Advanced automatic mixing tools for music

Perez Gonzalez, Enrique

Advanced Automatic Mixing Tools for Music

Submitted by Enrique Perez Gonzalez
For the Ph.D. degree of
Queen Mary
University Of London
Mile End Road
London E1 4NS

September 30, 2010

I certify that this thesis, and the research to which it refers, are the product of our own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. I acknowledge the helpful guidance and support of our supervisor, Dr. Johua Daniel Reiss.

Abstract

This thesis presents research on several independent systems that when combined together can generate an automatic sound mix out of an unknown set of multi-channel inputs. The research explores the possibility of reproducing the mixing decisions of a skilled audio engineer with minimal or no human interaction. The research is restricted to non-time varying mixes for large room acoustics. This research has applications in dynamic sound music concerts, remote mixing, recording and postproduction as well as live mixing for interactive scenes.

Currently, automated mixers are capable of saving a set of static mix scenes that can be loaded for later use, but they lack the ability to adapt to a different room or to a different set of inputs. In other words, they lack the ability to automatically make mixing decisions. The automatic mixer research depicted here distinguishes between the engineering mixing and the subjective mixing contributions. This research aims to automate the technical tasks related to audio mixing while freeing the audio engineer to perform the fine-tuning involved in generating an aesthetically-pleasing sound mix. Although the system mainly deals with the technical constraints involved in generating an audio mix, the developed system takes advantage of common practices performed by sound engineers whenever possible. The system also makes use of inter-dependent channel information for controlling signal processing tasks while aiming to maintain system stability at all times. A working implementation of the system is described and subjective evaluation between a human mix and the automatic mix is used to measure the success of the automatic mixing tools.

Acknowledgments

Thanks to Dr. Joshua D. Reiss for supervising me during this long journey. Thanks to Professor Mark Sandler for giving me the opportunity to be part of The Centre For Digital Music and for believing in my ideas. Thanks to Xui for great conversations during the early starts of this journey, which ended in great ideas. Thanks for Nagel and Andrew Robertson for allowing me to use some multi-track recordings that ended being invaluable research data. Thanks to Antonio Zacarias, Mauricio Ramirez, Francisco Miranda, Jorge Urbano, Renato de la Rosa, Oscar Aguilar and Jaime Gonzalez for their kind support toward my research.

Thanks to Alice Clifford for her support especially when things seemed impossible to achieve. Thanks to Agnes Doeringer, who played a paramount part in my life during the past 5 years of my life. Upmost thanks to my parents, Enrique Perez Adame and Lucia Gonzalez Iñiguez, without whom none of this would have been possible. Huge thanks for Matthew Davis, Michel Terrell, Martin James Morrell, Steve Welburn, Dan Stowell, Rebecca Stewart, Andrew Nesbit, George Fazekas, and Angi Atmadjaja witch suffered correcting my spelling more than once. Thanks to my examiners Udo Zoelzer and Tony Stockman I truly appreciate you took the time to do so. Big thanks to: Laura Margottini, Youtha Cuypers, Ilya Cuypers, Sabine Altendorf, Leonado Jaso, Adam Stark, Maria Jafari, Larisa and Kurt Jacobson, Louis Martignon, Sylvie Stuiz, Asterios Zacharakis, Vincent Verfaille, Christian Uhle, Tomas Wilmering, Chistopher Harte, Chris Landone, Chris Sutton, Chris Cannam, Matthias Mauch, Mark Pumbley, Simon Dixon, Katy Noland, Heather Andrews, Ben fields, Yves Raimond and Anne-So Noiret, Juan Pablo Angulo, Rodolfo Rodriguez, Alberto Garcia, Adrian Bisiacchi, Jmmy Robertson, Robert Macrae, all C4DM and to all the people which some how participated in testing or contributed in some form. Finally massive thanks to those which in the process of printing and submitting I forgot to add them to this thanks, but if you should be here you know who you are.

Contents

Table of contents

List of figures and tables

List of figures

List of tables

List of symbols and abbreviations

List of math symbols

ΔG	Gain before feedback
NOM	Number of open microphones
NOMGR	Number of open microphones gain reduction
AAD	Auto attenuation depth
V	The volume of a large room
F_L	Large room limiting frequency
RT_{60}	Time in seconds it takes the amplitude of the source to decay 60dBs
$x(n)$	Input
$y(n)$	Output
n	Sample
$x_e(n)$	External source input
m	Channel number, define from $0,\ldots, M-1$
M	Maximum number of sources
μ	Reference channel number, define from $0,\ldots, M-1$
$x_m(n)$	M channel inputs
$y_m(n)$	M channel outputs
$x_\mu(n)$	Reference channel input
c	Speed of sound
T_c	Temperature in Celsius
R_1	Distance 1 from audio source
R_2	Distance 2 from audio source
dB_{spl}	Sound pressure level measured in dBs
τ	Delay
f	Frequency
$\Delta\phi$	Phase difference
dBFs	Decibels full scale
H(ι)	Analogue transfer function in the Laplace domain
H(k)	Sampled transfer function
FFT	Fast Fourier transform
IFFT	Inverse fast Fourier transform
$cv_m(t)$	Control vector in the time domain
$cv_m(n)$	Control vector
$fv_m(n)$	Feature vector
$\boldsymbol{cv}_m(n)$	Control vector multiple dimentions
$\boldsymbol{fv}_m(n)$	Feature vector multiple dimentions
$w(n)$	Time domain Window function
$w_{HN}(n)$	Hann Window function
$xg_m(n)$	Gated M channel inputs
$xg_\mu(n)$	Gated reference channel input
$X_m(k)$	Windowed FFT transform of $xg_m(n)$

