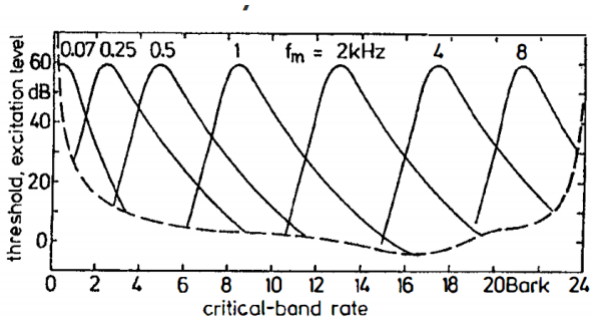


Figure 2.20: Excitation patterns in the bark scale

z (Bark)	f_c (Hz)	f_f (Hz)	Δf (Hz)	z (Bark)	f_c (Hz)	f_f (Hz)	Δf (Hz)
1	50	100	100	**13**	1850	2000	280
2	150	200	100	**14**	2150	2320	320
3	250	300	100	**15**	2500	2700	380
4	350	400	100	**16**	2900	3150	450
5	450	510	110	**17**	3400	3700	550
6	570	630	120	**18**	4000	4400	700
7	700	770	140	**19**	4800	5300	900
8	840	920	150	**20**	5800	6400	1100
9	1000	1080	160	**21**	7000	7700	1300
10	1170	1270	190	**22**	8500	9500	1800
11	1370	1480	210	**23**	10500	12000	2500
12	1600	1720	240	**24**	13500	15500	3500

Table 2.1: Bark scale critical bands.

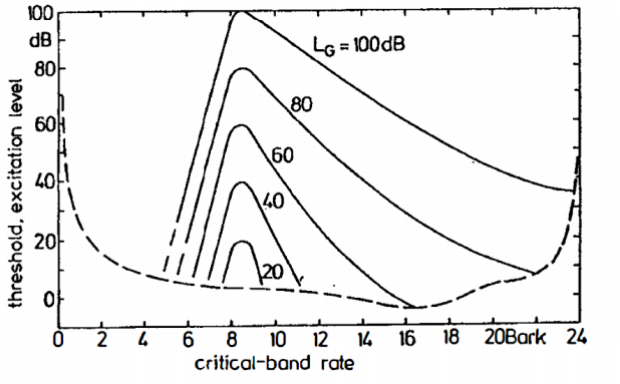


Figure 2.21: Level variation of a excitation pattern in the bark scale

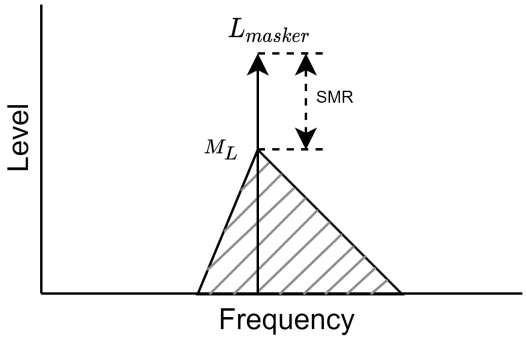


Figure 2.22: Masking pattern at level M_L from a masker with level L_{masker}

level and the excitation level is called signal-to-mask ratio). Then, depending on that excitation level a spreading of that excitation is produced over the near frequencies following a spreading function. In this case, the spreading function used is the Schroeder function:

$$10\log_{10} F(dz) = 15.81 + 7.5(dz + 0.474) - 17.5(1 + (dz + 0.474)^2)^{1/2} \quad (2.13)$$

With all these functions we can compute the masking threshold curve for a given masker. The problem is all these functions work with dB SPL, which corresponds to a physical magnitude so the masker spectrum has to be normalized on an SPL scale to use these functions. One way is to assume that the maximum digital amplitude (which equals $A = 1$ or $0dB_{FS}$) is equivalent to $96dB_{SPL}$. Then, summing $96dB$ to the dB_{FS} spectrum gives the SPL spectrum.

$$\mathrm{fft}_{SPL} = 96\mathrm{dB_{SPL}} + 20\log_{10}(|X(f)|)\mathrm{dB_{FS}} \quad (2.14)$$

As we are working with the digital Fourier transform we got to apply some corrections by dividing the spectrum by N which is the DFT size and also by multiplying the spectrum per two if we only take the real part of the spectrum.

The next step is to find the peaks of the spectrum and consider them as tone maskers and then apply the Schroeder spread function on each one of them, obtaining similar results as shown in figure 2.19. So now we only have to compare the masking threshold with the maskee spectrum (also performing the SPL transformation) to see which frequencies will be masked.

Chapter 3

MIXING PRINCIPLES

The present project aims to improve the mixing of a multitrack recording with an automatic algorithm, more precisely to reduce masking. But masking is not the only important aspect to take into account to make a good mixture, the proper use of the different audio processing is necessary to make good use of the algorithm described in this document. Bobby Owsinski [11] described six elements on the mixing: *balance, panorama, equalization, audio effects, dynamics and "interest"*. Nevertheless, after all these mixing processes a *mastering* process is usually done to enhance the general tone and dynamics of the final audio track.

3.1 Balance

This element is the most musical aspect of all the six elements and is related to the selection of instruments, frequency range, and rhythm of each instrument and selecting when an instrument sounds. It is commonly called *arrangement* and a good arrangement makes the mixing much easier.

Let us say you have an electric guitar or a piano playing on a similar frequency range, that would probably cause some "fighting" between both instruments. If both play almost the same rhythm and notes they would work as a "single" instrument and is not that harmful but if they do different notes and rhythm the audibility of each instrument will be hard. The same would happen with a solo guitar and a singer playing at the same time, as they are playing a similar role over the same frequency range they would have some "fighting". Nevertheless, there

are some methods to allow this kind of situation, for example situating each instrument in a different place of the panorama (discussed at section 3.2) or forcing with equalization on each track to have different predominant frequency areas.

Bobby Owsinski [11] organizes the instruments of a song in five different groups:

- **Foundation**: Here are the instruments responsible for the song metric, tempo, and feeling. Drums and bass are the most common foundation instruments.
- **Pad**: Pads are instruments that play long notes, usually giving harmony information and also filling the empty space of the mixture. Synthesizers and organs are typical pad instruments, but orchestra strings or guitar could also give a bad feeling.
- **Rhythm**: Those are the instruments that complement the foundation instruments, gives direction and motion to the song.
- **Lead**: Those are the melodies, by a singer, guitar, or any other instrument able to perform a lead role.
- **Fills**: Fills are used to fill the spaces left by the lead, sometimes done by the rhythmic instruments or sometimes are independent tracks or instruments.

Each of these roles must have a different mixing philosophy to enhance the proper characteristics of each instrument, for example, a lead instrument has to be prominent and the pad has to be subtle and not annoying.

But before entering the mixing stage, if the song arrangement is not properly done the mixing can be hard, so it is recommended to follow some tips to have a good arrangement.

- No more than three or four instruments in a single group usually work well. It is better to have fewer instruments or not having all of them sounding at the same time, switching between instruments on each part of the song so not all of them are sounding at the same time.
- Try to make each instrument of the same group work in a different frequency range, and rather than using equalization just change the octave of the notes or the actual notes of the instruments, i.e, playing another idea.

- Another way to distinguish better the instruments is using a panorama, but that does not always work because not all music is done to be reproduced on stereo equipment, for example, pop music or music for shop centres.

Owsinski also talks about levelling in the balancing process, which function is to set the proper volume of each track. The problem is that the level of each track will change a lot when they are equalized and compressed, so it is common to give each track an indicative levelling on that stage and the other stages of the mix correct the levels of each track if needed. Also is good practice to normalize and clean each track before processing so the gains on each track reflect better the actual level of the track.

3.2 Panorama

Panorama refers to the location of the audio on a loudspeaker layout. Music is almost always produced in a stereo layout which is conformed by two loudspeakers in 30 degrees from the frontal axis (see figure 3.1), but other layouts exist as Dolby 5.0, 7.0 and others. These last ones are commonly called "surround" layouts but are more used for cinema, and even in these layouts, the music is commonly placed on the two loudspeakers that correspond to the stereo layout.

Panning is performed by adjusting the gain for the audio on each loudspeaker, so the location of the audio source is located between the two loudspeakers and controlled by the individual gain of the sound on each loudspeaker.

To calculate the gain of the audio on each loudspeaker we can use the tangent law, given the angle for the loudspeakers respect the frontal axis θ_0, the angle of the source from the centre α and the gain relation of the left loudspeaker g_L and right loudspeaker g_R is

$$\tan\alpha = \frac{g_L - g_R}{g_L + g_R}\tan\theta, \tag{3.1}$$

then a second equation is used to set the *normalisation factor*. Two normalization factor expressions are depending if the sound sum is coherent or incoherent, but the incoherent sum is usually assumed because the sound behaves in that way for most of the middle and high frequencies. The normalization factor expression

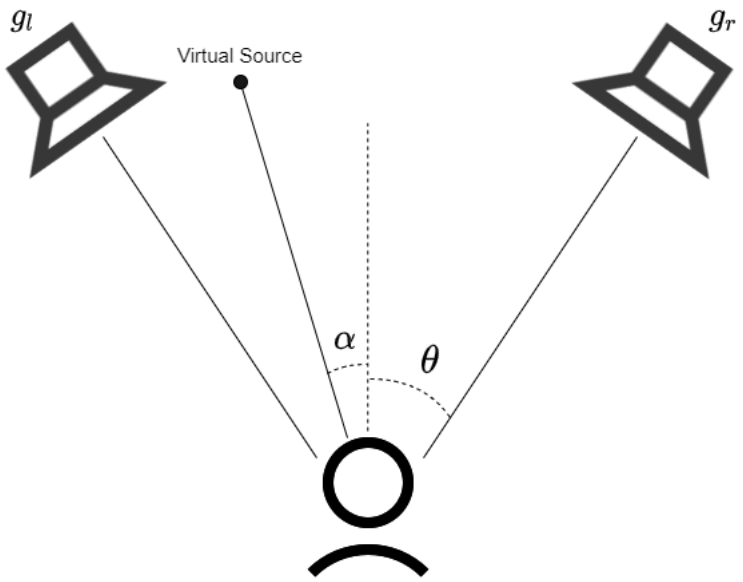


Figure 3.1: Panning scheme for the calculation of loudspeakers gain on a stereo layout.

for incoherent and coherent summation respectively are

$$g_L^2 + g_R^2 = 1, \tag{3.2}$$

$$g_L + g_R = 1. \tag{3.3}$$

Notice that this method only allows placing the source in a line between the two loudspeakers and does not allow moving it over the frontal axis.

Lead instruments are usually placed in the middle[1] because we perceive this effect like a fact of prominence and clarity (chapter four at [11]). For the same reason also drum kick and snare are usually are placed in the middle, but also because it is more musically coherent (Chapter four at [11]) and because as two additional microphones are used to record the drum cymbals and those microphones are hard panned[2] each one on a different speaker, it is better to have the central drum instru-

[1]Placing an audio in the middle means having an equal audio gain on each loudspeaker, perceptually the audio is perceived to be in the middle of both speakers.

[2]Hard panning consists on giving to an audio no gain on one loudspeaker and maximum gain on the other.

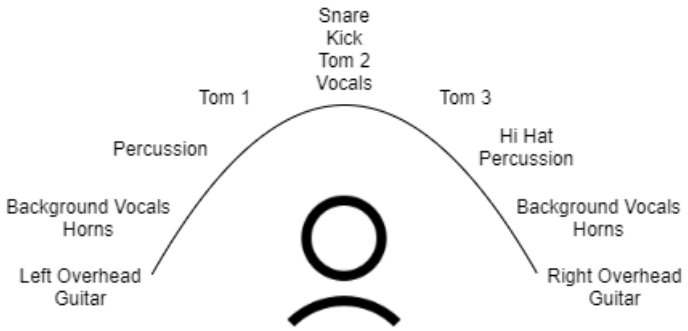


Figure 3.2: An example of panning landscape. Each song has different requirements, but there is a common tendency on locating instruments in a mixture.

ments (kick and snare) in the middle to avoid phase problems. Sometimes when phase problems occur due to having more than one recording of the same audio source, if these tracks are hard-panned the location of the source can be perceived outside of the loudspeaker (outside the θ_0 angle), generating *phantom images* of the sound.

An example of panning landscape is shown in figure 3.2. Generally, when an instrument is doubled distributing each track on different loudspeakers gives a feeling of "overture" or spacing. That is also why lead instruments usually are in the centre, lead instruments have to concise.

3.3 equalization

Equalization is one of the main keys to a good music mixture. It helps to clean up the audio or to make the audio sound deeper, clearer, or brighter. Good use of the equalization makes the difference between an amateur mix and a professional mix. Owskinski defines in his book [11] these reasons to make use of the equalizers:

1. "To make an instrument sound clearer and more defined". To clean up the frequency content, decrease undesired sounds presence on the audio clip...
2. "To make an instrument sound bigger". This is more an artistic choice, with equalization you can make an instrument sound better by enhancing the proper frequencies.
3. "To make all the elements on the mix fit together by juggling frequencies

so that each instrument has its predominant frequency range", i.e. to reduce masking between tracks. This idea is not always compatible with the previous one, that's why equalization is commonly done by listening to the track combined with other tracks of the mixture at the same time and not by isolating the track because otherwise you will make each track sound good but they won't work with the others, so it is better to make a sound each track as good as possible with the presence of the other tracks.

Reason number three is the aim of the present work: reducing the masking between tracks, but in this case, the algorithm does not implement any improvement respect the reason number two and that is why further processing could be necessary after using the masking reduction algorithm. Also notice that the masking reduction is done after some processes and that means previous processing can improve significantly the effectiveness of the algorithm. You must first clean up the track, add effects if needed, and make it sound "good" in its way using equalizers and compressors if needed too. Then, after all these processes, the automatic masking reduction will be applied.

On the table (3.3) is described some effects of increasing or decreasing specific frequency ranges. The consequences are described in a subjective and perceptive manner, meaning that could be a little ambiguous and it would be better if these effects could be heard but there are some common terms used to define the feeling of increasing or decreasing these frequency ranges.

equalizers are commonly grouped in two groups: *parametrics equalizers* and *graphic equalizers*. There are other classes usually found on audio equipment as shelving equalizers or high-pass/low-pass equalizers.

Graphic equalizer

These are equalizers that split the spectrum into n fixed bands and allow filtering the signal by controlling individual filters with gain sliders for each band. The "graphic equalizer" name is because on these equalizers you can "draw" the filter spectrum because on the horizontal axis are distributed the sliders and the vertical axis is the gain on each band (see figure 3.3). The filters centre frequencies are distributed exponentially, keeping the octave relation. A common graphic equalizer is the one with 10 bands, starting with 31.5Hz up to the 16kHz band so it covers all the audible spectrum.

RANGE	DESCRITION	EFFECT
16-60Hz Sub-Bass	Sense of power; felt more than heard	Too much makes the music sound muddy
60-250Hz Bass	Contains funamental notes of rhythm section; makes music far or thin	Too much makes the music boomy
250-2kHz Low Mids	Contains the low order harmonics of most instruments	Boosting 500-1kHz sounds hornlike; 1-2kHz sounds tinny
2kHz-4kHz High Mids	Contains speech recognition sounds like "m", "b", and "v"	Too much causes listener fatigue
4kHz-6kHz Presence	Responsible for clarity and definition of voices and instruments	Boosting makes music seem closer
6kHz-16kHz Brilliance	Controls brilliance and clarity	Too much causes vocal sibilance

Table 3.1: From Owsinski B. "Mixing Engineer's Handbook"[11]: An example of the effect of different frequency bands

These filters can be approached in two ways: summing filters (parallel system) or using cascade filters (serial system).

The first one is probably the most natural way to implement it by using pass-band filters on each band with one-octave bandwidth, then the resulting system filter will be

$$H_{eq}(w) = \sum_{n}^{N} H_n(w) G_n \tag{3.4}$$

being $H_n(w)$ the frequency response of filter n with gain G_n. Then multiplying the spectrum of the signal by that filter (or convoluting on time-space) will result in the desired filtered signal. The pass-band filters have to be carefully selected so when all gains are at 0dB it acts as a flat filter.

The other way to implement it is using the cascade system, and instead of using bandpass filters, using notch/peak biquadratic filters, explained later in this same chapter. As these filters have a magnitude equal to one outside the notch the proper way to compute the overall response of the system is cascading the filters,

Figure 3.3: The GEC digital equalizer plugin from Waves with 20 bands and a spectrum visualiser

i.e. the product of each filter response.

$$H_{eq}(w) = \prod_{n}^{N} H_n(w), \tag{3.5}$$

Parametric equalizer

The parametric equalizer has a more limited number of bands but the difference is that the centre frequencies can be selected. That means that you need fewer filters to equalize the signal as you can be precise with the frequency selection. Nevertheless, sometimes when too many bands are needed the graphic equalizer is still useful.

For that approach, a cascade biquadratic filter system is used as the filters have a flat response with a peak or notch at a certain frequency. There are several varieties of parametric equalizers, some allow you to change the filter type for each available filter, some have pre-defined filter types, some are a mixture of both... Another parameter on these equalizers is the Q factor, which changes the slope or bandwidth of the filter, depending on the filter type. Modern equalizers, especially the digital ones, usually also implement a spectrum visualizer where the filter responses and input signal spectrum are plotted.

Figure 3.4: A 5 band parametric equalizer from the classical SSL mixer plugin from Waves.

3.3.1 The biquadratic filters

The biquadratic filter (or biquad filter for simplicity) is a second order IIR filter with the following transfer function:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-1}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}. \tag{3.6}$$

Coefficients are commonly normalised so $a_0 = 1$.

These filters have low order and that makes it easier to design and faster to implement, it can be adapted in many different shapes very suitable for audio equalization and it is more stable than higher-order filters. Also if a steeper slope is needed you can cascade two biquad filters. These characteristic makes the biquad filters ideal for audio equalization applications.

In the figure 3.5 is shown the most common biquad filter designs, with the classic low-pass, high-pass and band-pass but also some useful filters as the notch, peak, high-shelf, and low-shelf filters.

To calculate the transfer function coefficients some parameters are needed:

- F_s: sampling rate.
- f_0: the "significant frequency". Depending on the filter type is the centre frequency, the shelf midpoint frequency, corner frequency...
- G_{dB}: the gain for that filter in decibels.

- Q, BW, and S: those are parameters related to the slope or width of the filter. Q and BW (bandwidth) are related by $Q = f_c/BW$. S is only for shelving filters, affecting to the slope.

Before calculating the final filter coefficients some intermediate variables are needed:

$$A = \sqrt{10^{\frac{G_{dB}}{20}}} = 10^{G_{dB}/40}$$
$$w_0 = 2\pi\frac{f_0}{F_s}$$

$$\alpha = \frac{sin(w_0)}{2Q}, \qquad \textit{(case Q)}$$
$$\alpha = \sin(w_0)\sinh\Big(\frac{\ln(2)}{2}\cdot \mathrm{BW}\cdot\frac{w_0}{\sin(w_0)}\Big), \qquad \textit{(case BW)}$$
$$\alpha = \frac{\sin(w_0)}{2}\sqrt{(A+\frac{1}{A})(\frac{1}{S}-1)+2}, \qquad \textit{(case S)}$$

Once these variables have been calculated in the following table 3.2 the equations for the filter coefficients are shown.

3.4 Audio effects

Audio effects are used for two reasons: to give spatial context to sounds (reverb and delays) or to give more personality to the sound (flangers, chorus, envelope filters...).

Reverb and delay can be used to create an "aural space" [11], to make the sound wider and bigger, and to move the track to the background (making the track more "blurred" or diffused). Sometimes this effect is used to create a similar room response to other tracks so they combine better.