$X_\mu(k)$	Windowed FFT transform of $xg_\mu(n)$
$Ha_m(k)$	Simple transfer function
$X_{mm}(k)$	Auto spectrum
$X_{m\mu}(k)$	Cross spectrum
$H_m(k)$	Auto spectrum / cross spectrum transfer function
$Hv_m(k)$	Vectored averaged transfer function
$HR_m(k)$	Real transfer function
$HI_m(k)$	Imaginary transfer function
S	Number of iterations
r	Amount of decrement applied every time $x_m(t)$ is greater than 1
d	Mutual delay time between signals
F_c	Cancelation frequency notch
$\delta_{PHATm}(n)$	Impulse response of a phase transform weighed transfer function
t	Time
t_0	Time cero
t_∞	Infinite time
B	Fixed accumulator
$\delta_{Bm}(n)$	Accumulated impulse response of m accumulated B times
B_m	Adaptive accumulator
α	Constant of minimum number of validating operations
$\delta_m(n)$	Impulse response of m
$\tau_{\mu m}(n)$	Time between sources $x_\mu(n)$ and $x_m(n)$
$\rho_{\mu m}(n)$	Polarity time between sources $x_\mu(n)$ and $x_m(n)$
$\boldsymbol{fv}_{\tau\mu}$	Feature vector containing face an polarity for channel μ
$\boldsymbol{fv}_{\tau m}$	Feature vector containing face an polarity for a given channel m
$\boldsymbol{cv}_{\tau m}(n)$	Delay control data value
$\boldsymbol{cv}_{pm}(n)$	Polarity control data value per signal
$\Delta S_{\mu,i}(k)$	Spectral masking of channel μ
$Sa_{\mu,i}(k)$	Accumulated spectral masking of channel μ
$xg_m(n)$	Adaptive gated $x_m(n)$ with respect to $x_e(n)$
$x_e(n)$	External input
$H_L(n)$	All-pass filter network left
N	Frame
$Qx_{\mu,i}(k)$	Quantised calculation of the masking index before the effect
$Qy_{\mu,i}(k)$	Quantised calculation of the masking index after the effect
$Qp_{\mu,i}(k)$	Unmasked-rate percentage
R_m	Total number of sources in the same feature category
U_m	User priority
W	Panning width
P_m	Relationship vector between the user priority and $R_m(n)$
tr_{ps}	Psychoacoustic panning threshold
$y_L(n)$	Left output of a stereo signal processing device
$y_R(n)$	Right output of a stereo signal processing device
$y_{mix}(n)$	Monaural mix.
d	Decrement step
$r(n)$	Histogram gain scaling factor for all channels
$B_{max}(n)$	Highest bin in the histogram
$h_{k,m}$	Filter bank individual filters

$hq_{k,m}$	Equalizer individual filters
$h_{k,m}(n)$	Spectraly decomposed input
$he_{k,}(n)$	Spectraly decomposed external input
$tr_{k,m}(n)$	Band limited adaptive threshold
$rs_{k,m}(n)$	Gain scaling factor set per channel
$r_{k,m}(n)$	Histogram gain scaling factor
$Bmax_{k,m}(n)$	Value taken by the highest bin

List of abbreviations

RMS	Root mean square
NOMA	Number of open microphones attenuation
ACG	Automatic gain control
GPI	General port interface
TTL	Transistor-transistor logic
RS-232	Recommended standard 232 for serial communication
VCA	Voltage controlled amplifiers
DC	Direct current
MIMO	Multi-input multi-output
LPF	Low-pass filter
HPF	High-pass filter
APF	All-pass filter
FFT	Fast Fourier transform
IFFT	Inverse fast Fourier transform
VCA	Voltage controlled amplifier
DAC	Digital analogue converter
ADC	Analogue digital converter
RT30	Measurement of reverberation that measures the time it takes to decay 30dB

Part I Introduction and background

Chapter 1
Introduction

This research pursues the knowledge required to develop automatic mixes comparable in quality to those performed by professional human mixing console operators.

1.1 Justification

The justification of this research is the need of non-expert audio operators and musicians to be able to achieve a quality mix with minimal effort. Currently, mixing is a task that requires great skill, and experience, and can be sometime tedious. For the professional mixing engineer this kind of tool will reduce sound check time and will prove useful in festivals situations where there are multiple music groups and changing from one group to another should be done really quickly. Currently, large audio productions often have hundreds of channels, being able to group some of those channels into an automatic mode will ease the effort required by the audio engineer. There is also the possibility of applying this technology to remote mixing applications where latency is too large to be able to interact with all aspects of the mix. Finally, growth in demand for video games and their ever-increasing audio processing requirements makes automatic mixing for games a promising research area. Where hundreds of ever changing audio stems needs to be prioritised and mixed on real time to enhance the game experience.

1.2 Scope of the research

It is important to understand that in the context of this research remixing and mixing is not the same. Mixing refers to combining audio signals and spectrally or dynamically modifying them. On the other hand remixing refers to taking a series of audio sources, some of which have been previously mixed, and editing them in time to create a substantially different piece of music. For example the disc jockey changes the tempo of a mixed song to blend and extend its length. The final remix is a longer, more complex variation of the original.