Also, impulse responses can be used to achieve the same purpose. The difference between using an artificial delay or reverb from using impulse responses is

LPF	**HPF**
$H(s) = 1/(s^2 + \frac{s}{Q} + 1)$	$H(s) = s^2/(s^2 + s/Q + 1)$
$b_0 = (1 - \cos(w_0))/2$	$b_0 = (1 + \cos(w_0))/2$
$b_1 = 1 - \cos(w_0)$	$b_1 = -(1 + \cos(w_0))$
$b_2 = (1 - \cos(w_0))/2$	$b_2 = (1 + \cos(w_0))/2$
$a_0 = 1 + \alpha$	$a_0 = 1 + \alpha$
$a_1 = -2\cos(w_0)$	$a_1 = -2\cos(w_0)$
$a_2 = 1 - \alpha$	$a_2 = 1 - \alpha$
BPF (constant skirt gain)	**BPF (constant peak gain)**
$H(s) = s/(s^2 + s/Q + 1)$	$H(s) = (s/Q)/(s^2 + s/Q + 1)$
$b_0 = \sin(w_0)/2 = Q\alpha$	$b_0 = \alpha$
$b_1 = 0$	$b_1 = 0$
$b_2 = -\sin(w_0)/2 = -Q\alpha$	$b_2 = -\alpha$
$a_0 = 1 + \alpha$	$a_0 = 1 + \alpha$
$a_1 = -2\cos(w_0)$	$a_1 = -2\cos(w_0)$
$a_2 = 1 - \alpha$	$a_2 = 1 - \alpha$
PEAK	**NOTCH**
$4\ H(s) = (s^2 + 1)/(s^2 + s/Q + 1)$	$H(s) = (s^2 + s(A/Q) + 1)/(s^2 + s/(AQ) + 1)$
$b_0 = 1$	$b_0 = 1 + \alpha A$
$b_1 = -2\cos(w_0)$	$b_1 = -2\cos(w_0)$
$b_2 = 1$	$b_2 = 1 - \alpha A$
$a_0 = 1 + \alpha$	$a_0 = 1 + \alpha/A$
$a_1 = -2\cos(w_0)$	$a_1 = -2\cos(w_0)$
$a_2 = 1 - \alpha$	$a_2 = 1 - \alpha/A$
LOW SHELF	**HIGH SHELF**
$H(s) = A\frac{s^2 + \frac{\sqrt{A}}{Q}s + A}{As^2 + \frac{\sqrt{A}}{Q}s + 1}$	$H(s) = A\frac{As^2 + \frac{\sqrt{A}}{Q})s + 1}{s^2 + \frac{\sqrt{A}}{Q}s + A)}$
$b_0 = A((A+1) - (A-1)\cos(w_0) + 2\sqrt{A}\cdot\alpha)$	$b_0 = A((A+1) + (A-1)\cos(w_0) + 2\sqrt{A}\cdot\alpha)$
$b_1 = 2A((A-1) - (A+1)\cos(w_0))$	$b_1 = -2A((A-1) + (A+1)\cos(w_0))$
$b_2 = A((A+1) - (A-1)\cos(w_0) - 2\sqrt{A}\cdot\alpha)$	$b_2 = A((A+1) + (A-1)\cos(w_0) - 2\sqrt{A}\cdot\alpha)$
$a_0 = (A+1) + (A-1)\cos(w_0) + 2\sqrt{A}\cdot\alpha$	$a_0 = (A+1) - (A-1)\cos(w_0) + 2\sqrt{A}\cdot\alpha$
$a_1 = -2((A-1) + (A+1)\cos(w_0))$	$a_1 = 2((A-1) - (A+1)\cos(w_0))$
$a_2 = (A+1) + (A-1)\cos(w_0) - 2\sqrt{A}\cdot\alpha$	$a_2 = (A+1) - (A-1)\cos(w_0) - 2\sqrt{A}\cdot\alpha$

Table 3.2: List of equations for the computation of biquadratic filter coefficients

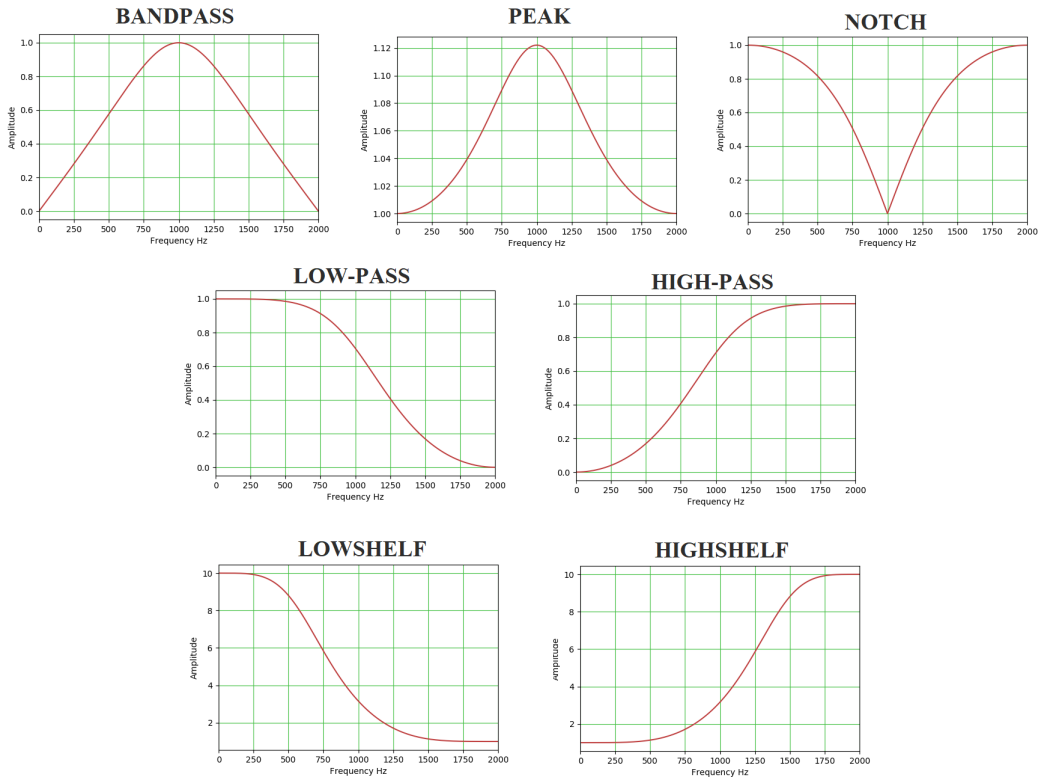


Figure 3.5: The most common biquadratic filter shapes.

that the impulse response loader tends to achieve a more realistic room feeling, but that does not mean it is the proper feeling on the context. Adding a small artificial reverb and delay can achieve the desired widening effect on the track, but if the purpose is to create a realistic room space sensation an impulse response loader could be a better option. This impulse response loaders just makes a convolution to the signal with the impulse response selected (or product in frequency space):

$$\begin{aligned} y(t) &= x(t) \circledast h(t) \\ Y(w) &= X(w)H(w) \end{aligned} \tag{3.7}$$

Also, the delay can be used as an artistic effect, i.e. the delay is not intended to create an ambient but intended to be heard almost the same as the direct sound. In that case, is common to set the delay time respect the song tempo, so based on song beats per minute ratio, let's say 60bpm for example, and the time signature, let's say a $4/4$ meter,[3] if we want a delay about an eighth note the delay time will be

$$t = \frac{1}{60\text{beats/min}} \cdot \frac{60\text{s}}{1\text{min}} \cdot \frac{1\text{beat}}{1\text{quarter}} \cdot \frac{1\text{quarter}}{2\text{eigth}} = 0.5s. \tag{3.8}$$

Synchronizing the delays also helps to make them more compact.

The previously mentioned flanger effect is the result of having two copies of the same signal but one delayed respect the other by a Δt time that changes over the time periodically. Having two exact signals, one delayed respect the other creates notches in the spectrum as shown in figure 3.6. The notches occur when the delay time is exactly half of the signal period plus a multiple of the period, creating destructive interference. That is

$$f_n = \frac{2n+1}{n\Delta t}. \tag{3.9}$$

Then changing the Δt value over the time makes all the notches to change on frequency over the time, for example by the expression $\Delta t(t_0) = \Delta t_0 \sin(t \cdot f_{fl})$ with Δt_0 as the initial time difference and f_{fl} the rate you want the flanger to change.

[3]The denominator indicates the note subdivision used and the numerator means the number of notes in a bar, for example, the 4/4 means that one beat equals a quarter note and there are four quarter notes in a bar

Common delay times for flanger are up to 20 milliseconds, and when that delay is higher than that the effect is commonly known as the chorus. Nevertheless, the chorus delay time is not modulated but fixed and it also usually has small pitch variations creating a similar effect than multiple instruments playing the same notes, similar to a choir effect.

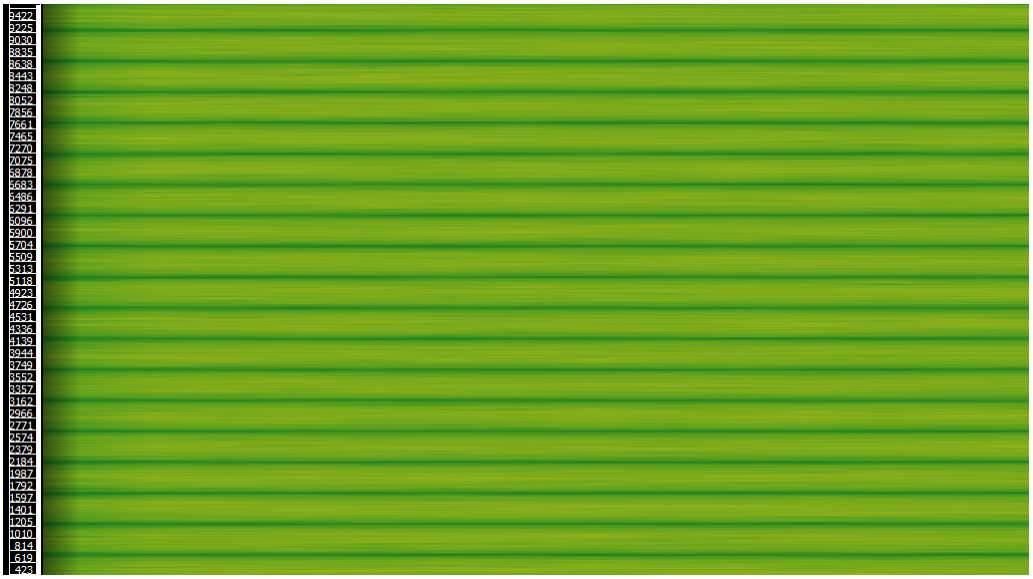


Figure 3.6: Spectogram of a two identical white noise signals delayed by 0.002s, with notches at 250, 750, 1250...

3.5 Dynamics

Dynamics are related to the dynamic range and loudness/level of the signal. A *gate* is a simple dynamic effect that enables the audio output when the input is above a certain level threshold, a *limiter* behaves in reverse when the input passes a certain level threshold the output level will be clamped at the threshold level. Finally, the *compressor* is similar to a limiter but it does not clamp the output to the threshold level but instead, it applies a ratio compression, for example, an 8:1 compression means that for a signal over the level threshold, for each 8dB in the input only 1dB will come out. We can say that a compressor with a ratio ∞:1 acts as a limiter.

These dynamic controllers usually have more parameters, for example, the attack (ms) that measures how fast the compressor starts acting when the level

threshold is passed, or the release (ms) that measures the time the compressor stops acting when the signal goes above the threshold. Also, as these processes reduce the level at the output they usually have an output gain control so you can adjust the level.

Some of the reasons to use a gate are to avoid the background noise or the sound of other instruments on a recording or to avoid feedback on live performances. The compressors are used to reduce the dynamic range so you can hear from the loudest sound to the quietest sounds, for example on a singer recording. Also, some instruments create small "spikes" that can be annoying, as a funk guitar that has to sound even or the bass that depending on the note is played more or less low-frequency content is emitted and then we want to keep the low end more or less constant. Also, the compression can be used as an artistic effect, with the proper compressor and parameters you can make a track sound more exciting, powerful, or aggressive, for example on rock voices.

Compressors are used almost always with an equalizer, but the order is relevant to the final result. Compressing a sound emphasizes the predominant frequencies, so if you boost with an equalizer that frequencies before compressing you will lose all the other frequencies as these frequencies are "doubled" emphasized. On the other hand, sometimes you want to cut some frequencies before compressing, for example, an electric guitar with an excess of low-frequency content that you do not want the compressor to be affected by these frequencies. Also equalizing before the compressor is vital when you have noises in the audio clip if those noises are not cleaned up the compressor will act oddly or will emphasize that noises. Also, another important fact to take into account is that compressors are non-linear, so it can create artifacts as the combination tones previously mentioned.

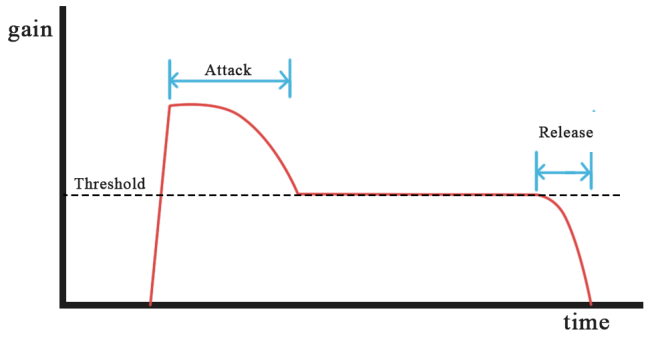


Figure 3.7: An example of a compressor behaviour on a signal (red) with "attack", "release" and "threshold" parameters.

3.6 Interest

Ed Seay said on Owsisnki book [11]: "*The tough part, and the last stage of a mix, is the several hours it takes for me to make it sound emotional and urgent and exciting so that it's just not a song, it's a record. It's not making it just sound good, it's making the sound like an event. Sometimes that means juggling the instruments or the balances or adding some dynamics help. That's the last stage of when I mix, and that's the part that makes it different or special*".

These words are a good summarise of giving interest to a mixed means. My little experience on mixing music taught me the importance of giving emotion to the track and not just make all the tracks work together and sound "good", but also create movement on the song by being meticulously precise with level amortization on tracks so each instrument is present the way it has to be at the moment it has to be or to choose the proper feeling for each instrument depending on the song context. By far one of the hardest processes on the mix.

That means that besides making a "correct" mix (understanding correct as there's no track fighting, you can hear everything and each instrument has a proper tone) the mix has to be emotional, so artistic decisions have to be made. For example, as mentioned on the compression section it can be used as an effect, so adding an aggressive compression on a hard rock vocalist track will help a lot to the musician to transmit the aggressive voice, or for example, creating a proper delay and reverb ambiance on a calm song or enhancing the rhythmic sounds on a funk song will create a better mix. The job of an audio engineer is not making a song recording sound good, it is to make the song recording transmit whatever the musicians want to transmit with the song.

3.7 Mastering

Mastering is the last stage of a song recording. It is considered out of the mixing process as it is commonly made by another audio engineer specialized on that topic. Bobby Owsinski [38] defines the mastering process as "the process of turning a collection of songs into a record by making them sound like they belong together in tone, volume, and timing (spacing between songs)".He talks about the mastering as a process to do over an entire album, but nowadays as the concept of singles is trending, mastering is applied to a single audio track too.

Mastering adapts a song track or group of song tracks to have the proper tone (not excessive bass, not too strident...) and similar between them. and also increases the overall loudness. This last reason is very important because we naturally perceive that a record sounds better when it has more loudness. Nevertheless, the tone adjustment is also important to assure that the conversion from the high-quality studio monitors to any other equipment is properly done. For the same reason, mastering can be adjusted to fit the song on a digital media (usually reproduced by mobiles or cheap loudspeakers), night clubs (usually have big subwoofers and big loudspeakers), or a shop centre audio system (usually non-stereo systems).

Another good reason to master the song by another engineer is to have a secondary objective opinion of the mixing. When a song mix has finished the ear of the engineer will be used to the mixing and sometimes small errors are not noticeable, and giving the final track to the master engineer adds that extra opinion that can extremely boost the quality.

An additional process made by the mastering engineer is to down-mix the song recording to the proper audio format. Nowadays that is quite easy but back in the days, that process was really hard because the mastering engineers had to transform from the magnetic audiotape to vinyl, and that involved expensive equipment and a "handcraft" process. To summarise, master engineers are audio engineers that know how a good audio track has to sound in terms of tone and dynamics, so they make the proper corrections to the audio tracks to make them sound as big, loud, and tonally balanced as possible.

Also mention that the compression at this stage is not usually made on the whole spectrum, sometimes it is done on individual bands and sometimes some further process is made, for example harmonic exciters.

Chapter 4

SYSTEM OVERVIEW

The present project is based on the algorithm described by Joshua D. Reiss [1]. He based the algorithm on professional audio engineer practices instead of auditory models. On one hand, previous works presented inaccurate results when using more realistic auditory models [22], and on the other hand the aim of the algorithm was to run real-time.

He found some similarities on how audio engineers [11][39][40] mix songs, and based on them he concluded:

1. It is better to attenuate the masked frequencies on the masker than boosting the frequencies on the maskee.
2. There are two classes of frequency regions: essential and non-essential. The essential frequencies are most likely to be the frequencies with higher magnitude and the non-essential the ones with lower magnitude.
3. Tracks with frequency regions covered by other tracks can be reduced.

In consequence, in his model the masking effect occurs when the masker magnitude is higher than the maskee magnitude on a certain frequency region and that frequency region is non-essential for the masker. That model does not contemplate all the masking cases, as masking also could occur when the frequency band is essential for the masker and when the masker magnitude is lower in that band. Nevertheless, applying the real auditory model could not assure the real-time constrain as it has to apply several corrections explained in chapter 2 related to the

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
f_c (Hz)	31.25	62.5	125	250	500	1000	2000	4000	8000	16000
f_0 (Hz)	0	31.25	62.5	125	250	500	1000	2000	4000	8000
f_f (Hz)	62.5	125	250	500	1000	2000	4000	8000	16000	22000

Table 4.1: Centres and boundaries used for the band-pass filters on the analysis stage. The same filter centres are used on the filtering stage. The first row represents the centre frequency f_c, f_0 is the left boundary of the band-pass filter and f_f the right boundary.

outer and middle ear frequency response and the non-linearities of the auditory filter.

Figure 1.1 shows the block diagram of the algorithm, where two main stages are described, the analysis stage and the filtering stage. In the analysis stage, the n input tracks are analyzed to extract the features to detect the masking, then on the filtering stage the tracks are equalized on the frequency bands where masking has been detected. For both analysis and filtering stages, a fixed centre frequencies of the filters are used. The goal is to cover almost the whole human audible spectrum respecting the logarithmic processing that it performs internally. For that reason octave filter centres has been used, as show at table 4.1.

This system can be performed using the whole audio length or using audio frames depending on the implementation. For example, for the online implementation using an overlap-add frames method is required, meaning that for each frame, all the analysis and filtering is computed.

4.1 Analysis stage

The analysis stage is performed by a *feature extraction* algorithm, where the magnitudes and rankings of the bands are computed on each track, a *masking detection* stage, where the bands with masking are detected, and a *masking selection* stage, where for each band the maximum masking amount is selected.

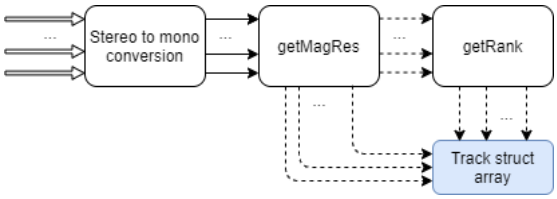


Figure 4.1: Analysis stage block diagram. All tracks are converted to mono and the magnitude and rankings of the frequency bands are computed and saved together on a data structure.

4.1.1 Feature extraction

The feature extraction is the first step on the analysis stage and it is performed by a mono to stereo conversion, a magnitude extraction, and a band ranking, as shown in figure 4.1.

All the analysis is performed with mono audios because it is assumed that the same masking is caused by each channel, so the stereo audios are down-mixed to mono and the same filtering will be applied to each channel.