For the purpose of this research it is also important to make a distinction between *automatic* mixing processes and *automated* mixing processes. An automatic process involves an autonomous process. This autonomous process can be treated as a constrained rule problem in which the design of the control rules determines the process to be applied to the input signals. The automated process, on the other hand, is the result of playing back in sequence a series of user recorded actions. Automated mixing boards are commonly referred as recallable mixers due to its ability to store and load its control parameters. This involves playing back previously recorded and stored actions, regardless of whether automatically or manually generated.

The work presented in this thesis is limited to large room acoustics for static adaptive mixing. The system makes thorough use of inter-dependent channel information for controlling signal processing tasks. Finally, the research utilises constrained rules based on engineering and common mixing practices to reduce the convergence time. This will be elaborated on in the following chapters.

1.3 Contributions of this thesis

Earlier systems by Dugan (Dugan 1975; Dugan 1989) and Julstrom (Julstrom and Tichy 1984) were only concerned with automatic gain handling and required a significant amount of human interaction during setup to ensure a stable operation. The automatic tools described in this thesis, however, take advantage of current mixing board recall functionality and add autonomous decisions to the mixing process. To the best of our knowledge, no current automatic mixing device is capable of equalizing and organising the gain mix structure while taking care of acoustic or technical constraints. The system described herein uses novel cross-adaptive side chain processing together with novel system stability signal processing. It is comprised of seven novel sections:

1- A self-normalizing algorithm, which keeps the system gain under a usable stable condition while optimising headroom.
2- A hardware gain corrector to ensure correct analogue to digital conversion.
3- An automatic signal time offset and polarity corrector.
4- A spectral cross-adaptive channel enhancer that permits the user to realize complex channel enhancing tasks with ease.
5- A self-equalizing system that minimises spectral masking.
6- A real time autonomous panner device based on common panning practises
7- A set of algorithms to optimise fader gain levels

This comprises a mixing system that is capable of adjusting input gain, fader levels and EQ; correcting delay and polarity problems; enhancing channel signals, and panning all channels while ensuring that all of the system remains stable.

1.4 Overview

This thesis is divided into 4 main parts and 11 chapters. The first part consists of the introduction and background part; then a second part discusses the automatic mixing tools in detail; the third part consists of the conclusion and future work and finally an appendices section is presented at the end. The bibliography is presented at the very end of this thesis.

This first chapter itself serves as an introduction and aims to clearly point out the objectives and contributions of the research. It also mentions the research scope and justification behind this research. Finally it will state a concise thesis statement that is the driving force behind the research presented herein. Chapter 2 is mainly a literature survey of automatic mixing. It touches the background aspects of it up to current state of the art.

Part 2 first introduces the core building block technologies proposed in this thesis for automatic mixing processing. Chapter 3 deals with the main concepts needed for the full understanding of this thesis. It contains the core generalized framework elements developed during this research and is considered by the author to be one of the most important contributions in this thesis. From chapter 4 to chapter 10 a series of automatic mixing tools are presented. Chapter 4 deals with normalization methods for maintaining system stability. It proposes a method for ensuring there is no undesired artefact introduced to the mixing system when performing an automatic mixing process. Chapter 5 researches a procedure for automatically setting head-amplifier gain. Chapter 6 deals with automatic polarity and delay correction of multiple microphones capture of a single musical source. Chapter 7 describes a cross-adaptive enhancer based on the introduction of a cross-adaptive mapping function. The process used by the proposed enhancer resulted in development of the framework presented previously in chapter 3 and the general framework

researched in this thesis. Chapter 8 investigates an automatic panning technique together with some in depth subjective and objective results. This chapter is core in understanding the potential and limitations of an automatic mixing tool, it also proposes a way in which the effectiveness of automatic mixing tools could be subjectively evaluated. Chapter 9 explores an automatic fader method, which differs from previous state of the art approaches due to its accumulative statistical method and the use of psychoacoustic loudness. Chapter 10 researches the basis for an autonomous self-equalization mixing system for music. Automatic mixing tools are presented in the order of a standard audio mixer signal flow, from head-amplifier to stereo mixing bus, except for the automatic equalization section, which due to its methodological similarity to the automatic faders, makes more sense to introduce it as the last tool presented in this thesis.

Part three consists of chapter 11. This last chapter states the conclusions of this thesis and states a series of future routes where this research can be expanded.

Part four consists of the appendix section. Appendix A contains a list of the authors publications related to the research presented in this thesis. A URL to multimedia files with demonstrations of each of the auto mixing tools presented herein is also included. Finally, a bibliography of sources cited in this work is presented.

1.5 Aim and objectives

The aim of this thesis is to create a set of tools, for live music mixing, that when used together will generate an automatic sound mix with comparable quality to a human generated one.

Objectives:

- Take care of the technical and physical constraints of the mix. For example, limit gain to avoid distortion or avoid spectral masking of sources to improve intelligibility.

- Simplify complex mixing tasks. For example, take care of signal polarities and time offset correction between signals

- Maintain the system under a stable condition at all time, for example, avoid acoustic feedback.

- Make use of common mixing practices to achieve autonomous tasks, such as automatically panning sources.

1.6 Thesis

The mixing of musical signals has always been considered to be a manual task. Many years of experience in the field of live musical production prior to the research presented herein have me to believe that there is scope within musical mixing for the development of automatic mixing tools. The purpose of these tools would be to reduce the work burden on the engineer, and the limitations on what tasks can be automated are explored in this thesis. From this we aim to determine whether the potential to automate aspects of musical mixing is correct, and to ultimately prove whether an automatic musical mix can be produced which is of comparable quality to a human mixer.

Chapter 2
Background and state-of-the-art

Sound reinforcement, recording and broadcasting has been key to the dissemination of music to the masses ever since Alexander Graham Bell invented telephony in 1876 and Thomas A. Edison successfully recorded “Mary had a little lamb” in 1877. The need for distribution of audio media and live events to remote locations took telephony into the broadcasting scene. During the 1940s the need for having multiple input transducers combined into a single amplifier system created the path for developing what is now called the mixing console.

2.1 The Mixer

A mixer is a device used to mix two or more signals into a composite signal (Ballow et al. 2002). Audio mixing is often performed for the purpose of down mixing or up mixing. Down mixing is used to reduce the number of input channels into a composite output mix with fewer output channels than input channels, and up mixing is performed when the resulting composite mix has more output channels than input channels. An example of up mixing is the case of panning a monaural signal to achieve a false sense of stereophony. In most cases, each input consists of a single channel. In more complicated scenarios each track to be mixed can have several channels, as is common practice for cinema.

In practice, both mixing procedures can be used together. For example, if we aim to down-mix eight channel monaural inputs into a stereo down-mix, first we may up-mix each of the monaural sources into a two channel source by using a panner and then we may down mix eight sets of two channel signals into a single two channel down-mix.

Some of the processes involved during generating an audio mix are artistic choices and therefore subjective. On the other hand, a fair amount of the mixing process is dictated by physical and electronic limitations of the devices involved in the mix. These devices includes microphones, speakers, amplifiers, a mixing console and in general any sound processor involved in the audio chain. Therefore generating a mix is also a technical engineering process. A clear example of an engineering approach is the general objective of delivering an audio mix free of noise and distortion (Snyder 1953).

In this section we will discuss the mixing path of a generic audio console, with special emphasis on identifying the engineering contributions and the creative contributions to each stage of the mixing path. The objective of this is to identify the difference between the subjective and the technical approach to mixing.

2.1.1 The input channel

An audio mix has several intermediate signal processing stages before achieving the final composite output signal as shown in Figure 1, some of which have dynamic and spectral impact over the input and output signals.

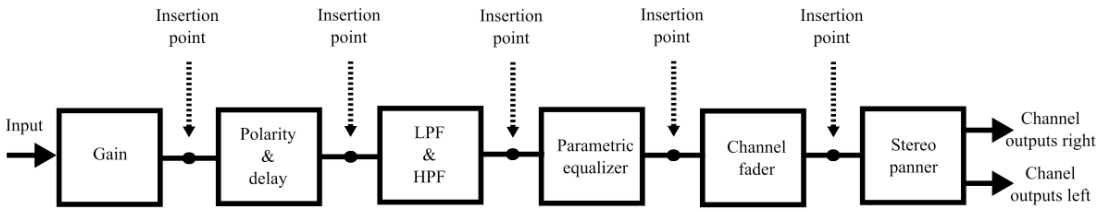

Figure 1 Processing stages of a mixer channel.

The first processing stage of any mixer containing an input stage is “Gain”. This gain control scales the signal with the basic requirement that the maximum input signal does not go beyond the overall amplitude limits of the electronics contained within the audio mixer. This is to avoid distortion due to signal clipping. The maximum limit to which the gain can be set is of an engineering nature, because of technical limitations in the electronic components, while setting it lower can have a subjective explanation, such as setting all the channel faders at a more comfortable level (Rumsey and McCormick 2006).

Intermediate link points between processing stages of a mixer allow for optional insertion sections where equalizers, effects or dynamic processors such as gates and compressors can be inserted into the signal path. The insertion point permits the output of the last section to be connected to the input of the inserted processor and the output of the inserted processor to be inserted to the input of the next stage.

Most mixers will offer a reverse polarity switch, usually located after the head-amplifier gain. The overall function of this switch is to change the overall phase of a signal by 180°. More modern designs also have a delay adjuster that permits the addition of delay to the signal. This allows for the introduction of a precise linear phase change to the input signal.

One of the most common processors inserted into the mixer is an equalizer. The equalizer is normally comprised of an adjustable Low Pass Filter (LPF) and a High Pass Filter (HPF) and a parametric equalizer just after the gain stage. A parametric equalizer consist of a set of filters in which the centre frequency, Q and gain of the filters is controllable by the user. The boundaries of the cut off frequency of the LPF and of the HPF have the purpose of reducing the noise floor of the system by constraining the bandwidth of the source and trimming out the spectral noise contained outside the useful bandwidth. For example in the case of a piccolo it might be necessary to set the HPF to avoid undesired noise on the low frequencies. A common reason for using a LPF and a

HPF is not only to remove noise but also to remove sections of the input spectrum of the signal in order to reduce spectral masking between other channels. The LPF and HPF can also have a subjective use where the sound mixer desires to cut the sub harmonics or upper harmonics of a signal for creative reasons.