After the stereo to mono conversion, all the audios are sent to the *getMagRes* function. That function divides each track into **aF** octave bands using band-pass filters, meaning that the centre of the next band is the double of the previous one. Given the maximum analysis frequency f_m and the number of bands N, the first frequency centre can be found by the following equation:

$$f_0 = \frac{2f_{max}}{3 \cdot 2^{N-1}}. \tag{4.1}$$

As we are using digital audio signals, by following the Nyquist–Shannon theorem to avoid aliasing, we know that the maximum frequency in our signal without producing aliasing will be half of the sampling frequency. Additionally, we will use $aF = 10$ analysis bands for the analysis stage, resulting in $f_0 = 31.25\text{Hz}$ as the first frequency centre and $f_{N-1} = 16\text{kHz}$ when using a sampling rate $f_s = 48\text{kHz}$. This function computes the root mean square of each band, returning aF magnitudes for each track, and that data is sent to the *getRank* function.

The "getRank" function ranks the band magnitudes of a track. The band with highest magnitude will be ranked as 1, while the lower one will be ranked with the number $aF = 10$. Both magnitudes and rankings of each track are saved on a

data structure together.

4.1.2 Masking detection

As mentioned before, the masking model used by J. Reiss [1] is based on professional audio engineer practices instead of auditory models. It is a good approximation and it is much faster than modelling the real behaviour of the ear and auditory system. It only needs, for each track, the magnitudes and rankings of the bands, which are fast to compute.

The expression 4.2 is used to compute the amount of masking $M_{AB}(f,t)$ that the masker A is producing to the maskee B at a given frequency f and time t. Masking is detected only if the frequency is considered as essential for the maskee (R_B less or equal than R_T) and non-essential for the masker (R_A greater than R_T). R_T is the maximum ranking score for a band to be considered as essential. When masking is detected, the amount of masking is computed as the difference of magnitudes of the signals on that frequency.

$$M_{AB}(f,t) = \begin{cases} X_A(f,t) - X_B(f,t), & if \;\; R_B(f,t) \leq R_T < R_A(f,t) \\ 0 & else. \end{cases} \tag{4.2}$$

This equation is evaluated for each pair of tracks, assuming that each track can be both a masker and a maskee. Also, instead of using a continuous frequency f, it uses the frequency bands, meaning that the equation is evaluated aF times. This results on a three-dimensional matrix, where the first dimension is the masker, the second one the maskee, and the third one is the band. All cases where the maskee equals the masker ($A = B$) are skipped, assigning all values to zero for that cases.

4.1.3 Masking Selection

On the previous stage we obtained the amount of masking caused to the other tracks by each masker. This implies that one masker could produce masking on the same band on different tracks at the same time. On each masker, the highest masking amount is selected on each band by the masking selection stage. The table 4.2 shows an example of the algorithm for track 2. Notice that the row

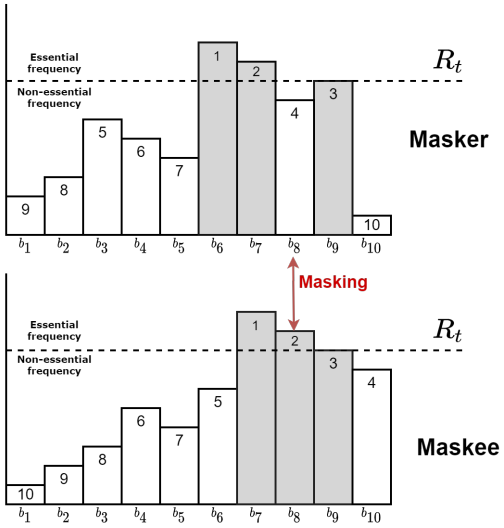


Figure 4.2: Example of how masking detection is performed by the algorithm. At frequency band b_8, the masker (upper graph) is producing masking over the maskee (lower graph) as the band is non-essential for the masker and essential for the maskee.

corresponding to the maskee 2 is filled with 0, as we are considering it as the masker.

This process is performed for each masker, reducing the three-dimensional matrix from the previous stage to a $N \times aF$ matrix, N being the number of tracks. The resulting matrix represents the maximum amount of masking that each masker is producing on each band. After that, it selects the highest nF amount of masking for each masker.

When using frames instead of the whole audio, smoothing between frames is required to avoid artefacts. Using no smoothing can result in abrupt and undesired changes between frames. For that reason, an Exponential Moving Average (EMA) filter is used. This filter is a weighted arithmetic mean with weight coefficients given by a negative exponential function, as shown in figure 4.3. It can be implemented easily by the following IIR filter:

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1], \quad 0 < \alpha \leq 1 \tag{4.3}$$

This equation is easy to implement as it only needs the previous value and the new one. The α value sets the influence of the new value to the output, so the

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Track 1	0	0	0	**10**	0	5	0	**6**	0	0
Track 2	0	0	0	0	0	0	0	0	0	0
Track 3	**6**	0	0	4	0	**7**	0	2	**3**	0
Max masking	6	0	0	10	0	7	0	6	3	0

Table 4.2: Example of masking selection considering track 2 as a masker. Bold numbers represent the selected masking amount for that column (frequency band). The shaded row corresponds to the masker, filled with 0 as it can not be a maskee simultaneously.

lower the $alpha$ value the faster the change between frames. Considering a frame size $N = 1024$ samples, $\alpha = 0.99$ gives a good balance.

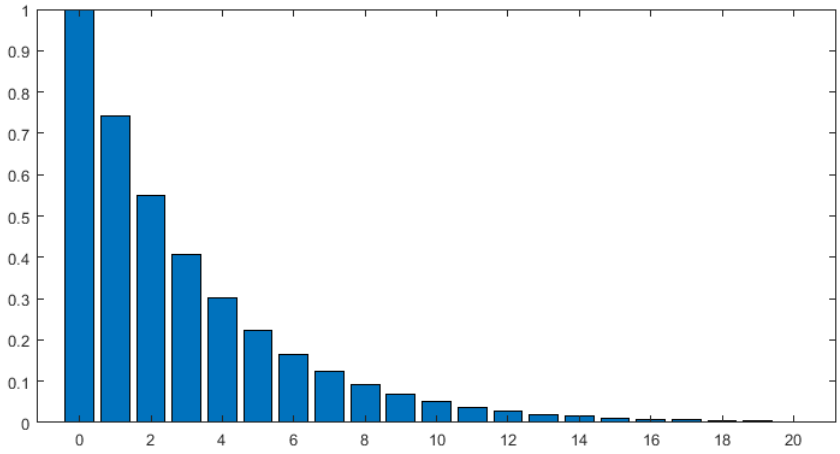


Figure 4.3: Example of Exponential Moving Average coefficients, following a negative exponential function.

4.2 Filtering

This is the last step on the system, where all the track features are received from the analysis stage and used to filter each track with the proper filter. Notice that, to increase the efficiency of the system, all bands with small masking (e.g. 0.1) are skipped as they do not make any difference. The filters used for that purpose are the biquadratic filters, explained previously in section 3.3. These filters are low

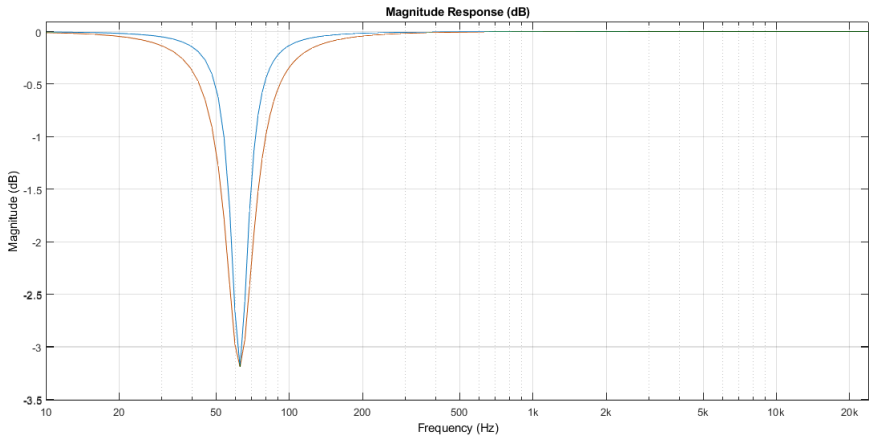


Figure 4.4: Blue curve represents a peak filter at 63 Hz with $Q = 5$, and the orange curve represents the same filter with $Q = 3$.

order IIR filters, causing them to be faster to apply and design, and also they have a proper shape to equalize.

For each track, the nF filters are designed using the previously computed masking amounts as gains. The masking amount has to be turned negative, as we want to decrease the magnitude instead of boosting it. The Q factor used varies in the implementation, but $Q = 2$ gives the most stable results.

The filters are applied to the track using a cascade system, i.e., applying the filters one after the other to the signal. If the track is stereo, the same filtering will be applied on each channel. J. Reiss [1] performs a mix-down of all the tracks, but in this project all individual tracks are kept separated.

4.3 Implementation

Two variations of the algorithm have been implemented: an online version (real-time processing), and an offline version (non real-time processing). The main difference is that the online version performs the analysis and filtering on individual audio frames that are overlapped, while the offline version averages the magnitudes extracted from non-overlapped frames and then, performs the remaining processes using the averaged magnitudes for the whole audio. Both versions have been tested and implemented using Matlab, but an additional online VST plugin

has been implemented.

Based on the J. Reiss [1] implementation, I have introduced several parameters to adjust the algorithm behaviour. The parameters that the user can adjust are the following:

- **Q**: It is the quality factor of the filters. Higher Q implies narrower filter, as shown at figure 4.4. It is proposed using $Q = 2$ or $Q = 3$.
- **nF**: It is the amount of filters used for the equalization. It has to be smaller than the analysis filter amount aF. The suggested number of filters is $nF = 5$, a balance between efficiency and precision.
- **S**: This term scales the gain of the equalizer filters. Modifying this paramete,r the user can intensify or diminish the effect of the equalization. By default this parameter equals 1.
- **"Eq. Normalization"** restricts the maximum amount of filtering on each band, dividing the filter gains by the amount of tracks producing masking on the band. This does not increase the efficiency of the algorithm but it produces a more natural result. A clear case to use this option is when there is a track with high frequency bands as essential (for example a voice track) and all the other tracks have these bands as non essential. This could result in an excessive filtering on the high frequency contain. This option can be activated using the "Eq. Normalization" boolean.
- **Weighted mean**: that option is only available for the offline versions. As mentioned before, the magnitude extraction is performed in individual audio frames and then they are averaged. That option allows to use a weighted mean instead of a arithmetic mean. The weights are the RMS of each individual frame respect the sum of all the RMS, giving more relevance to the frames when the instrument is sounding and less relevance when there is noise from other sound sources.