The parametric equalizer is usually comprised of more than one peaking filter. Each peaking filter is capable of adjusting frequency, bandwidth and gain. In some cases they have a switch, which can turn them into a shelving filter instead of a peaking filter. From a practical point of view the equalizer can be used for altering the spectrum of the signal. In many cases it is used to boost frequencies, making the system more likely to yield unstable behaviour. An audio system is an acoustic feedback loop system and whenever the overall gain of the feedback loop goes beyond the nominal level, it generates instability. The phenomenon is commonly known in the audio industry as “acoustic feedback”, named howlback or the Larsen effect (Rombouts et al. 2006).

From a subjective point of view equalisation is commonly used for enhancing or diminish sound qualities of the source. One of the problems of boosting the equalizer is that it requires compensating the overall gain of the channel so that the gain remains nominal. This is a tedious reiterative problem, which can be automated. Inexperienced mixers tend to boost the equalizer parameters more than to cut them. This makes it difficult to achieve a stable system with good acoustic gain.

It is important to understand that not everything is equalizable. For instance, some room effects in the spectrum, like comb-filters, are not equalizable, since they are primarily a time domain problem. This might open the door to some automatic delay correcting. For example, when capturing a single source with two microphones that are separated from each other and then mixed together, there will be a comb-filter effect, which will suppress frequencies whose wavelengths are integer multiples of half the distance of the separation of the microphones. Adding delay to one of these microphones will

avoid this type of artefact. A line input delay in a channel is not a new concept, e.g. the Yamaha PM5D mixing console, but setting it up automatically to reduce comb-filtering is.

The channel faders set the mix levels of each channel and are one of the more subjective parts of the mixing process as they are influenced by taste and priority. For example if the main artist is a guitar player it is quite probable that the guitar levels will be relatively high in comparison to the other channels. In live mixing, the lead singers maximum acoustic gain tends to be the reference to which the rest of the channels are mixed.

Mixing consoles often have numerous routing stages during the mixing path. The routing stages in a mixing console tend to be used to generate sub mixes and bus assignments. Examples of routing usage are stage monitoring, or spatial representations of a mix (5.1, 7.1, etc.). Panning is the most common routing procedure and can also be considered a type of insertion effect as it attempts to introduce a false sense of space by rerouting a monaural signal into a stereo output. A number of engineering considerations can be used for determining panning, including spectral multi-channel masking and low frequency content.

2.1.2 The master section

Finally, at the end of every mixer there are the summing busses that are in charge of adding all channels routing the signal to the master faders. The master fader is the final gain stage before the signal reaches the output. In more complex systems with multi-channel outputs, the master fader tends to be substituted with a matrix routing system. Its level is dependent on the intended sound pressure level. It is possible to automate its maximum level to match the maximum possible acoustic gain by knowing acoustic parameters such as sensitivity of the transducers. It should be possible to rescale the master fader so that it can assure that the overall mix level remains without feedback. In our knowledge, this automatic rescaling has never been attempted and is a subject of current research.

2.2 State of the art in automatic mixing

Perhaps one of the most important changes in mixing consoles in recent years is the introduction of digital electronic mixing boards. Based on their electronics, they can be classified into analogue, digital or hybrid. In general, mixers can be divided into active, passive, adjustable or non-adjustable (Ballow et al. 2002). For the purpose of this research the most important classification is to distinguish them according to the way they perform the mixing process, from this perspective they can be classified into *automatic* and *automated*.

In the context of this research the automated process is the result of playing back in sequence a series of user recorded actions. This involves playing back previously recorded and stored actions, regardless of whether being automatically or manually generated. Automation was first introduced by MCI in the VCA automation for their JH500 series mixing consoles (Rumsey and McCormick 2006). Automated mixers are becoming more common with the current approach being to pre-record settings and presets. These presets are independent of the input and, if the inputs are changed, they have to be either heavily edited or completely reprogrammed.

An automatic process involves an autonomous process. This autonomous process can be treated as a constrained rule problem in which the design of the control rules determines the process to be applied to the input signals. At present automatic mixers have found their way mainly into the speech market.

Currently, automatic mixing for music is under-developed. This is partly because their designs are more suitable for dedicated installations like conference halls and lecture rooms. Another factor that has prevented them from entering the music market is the fact that most current designs tend to introduce undesired gating artefacts on non-speech signals. In the current state-of-the-art of automatic mixing the term “automatic microphone mixer” and “automatic mixer” is often used interchangeably. A study of the current state of automatic mixers is presented next.

2.2.1 Automatic mixing

In the literature automatic mixing usually refers to automatic microphone mixing. Perhaps one of the best analyses of the current state of automatic mixers has been compiled by Glen M. Ballou (Ballow et al. 2002) and is best presented by the following quote:

> *"To date the operational concepts used in digital automatic microphone mixers have not varied far from the previously described concepts underlying the analog automatic microphone mixers. This is likely to change, but as future digital automatic mixing concepts will be hidden deep within computer code the manufacturers may be unwilling to reveal the details of operational breakthroughs; they will likely be kept as close guarded company secrets. New concepts in automatic mixing might only become public if patents are granted or technical papers are presented."*

This statement by Ballou clearly identifies a lack of development in the automatic mixing development. It also acknowledges that due to the inability of exploring digital code inside industrial products many of these advancements are likely to pass unnoticed in the scientific community. Finally, he acknowledges the need for the publication of the methods used for developing this new auto mixing technologies.

In standard speech mixing, it is a common practice to open only those microphones that are in use. This maximises gain before feedback. Dugan (Dugan 1975) stated the basics of automatic mixing and showed that every doubling of the number of microphones reduces the available Gain Before Feedback by 3dBs. In order to maintain the audio system under feasible gain before acoustic feedback occurs a circuit known by the name of NOMA, or number of open microphones attenuation has been devised by (Dugan 1975). The automatic attenuation produced by such a circuit is known as the number

of microphone gain reduction, *NOMGR*, and is given in dBs. It is dependent on the number of open microphones, *NOM*. Such a NOMA circuit is characterised by the following equation:

$$NOMGR = 10\log NOM\ . \qquad (1)$$

Most current designs will restrict the maximum number of microphones to be open regardless of the overall number of microphones in the system. Most automatic mixers do not go from a complete "on" to "off" state. In practice they tend to go from one state to another by producing only a 15dB gain change (Ballow et al. 2002). Some automatic mixers will offer an "off" state attenuation, also known as auto attenuation depth, *AAD* and is given in dBs. The input channel attenuation setting accomplishes the purpose of optimising the system gain before feedback, especially when the number of microphones is large. The relation between the gain before feedback, ΔG, and the auto attenuation depth, *AAD*, is given by the following equation:

$$\Delta G = 10\log\frac{NOM}{1+(NOM-1)10^{AAD/10}}\ . \quad (2)$$

Another method for controlling the gain is the use of a device called automatic gain control or ACG. The automatic gain control mixers operate by setting up the quietest active microphone as the reference gain. The microphone has maximum gain before feedback while louder talkers will activate the AGC to reduce the overall gain level. AGC tends to have similar control parameters to a compressor, for which, in order to minimise artefacts, an attack and release time must be set up. This increases the user complexity of an automatic mixer. In many cases the settings are fixed by the manufacturer to simplify operation. Unfortunately (Ballow et al. 2002) this limits their application.

According to (Ballow et al. 2002), the design objectives of an automatic microphone mixer are:

1. *"Keeps the sound system gain below the threshold of feedback."*
2. *"Requires no operator or sound technician at the controls."*
3. *"Does not introduce spurious, undesirable noise or distortion of the program signal."*
4. *"Can be installed as easily as a conventional mixer."*
5. *"Responds only to the speech signals and is relatively unaffected by extraneous background noise signals."*
6. *"Activates input channels fast enough that no audible loss of speech occurs."*
7. *"Allows more than one talker on the system when required by the discussion content, while still maintaining control over the overall sound gain."*
8. *"Adjusts the system status outputs for peripheral equipment control and can interface with external control systems for advanced system design if desired."*

Objectives 5,6 and 7 are speech-specific, and would require modification for designing a music automatic mixer. For automatic mixing of music, opening and closing inputs may result in unnatural artefacts. For this reason, we aim to research a system that ideally will require minimal or no opening and closing of microphones.

Currently automatic mixers tend to be part of huge conference systems and tend to interact with other devices, in many cases by using a general port interface, GPI. This typically uses standard communication protocols and technologies like TTL, RS-232, 1 to 10V, etc. Some of these conference systems tend to have multiple rooms and outputs. This adds complexity to the mixer output stages, needing in many cases output matrix sections. Interconnectivity is an important feature of automatic mixers.

2.2.2 Automatic mixing classification

According to (Ballow et al. 2002) automatic microphone mixers can be classified according to the way they operate as fixed threshold, gain sharing directional sensing, multivariable dependent, noise-adaptive threshold and variable threshold. From the different methods just mensioned, only the variable threshold design by Dugan, developed in 1975, is intended for music, the rest is for speech only use. Next we will explain the methodology used by each of the previously mentioned automatic mixer designs.

Fixed threshold by (Shure Brothers Inc. 1978) This is a mixer whose operation is based on a gate. Gates are essentially voltage-controlled amplifiers, VCAs, whose control voltage is used to open and close the microphone channel. When the voltage measured from the microphone input signal exceeds a given threshold, the VCA passes from a minus infinite gain to a nominal gain. This has the disadvantage of having a fixed threshold, which might impede the passing of a low-level signal. In other words, there is no one single answer to the correct threshold setting. Also, depending on the attack time it will introduce artefacts or will miss the first sections of a word. This type of gated automatic mixing device is rarely seen now in conference rooms.

Gain sharing by (Dugan 1975; Dugan 1989) This design works on the premise that no matter what, the sum of the signal inputs for all microphones in the system must be below a maximum value in order to avoid reaching the maximum gain before feedback, ΔG. This type of mixer does not need to use a NOMA circuit to restrict the number of available microphones. In most cases this mixer requires a human installer to determine what is the value of ΔG.

Directional sensing (Julstrom and Tichy 1984) Directional sensing is a technology that works by opening and closing microphones based on an acceptance angle. The technology uses special microphones with 2 cardioids in a back-to-back position. The cardioid facing the speaker is connected to the mixer channel while the other capsule is used for estimating the ambient noise

and the direction of the source. An input channel turns on if there is a difference of 9.5dB between the front and back capsule inputs. If only ambient noise is present the difference in level between both capsules should be very small and will not activate the channel. The level difference of 9.5dB is derived from the fact that for a cardioid a 60 degree off axis response is 1/3 of the level of a cardioid axial response. Therefore these microphones have an acceptance angle of 120 degrees. In combination with this technology a special circuit is used to guarantee the activation of a single microphone per talker even when in the acceptance angle of multiple microphones.