Another variables are set to a default value, as $R_T = 3$ (used on equation 4.2), $aF = 10$ or the frame size $N = 1024$ samples. Additionally, the user can specify which audio region is used.

J. Reiss also implemented a maximum gain constrain for the equalizer filters, but after some tests, I concluded that it is not necessary as it reduces the performance and similar effect can be achieved adjusting S and using the "eq.

Name	Run Type	Autonomy	User Parameters
Matlab Aut.	Offline	Fully	Q, S, eq.Norm, w. mean
Matlab Semi	Offline	Semi	Essential bands selection Q, S, eq.Norm, w. mean
Matlab Real-Time	Online	Fully	Q, S, eq.Norm
VST Real-Time	Online	Fully	Q, S, eq.Norm

Table 4.3: All implementations and available parameters. The autonomy reflects the human interaction requirement of the system, meaning that fully-autonomous implementations works without any interaction and the semi-autonomous needs some manual interaction.

normalization" option. All the implemented versions are shown at table 4.3, two offline versions using Matlab, and two online versions using Matlab for testing and another using C++ as final (VST format). All implementations source codes can be found in [41].

4.3.1 Offline

Two offline versions have been implemented using Matlab, a fully-autonomous and a semi-autonomous version. For the offline approach the audios are analyzed with non-overlapped frames of size $N = 1024$. After the analysis, all the magnitudes are averaged for each track and then the ranking is performed using the averaged magnitudes. This implementation has the "weighted mean" option that helps the tracks that are not always present, for example, a voice track. When the main sound source is not present, the system will analyse the sound coming from the other sounds captured by the microphone (noise, bleeding[1]...), causing inaccurate results.

The fully-autonomous offline version uses the equation 4.2 with rankings computed by the "getRank" function, but the semi-autonomous approach uses a manual essential/non-essential classification system. At the beginning of the program execution, the user is asked to introduce the essential frequency bands of each track. Using that information, the "Masking detection" algorithm adapts the equation 4.2 to detect masking when a certain band is non-essential for the masker and essential for the maskee. The masking selection and the filtering remains the same for both offline versions.

[1] "Bleed" is a term used to define the not desired incoming sound on a microphone, usually others instruments of the same room.

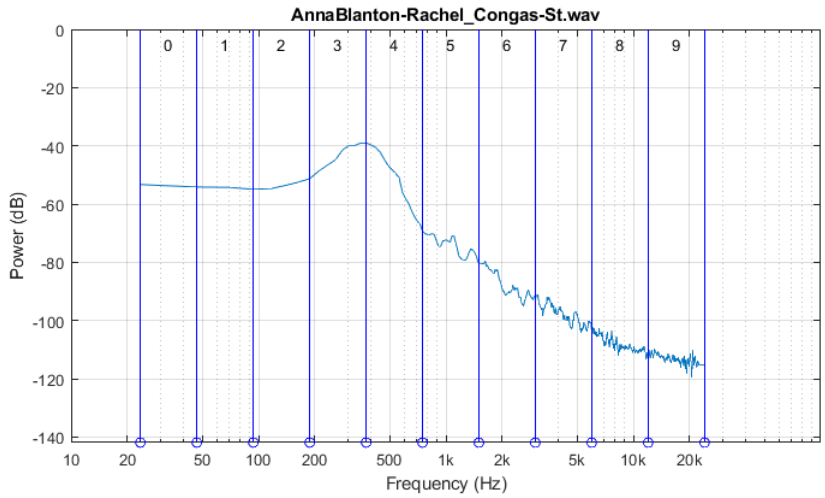


Figure 4.5: Graph showed to the user in the semi-autonomous approach in order to select the essential bands.

The spectrum of the signal and the analysis bands are showed to the users (figure 4.5) and it is asked to introduce the first Rt most important bands for that track. The user does not necessarily has to mark as essential the bands with higher magnitudes, that approach allows the user to distribute the essential bands, for example, to split the low frequencies between the bass and the drum kick.

After the equalization, mix-down copy and the filtered tracks are saved on disk. Additionally, the same process is done for the raw audios because the user could have selected only a portion of the song. Both mix-down files (filtered and raw) are set to the same loudness level with the maximum possible amplitude before one of them starts clipping.

The fully autonomous uses $Q = 3$, $\mathrm{nF} = 5$ and $S = 2$ as default and "eq.normalization" and "weighted mean" disabled. The semi-autonomous uses $Q = 2$, $\mathrm{nF} = 5$ and $S = 2$ as default with "eq.normalization" disabled and "weighted mean" enabled. Those are the parameter values that give better results for each version, but some adjustments could be done if needed.

4.3.2 Online

The online version has been implemented with Matlab to test the algorithm. Then, using JUCE along with the VST C++ framework a DAW plugin has been created.

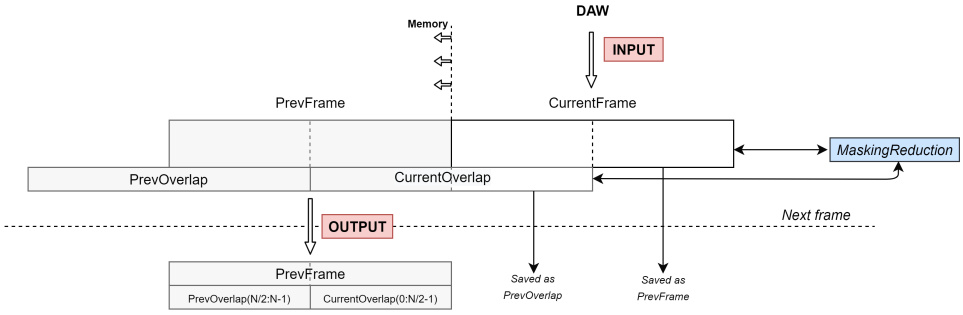


Figure 4.6: Overlap-add block diagram. The right side corresponds to the current frame, and the left side corresponds to the saved frames. The output on that iteration is the processed previous frame with the overlapped frames. The vertical axis represents the data flow of the VST.

This version performs the analysis and filtering on each frame individually. The frames are smoothed using overlap-add and using EMA filters on the equalizer gains. The VST is the DAW slave, meaning that it depends on what the DAW sends to the plugin. The problem is that the DAW sends to the plugin non-overlapped frames and it expects to receive also a non-overlapped frame. For this reason, the plugin can not perform the overlap-add the same way, as it would need the next frame to perform the overlap-add.

The solution is to be always N samples late, so the plugin can save the previous frame in memory and perform the overlap-add as shown in 4.6, returning the previous frame with the corresponding overlapped frames.

To perform the overlapping of the "PrevFrame" we need the "PrevOverlap", that is saved in memory from the last iteration, and the "CurrentOverlap" frame, that can be constructed using the "PrevFrame" and the "CurrentFrame". Once all the required frames have been computed, the frames are processed with the "MaskingReduction" algorithm and then the overlap-add is performed.

The required saved data to perform this overlap-add method is the processed "PrevOverlap" and "PrevFrame", and the raw "PrevFrame" to be able to construct the "CurrentOverlap".

Another difference with the offline implementation is that the analysis filters coefficients are pre-computed, as the same band-pass filters are used on each frame.

I used JUCE to create the VST plugin. It is a software to design and cre-

Figure 4.7: Visual interface of the VST plugin

ate multi-platform audio software, including graphic tools. It makes it easier to create a VST that can be compiled on Windows or iOS. JUCE allows to create several plugin formats using the same source code, as AAX, RTAS, or to create a standalone version. Four graphical controllers have been added to adjust the parameters (Q factor, the gain S, toggling the eq. normalization and an output gain).

On this implementation, the default parameters are hard to set, as the analysis is performed on each frame and it makes the algorithm very sensitive to changes. Nevertheless, it seems to perform better using $Q = 4$ and $S = 2$, but as the plugin is used manually for users, they can adjust the parameters based on the song necessities or even automatize the parameters on the different parts of the song.

4.4 Evaluation criteria

The implementation of the criteria explained on section 2.5 is performed after all the system filtering. It computes the amount of masking all the tracks together are causing to a single given track. This process is repeated for each track with both raw and filtered versions. Then, the masking reduction ratio (MRR) for a track A can be computed as:

$$\mathrm{MRR}(A) = \frac{M(A_{filtered})}{M(A_{raw})}. \tag{4.4}$$

Also, if all the tracks MRR are averaged we obtain the MRR of the song. However, the individual ratios are useful information as not all tracks have the same relevance, so we also have to take into account the MRR of each track.

When computing the masking for a track, the masker and the maskee signals are normalized so the loudest one is at -10dB FS of loudness and the other track is increased the same dB amount (does not mean that is set to -10dB FS). Then it is translated to the SPL scale as explained in section 2.5.

Chapter 5

EVALUATION

5.1 Procedure

A subjective and objective evaluation have been used for 6 songs, shown in table 5.1. Two tests has been made in order to understand the behaviour and efficiency of the algorithm.

The first test is intended to evaluate the effect of editing the audio clips before the masking reduction algorithm and also the effect of the "eq. normalization" and the "weighted mean" options. The fully-autonomous version has been used with default parameters (see table 5.2).

The second test is intended to evaluate the subjective and objective understandability of the song using the three type of implementations (both offline implementations and the online Matlab implementation). The parameters used on this test are adapted on each song following two conditions: the computed MRR must be lower than 1, and the resulting output must sound as natural as possible. When the algorithm is used aggressively, the masking decreases, but the resulting

Song nº	Artist	Title	Genre	Tracks	Duration
1	Angela Thomas	Milk Cow Blues	Country	7	29s
2	Bolz & Knecht	Summertime	Sax/Guitar duo	2	33s
3	Dino On The Loose	Queen's Light	Jazz Fusion/Electronic	5	33s
4	The Penniless Wild	Closer	Indie Rock	8	26s
5	Megan Slankard & Alex Wong	There Are No Shadows In L.A	Indie Scoustic	3	10s
6	Anna Blanton	Rachel	Acoustic Pop	6	17s

Table 5.1: Song list used for the evaluation.