Multivariable dependent (Peters 1978) This technology differs from the others in the fact that it does not only use amplitude of the incoming channels but also takes into account the time at which they take place. In this approach the instantaneous positive amplitudes of all inputs are simultaneously compared to a DC ramp threshold wave. This thresholding wave falls 80dB in 10ms or less. Initially, all inputs are attenuated. The first channel to have an amplitude value equal to the ramp value is opened for a period of 200ms. This process is repeated over and over again, renewing and opening new microphones, which are first to match the threshold signal. Every time the instantaneous amplitude of an input matches the DC ramp it gets reset to its highest amplitude to start the process again (Ballow et al. 2002). This system preserves the relative gains of the speakers since all output gains are the same. If many speakers attempt to talk at the same time the probability of their microphones being open decreases. This is not a problem since it is said that only 3 speakers can talk at a time while maintaining intelligibility (Peters 1978). However, this is not the case for music.

Noise-adaptive threshold (Julstrom and Tichy 1987; Shure Brothers Inc. 2000) This method uses a dynamic thresholds for each channel that are capable of distinguishing between signals whose frequency content and amplitude is constant, such as air conditioning noise, from rapid changing in frequency an amplitude, such as speech. The mixer will activate when the input signal is bigger than the dynamic threshold. Some other considerations can be added to

this design to ensure that a loud talker does not activate multiple inputs. This design is oriented towards speech and could cause numerous artefacts if use for music applications (Julstrom and Tichy 1987; Shure Brothers Inc. 2000).

Variable threshold by (Dugan 1975) In the case of variable threshold mixers they have a similar approach to the previously mentioned automatic mixers but with the difference that they have the ability to adapt the threshold value based on the voltage received by an external microphone. This approach needs a measurement microphone to be placed in an area in the room that is representative of the room noise contributions. This is known as adaptive gating. This system is the only current system that is designed for use in live music situations. The system is only capable of controlling gain levels and is not based on any type of perceptual attributes. Variable threshold uses gain amplitude measurements to determine the adaptive gain threshold and therefore is only capable of controlling fader gain.

2.2.3 Related work to automatic mixing

Other relevant related work includes the idea of maintain the intentions of the composer and sound engineer while providing the final user with some degree of control (Pachet and Delerue 2000). This system has the intention of providing the user with controllable parameters, which have been constrained in order to keep aesthetic intention. The system is design to work with pre-recorded material and is unable to deal with live musical sources. This system also requires human programming.

In recent years, relevant papers have been released related to this work. (Kolasinski 2008) deals with reconstructing the fader gain values of each channel through analysing a target mix. (Reed 2000) uses nearest neighbour techniques to attempt to recreate expert mixing. Finally some work on perception and automatic detection of frequencies which require equalisation compensation has been research, by (Bitzer and et al. 2008; Bitzer and LeBeuf 2009).

2.3 Going beyond the state of the art (challenges)

Automatic mixing for music is underdeveloped. Currently, automatic mixing does not take advantage of the full recalling capabilities of automated mixers and is only capable of gain control. One proposed approach to improving the current state of automatic mixing is to employ the recall capabilities of automated mixers and combine them with a decision process in order to make them perform automatic mixing operations. It is clear by now that there is room for generating a constrained rule approach to mixing based on engineering solutions. On the other hand, no simple solution exists for solving the subjective approach to mixing.

Mixing engineers make use of the information in other channels to perform signal processing on a given channel in order to achieve a harmonious musical blend of sounds between sources. For this reason a significant part of this research will concentrate on exploiting this concept. It is also clear that to have a complete approach to automatic mixing some sort of measuring mechanism is needed to determine the maximum gain before feedback in order to limit the overall output gain of the system.

Two important considerations that will affect the direction of this research and will help determining the boundaries of the research are, whether the automatic mix is static or time varying and whether the automatic mix is going to be delivered into a small or a large room. An analysis of these constraints is presented next.

2.3.1 Large room and open space mix versus small room mix

The volume of a large room, *V*, can be distinguished from that of a small room based on Manfred Schroeder's definition (Schroeder 1996; Davis and Patronis Jr. 2006) in equation 3

$$V = 4 \times 10^6 \frac{RT_{60}}{F_L^2}, \qquad (3)$$

where F_L is the large room limiting frequency, which is the frequency above which a large number of room modes will be excited to vibrate at the source frequency. For a 20Hz to 20KHZ reproduction system F_L tends to be equal to 30Hz, and RT_{60} is the time, in seconds, it takes the amplitude of the source to decay 60dBs.

In the context of this research, if a room is small there is no need to amplify high sound pressure level instruments, like drums or trumpets. It is only necessary to reinforce the acoustic sources with a low sound pressure level. For this reason the mix inside the mixing console might be completely different from the acoustic mix heard by the audience. On the other hand, for a large room, a large open space, or headphones, we can approximate the mix happening inside the mixing board to be similar to the acoustic mix being delivered.

In the case where the tracks are pre-recorded we can take advantage of some, but not all, large room acoustic assumptions and apply them to a small room regardless of the size. This is because the real sources are not interacting with the acoustics of the room, giving us an extra degree of mixing freedom. For this reason, this thesis will concentrate mainly on automatic mixing for large rooms.