Offl. Auto.					Off. Semi A.					Online - VST					Online - Matlab				
Q	**S**	**nF**	**eqN**	W_{avg}	**Q**	**S**	**nF**	**eqN**	W_{avg}	**Q**	**S**	**nF**	**eqN**	α	**Q**	**S**	**nF**	**eqN**	α
3	2	5	false	false	2	2	5	false	true	4	2	5	false	0.99	4	2	5	false	0.99

Table 5.2: Default parameters for the different implementations. Both online versions share the same parameters as they only differ on the programming language.

audio feels unnatural, so the goal is finding balance. The best parameters following this rules are shown on table 5.3.

The subjective evaluation has been performed via Google Form. The subjects are asked to evaluate 4 versions of the same song, located in a google drive folder. These 4 versions corresponds to the three implementations (Matlab Online, semi-autonomous and fully autonomous) and the raw audio, and then the files are renamed as "a", "b", "c" or "d" randomly. Then, it is asked to rate the understandability of each version of the song using a value between 1 and 10, with a value less or equal to 2 and a value greater or equal to 9. With this constrain, the results are more normalized for each subject. Only four songs have been used to reduce the test duration.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**
Matlab Online	Q=4 S=2.5	Q=4 S=2	Q=2 S=1	Q=2 S=1.5	Q=1.5 S=4.8	Q=5 S=4
Matlab Automatic	Q=3 S=2 F,F	Q=3 S=3 TT	Q=2 S=2 TT	Q=3 S=2.5 F,F	Q=3 S=4 F,F	Q=3 S=2 F,F
Matlab Semi	Q=2 S=2 F,T [012;346;457;467 ...;345;578;278]	Q=2 S=2 F,T [245;367]	Q=3 S=2 [012;278;378... ...458;356]	Q=2 S=2 F,F [012;278;345; ...346;456;678]	Q=2 S=3 F,T [345;126;456]	Q=2 S=2 F,T [467;367;125 478;347;678]

Table 5.3: Parameters used for the second test. The third parameter correspond to the "eq normalize" and "weighted mean" booleans, and the fourth parameter is the selected essential bands on each track, separeted with ";".

5.2 Results

The first test results are song-dependent, as shown in table 5.4. The selected default parameters of the system performs good results most of the cases. However, song 2 and song 3 are reporting an increase of the masking amount. Notice that the MRR can increase because when the masking produced by a track is reduced by filtering the track, the masking caused to it can increase.

S1	Raw	Edited
f,f	0.9296	0.9576
f,t	1.0203	0.9367
t,f	1.0220	1.0092
t,t	0.9865	0.9716

S2	Raw	Edited
f,f	1.2696	1.1650
f,t	1.1236	1.1990
t,f	1.2696	1.1650
t,t	1.1236	1.1990

S3	Raw	Edited
f,f	1.0534	1.0629
f,t	1.1102	1.0483
t,f	1.0399	1.0935
t,t	1.0953	1.0285

S4	Raw	Edited
f,f	0.9298	0.9784
f,t	0.8710	0.9067
t,f	1.0094	1.0189
t,t	0.9362	0.9856

S4	Raw	Edited
f,f	0.8782	0.9036
f,t	0.8114	0.8818
t,f	0.9029	1.0149
t,t	0.9413	0.9653

S6	Raw	Edited
f,f	0.914	0.956
f,t	0.859	0.996
t,f	0.971	0.950
t,t	0.911	0.954

Table 5.4: Masking reduction ratio (MMR) of the first test using the fully-autonomous implementation with default parameters. For each song, the algorithm is tested using the raw and edited audios, and toggling the "eq normalize" and "weighted mean" options.

The second test is divided into an objective and a subjective evaluation. The results are shown in table 5.5 and 5.6. The objective evaluation also reported worst performance on songs 2 and 3. As the subjective evaluation only tested 4 songs, it has been selected songs 1, 4, 5, 6 as they reported masking reduction.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**
Matlab Online	0.97671	1.1356	1.0560	0.8254	0.9730	0.9761
Matlab Automatic	0.9576	0.970	1.0556	0.9784	0.9242	0.8695
Matlab Semi	0.9130	1.0988	1.0714	0.9194	0.7737	0.9051

Table 5.5: Objective evaluation of the second test. It shows the Masking Reduction Ratio for each song and implementation.

Twelve people answered the Google Form poll. The subjects were asked to rate the understandability of the tracks using a number between 1 and 10. The results are shown in figure 5.1 and the average score for each song and implementation is shown in table 5.6. Song one results have small variance, with a clear preference for the original audio. The other songs presented more score variability, however, the semi-autonomous has the lower score.

Ideally, the subjective test would have been performed on an acoustically treated room using studio loudspeakers, and using a software similar to MUSHRA,

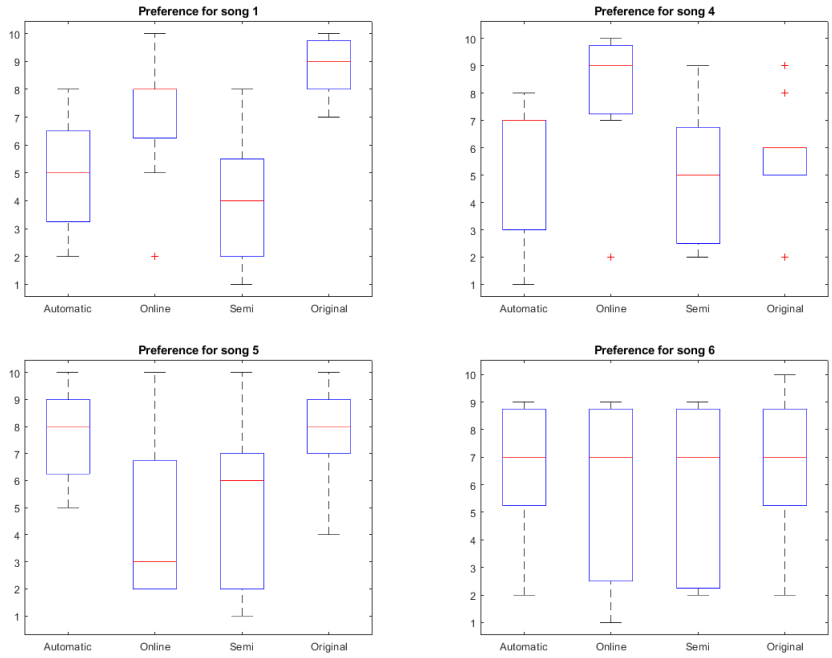


Figure 5.1: Subjective test results for songs 1, 4, 5 and 6

	Automatic	**Online**	**Semi**	**Original**
s1	4.8182	7.0909	3.8182	8.7273
s4	5.5455	8.1818	5.0909	5.4545
s5	7.8182	4.5455	5.0909	7.8182
s6	6.3636	5.6364	5.7273	6.6364

Table 5.6: Mean score by the users, for each song and implementation.

which randomizes the order of the songs for each subject and has an intuitive interface. That affected to the test length because, due to the test format, the evaluation has been harder for the subjects.

Chapter 6

CONCLUSION

6.1 Discussion

The first remarkable fact is that the system reports better performance using non-processed data. Only three types of processing have been used for the test; reverberation, compression, and equalization. This processes were intended to make an instrument sound in a specific way, for example, boosting the drum kick around 4kHz to make more audible the hit sound, i.e., to make it sound more metallic. Nonetheless, that processes do not improve the results, probably because of the compressor, as the non-linearity creates artefacts that can interact with the algorithm. Also, the "weighted mean" option seems to help most of the songs, as it helps to ignore the present noise on the signals when the main sound source is quiet. The "eq. normalization" does not always help and it is only useful when the algorithm applies hard filtering. However, a similar effect can be achieved using the parameter S. It affects to all bands instead of only the bands with excessive filtering, so decreasing S improves the overall tone of the song, but decreases the masking reduction efficiency.

In general terms, the tracks with better and worst performance differ in one characteristic; it seems that if a raw mixture has a proper arrangement and it has been recorded using proper microphones, i.e., the raw mixture has low masking, the performance of the algorithm is poor as the masking reduction is not really necessary and it probably filters more than what is needed. Then again, a raw mixture where all the instruments combined sound "diffuse" and hard to hear clearly, i.e., with a clear audible masking, the performance of the algorithm is bet-

ter. The algorithm is more suitable for amateur recordings than for professional ones, where this automatic masking reduction algorithm is probably not necessary as there are expert audio engineers.

Each type of implementation has its advantages. The fully-automatic and the semi-automatic are very similar as they share almost the same analysis and filtering. The manual essential band selection makes the semi-autonomous more capable to reduce masking if the user makes a proper selection, but that also means that can decrease the performance if not used properly.

The online version has a less stable parameter selection, it highly depends on the song context and it requires manual parameter selection. Nevertheless, this type of implementation is aimed to be used on a DAW by a user, meaning that the user is already selecting parameters on other plugins. It has the worst MRR of all versions and, as it uses EMA smoothing, when the song has abrupt changes the algorithm takes a second to be fully adapted to the new context. This could be fixed by automatizing the EMA α value. Also, this implementation seems to be more aggressive, achieving more difference compared with the raw audio, meaning that can have either an excellent or an awful performance.

The subjective test shows the semi-autonomous as the less preferred method for the subjects. The automatic version seems to be similar to the original one, most of the subjects do not hear any difference. If the parameter S is set too high, instead of decreasing the masking, it starts to increase it, so it is hard to have objective and subjective masking reduction at the same time. All three implementations have a similar MRR but the subjects tended to prefer the online version along with the raw audio.

The algorithm clearly makes a difference on the understandability of the mixtures, but maybe the used model is not precise enough to make it useful for music mixtures. That could be the reason why the subjective tests show that there is no huge improvement respect the raw audio. Auditory masking reduction is a complex problem and artistic choices could be also taken into account.

6.2 Conclusion

Masking is one of the most relevant processes in a musical mixture. The professional audio engineers focuses on reducing it and it makes the difference between an amateur and a professional mixture. This project aimed to automatically re-

duce the masking of a mixture, but due to the complexity of the human auditory system, a model based on manual equalization is used for the analysis stage to ensure the real-time approach.

Even though the algorithm reduces masking on most of the songs, it is not very consistent. The semi-autonomous implementation needs to be properly used to perform good masking reduction. At the same time, the online implementation can be useful as it requires less knowledge than the semi-autonomous and it is easier to adjust by the user. Also the VST format makes it easier to use on a mixture. In general, the algorithm needs to be adjusted, so the fully-autonomous approach is not completely achieved. However, all implementations with the parameters shown in table 5.2 reduces masking for most of the songs. The algorithm is better suited for amateur recordings, as the original masking is higher it also increases the utility of an automatic masking reduction plugin.

I have successfully implemented the offline and the online VST version. The VST has been harder to implement because it was the first time I built a VST plugin and because it is written using C++. That language does not have the same utilities as Matlab by default and I had to create a lot of functions. It is also harder to debug due to the inability to plot signals whenever I needed them. That is the reason why I have additionally implemented the online version on Matlab, so I could check the overlap-add algorithm and the frame-to-frame analysis and filtering. The results have not been as successful as I expected, the algorithm is not always consistent and is very parameter-sensitive. Nevertheless, the proper use of it can increase the understandability of the mixture, for these reasons I would recommend it for amateur users that has some idea on the mixing topic.

For amateur users, finding the most problematic frequencies and applying proper filtering can be hard and frustrating. For this reason, this algorithm and future works on the topic are highly relevant, this work could help reducing masking with low time cost and knowledge, even though there is a lot of hard work to do on the topic.

6.3 Future work

After analyzing the algorithm performance on each of its stages, it seems like one of the problems is related to the magnitude and ranking system. It assumes that the bands with higher magnitude are most likely the most essential bands, but sometimes these bands are the ones creating the highest masking, i.e., the bands

with higher magnitude are also the bands more likely to cause masking. Another way to rank the bands is taking into account the overall magnitude of the track and controlling the surplus of magnitude in individual bands.

Another way to perform better ranking is the method that J. Reiss [27] created after the one used on this project. It uses pre-defined profiles for different types of instrument, allowing to perform a proper filtering to each instruments, as each one has very different necessities.

Likewise, the equation 4.2 used to quantify the masking is not completely accurate and maybe the new hardware advances allow to implement a more complex model, similar to the human auditory system. It includes implementing the ear impedance, the transfer functions and all the internal processing including its non-linearities.

Bibliography

[1] S. Hafezi and J. D. Reiss, "Autonomous multitrack equalization based on masking reduction," *AES: Journal of the Audio Engineering Society*, vol. 63, no. 5, pp. 312–323, 2015.

[2] L. Chittka and A. Brockmann, "Perception Space—The Final Frontier," *PLOS Biology*, vol. 3, no. 4, 2005.

[3] Wikimedia Commons, "File:Cochlea.png — Wikimedia Commons, the free media repository," 2016.

[4] Wikimedia Commons, "File:Organ of corti.svg — Wikimedia Commons, the free media repository," 2019.

[5] H. Fletcher, "Auditory Patterns," *Rev. Mod. Phys.*, vol. 12, pp. 47–65, jan 1940.

[6] T. Reichenbach and A. J. Hudspeth, "A ratchet mechanism for amplification in low-frequency mammalian hearing," *Proceedings of the National Academy of Sciences*, vol. 107, no. 11, pp. 4973–4978, 2010.

[7] M. A. Ruggero and N. C. Rich, "Furosemide alters organ of corti mechanics: evidence for feedback of outer hair cells upon the basilar membrane.," *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 11, pp. 1057–1067, apr 1991.

[8] M. B. Sachs and N. Y. Kiang, "Two-tone inhibition in auditory-nerve fibers.," *The Journal of the Acoustical Society of America*, vol. 43, pp. 1120–1128, may 1968.

[9] B. R. Glasberg and B. C. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Research*, vol. 47, no. 1, pp. 103–138, 1990.

[10] B. C. Moore, "Frequency Analysis and Masking," 1995.

[11] B. Owsinski, "Mixing Engineer's Handbook," 1999.

[12] M. Brecht De, J. D. Reiss, and R. Stables, "Ten Years of Automatic Mixing," *Workshop on Intelligent Music Production*, no. September, pp. 1–5, 2017.

[13] D. Dugan, "Automatic Microphone Mixing," *JAES*, vol. 23, no. 6 pp, pp. 442–449, 1975.

[14] S. Julstrom and T. Tichy, "Direction-Sensitive Gating: A New Approach to Automatic Mixing," *JAES*, vol. 32, no. 7/8 pp, pp. 490–506, 1984.

[15] E. P. Gonzalez and J. D. Reiss, "Automatic mixing: Live downmixing stereo panner," *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07)*, pp. 1–6, 2007.

[16] J. A. Maddams, S. Finn, and J. D. Reiss, "An autonomous method for multitrack dynamic range compression," *15th International Conference on Digital Audio Effects, DAFx 2012 Proceedings*, pp. 1–8, 2012.

[17] M. Hilsamer and S. Herzog, "A statistical approach to automated offline dynamic processing in the audio mastering process," *DAFx 2014 - Proceedings of the 17th International Conference on Digital Audio Effects*, pp. 1–6, 2014.

[18] Z. Ma, B. De Man, P. D. Pestana, D. A. Black, and J. D. Reiss, "Intelligent multitrack dynamic range compression," *AES: Journal of the Audio Engineering Society*, vol. 63, no. 6, pp. 412–426, 2015.

[19] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "Deep neural networks for dynamic range compression in mastering applications," *140th Audio Engineering Society International Convention 2016, AES 2016*, 2016.

[20] E. T. Chourdakis and J. D. Reiss, "A machine-learning approach to application of intelligent artificial reverberation," *AES: Journal of the Audio Engineering Society*, vol. 65, no. 1-2, pp. 56–65, 2017.

[21] E. T. Chourdakis and J. D. Reiss, "Automatic Control of a Digital Reverberation Effect using Hybrid Models," *AES 60th International Conference: Dereverberation and Reverberation of Audio, Music, and Speech*, pp. 1–8, 2016.

[22] D. Ward, J. D. Reiss, and C. Athwal, "Multitrack Mixing Using a Model of Loudness and Partial Loudness," *133th Audio Engineering Society*, vol. 133, p. 8693, 2012.

[23] A. Wilson and B. Fazenda, "An evolutionary computation approach to intellgient music production informed bu experimentally gathered domain knoledge.," *Proceedings of the 2nd AES Workshop on Intelligent Music Production*, no. September, pp. 4–5, 2016.

[24] E. Perez-Gonzalez and J. D. Reiss, "Automatic equalization of multi-channel audio using cross-adaptive methods," *127th Audio Engineering Society Convention 2009*, vol. 1, pp. 453–458, 2009.

[25] S. I. Mimilakis, K. Drossos, A. Floros, and D. Katerelos, "Automated Tonal Balance Enhancement for Audio Mastering Applications," no. May, 2013.

[26] Z. Ma, J. D. Reiss, and D. A. Black, "Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering," *134th Audio Engineering Society Convention 2013*, pp. 173–182, 2013.

[27] D. Ronan, Z. Ma, P. M. Namara, H. Gunes, and J. D. Reiss, "Automatic Minimisation of Masking in Multitrack Audio using Subgroups," pp. 1–13, 2018.

[28] T. Necciari, "Auditory time-frequency masking : Psychoacoustical sound signals," 2011.

[29] H. Fletcher and W. A. Munson, "Loudness, Its Definition, Measurement and Calculation," 1933.

[30] R. D. Patterson, I. Nimmo-Smith, D. L. Weber, and R. Milroy, "The deterioration of hearing with age: Frequency selectivity, the critical ratio, the audiogram, and speech threshold," *The Journal of the Acoustical Society of America*, vol. 72, no. 6, pp. 1788–1803, 1982.

[31] R. S. Tyler, J. W. Hall, B. R. Glasberg, B. C. Moore, and R. D. Patterson, "Auditory filter asymmetry in the hearing impaired," *The Journal of the Acoustical Society of America*, vol. 76, no. 5, pp. 1363–1368, 1984.

[32] M. Hiipakka, "Measurement Apparatus and Modelling Techniques of Ear Canal Acoustics," *Science*, p. 93, 2008.

[33] T. Zibakowski, "Combination tones in the model of central auditory processing for pitch perception," *Archives of Acoustics*, vol. 37, no. 4, pp. 571–582, 2012.

[34] B. C. Moore, R. W. Peters, and B. R. Glasberg, "Auditory filter shapes at low center frequencies.," *The Journal of the Acoustical Society of America*, vol. 88, pp. 132–140, jul 1990.

[35] M. J. Shailer, B. C. Moore, B. R. Glasberg, N. Watson, and S. Harris, "Auditory filter shapes at 8 and 10 kHz.," *The Journal of the Acoustical Society of America*, vol. 88, pp. 141–148, jul 1990.

[36] J. R. Dubno and D. D. Dirks, "Auditory filter characteristics and consonant recognition for hearing-impaired listeners.," *The Journal of the Acoustical Society of America*, vol. 85, pp. 1666–1675, apr 1989.

[37] B. C. Moore and B. R. Glasberg, "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns.," *The Journal of the Acoustical Society of America*, vol. 74, pp. 750–753, sep 1983.

[38] B. Owsinski, *The Mastering Engineer's Handbook: The Audio Mastering Handbook*. 2008.

[39] M. Senior, *Mixing secrets for the small studio*. Taylor & Francis, 2011.

[40] R. Izhaki, *Mixing audio: concepts, practices and tools*. Taylor & Francis, 2013.

[41] A. R. Fernandez, "Automatic Masking Reduction," 2020. `https://github.com/arufer720/multitrack_eq`.