2.3.2 Static versus time varying mix

The way automation is currently implemented is by using either scenes or by using automation tracks. Automation tracks are the equivalent of a time-line representation of the state of a parameter and are common in digital audio workstations. It is a time description of all the parameters in the console. The scene approach is a snapshot representation of a mix and it is recalled at a

certain point in time. This facilitates changing the state of the mixing parameters dynamically. There is usually some sort of interpolation mechanism between snapshots to reduce any artefacts introduced when recalling a scene.

For recording, the mixing of a single song is comprised of several automation scenes, while for live performances the tendency is to have a smaller number of scenes. In general, for live mixing, a steady static mix is built first and then the mixing engineer enhances it manually during the show. In current systems the scenes are pre-recorded, therefore the programming of the scene requires prior knowledge of the input signal and must be re-programmed if the input sources are to be changed.

2.4 Summary

In this thesis we will put emphasis into generating an acceptable autonomous adaptive static mix, which can later be enhanced by a human operator if desired. The proposed approach differs from current approaches in that it requires minimal or in most cases non-prior knowledge of the inputs. The proposed method is able to be adapted to different input with out the need of extensive human intervention. The system proposed takes an automatic approach to mixing by reducing user interaction and taking into account minimal or no aesthetic considerations and is meant to work in real time for the purpose of live mixing of music. The proposed system is envisaged as a helper to the sound engineer and composer rather than giving the engineer an additional set of constraint parameters to be manipulated. The proposed approach seeks to enhance the user experience by automatically mixing the input sources while reducing or eliminating the technical mixing tasks required to be performed by the user, this allows the engineer to concentrate on the aesthetics of the mix. Next we propose methods for automatic mixing tools for live music applications.

Part II Automatic mixing tools for music

Chapter 3
Automatic mixing building blocks

From the point of view of signal flow, an audio mixer is composed of several chained audio processing effects. Currently digital audio mixers are composed of a series of digital audio effects with recallable control parameters. Each individual digital audio effect is a device that takes an un-processed input signal and outputs a processed signal. In most cases the user can control the signal processing behavior by manipulating a number of control parameters through a graphical user interface. The aim of the user is to manipulate the signal processing parameters in order to produce the desired transformation of the input signals. Figure 2 shows the standard implementation of an audio processing device, where *x(n)* is the input source and *y(n)* is the output resulting from the signal processing given that *n* denotes the discrete time index in samples.

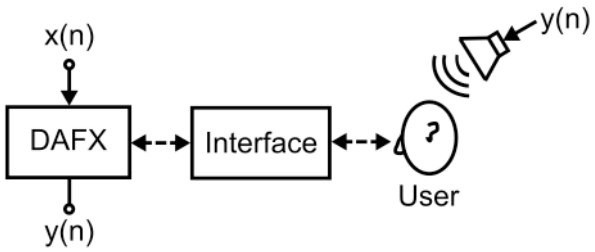

Figure 2 Diagram of an audio effect and a user.

In an automatic mixing context we aim to aid or replace the task normally performed by the user. In order to achieve this some important design objectives should be performed by the automatic mixing tools:

1. The system should comply with all the technical constraints of a mix, such as avoiding distortion and maintaining adequate dynamic range.

2. The design should simplify complex mixing tasks while performing at a standard similar to that of an expert user.

3. For sound reinforcement applications, such as live music or live performance inside an acoustic environment, the system must remain free of undesired acoustic feedback artefacts.

Our aim is to emulate the user's control parameters. An automatic mixing tool is formed of two main sections; the signal processing section and the side chain-processing portion. The signal-processing algorithm is a standard audio effect-processing device and can include a user interface if the automatic mixing tool is meant to give visual feedback for its actions. The analysis decision section of the automatic mixing algorithm is what we will refer as the side chain processing. The analysis decision-making portion of the automatic mixing tool takes audio from one or more channels together with optional external inputs, and outputs the derived control data. The controlling data drives the control parameters back to the signal-processing algorithm. A diagram depicting a generic automatic mixing tool can be seen in Figure 3, where $x_e(n)$ is an external source.

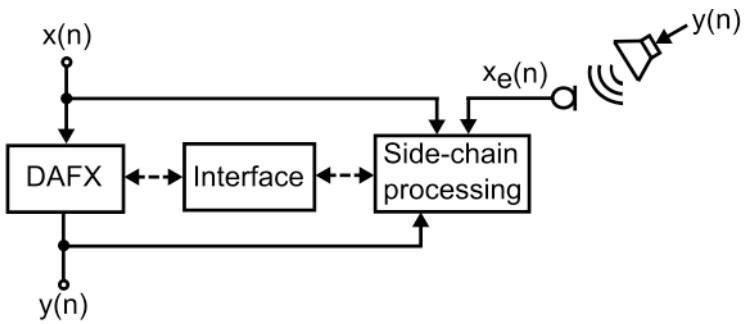

Figure 3 Diagram of generic automatic mixing tool.

This cycle of feature extraction analysis and decision making process, presented in Figure 3, is characteristic of adaptive effects and therefore characteristic of automatic mixing tools.

3.1 Adaptive effects

Audio processing effects architectures have been classified by their implementation (Zölzer 2002; Zölzer 1997); filters, delays, modulators, time-segment processing, time-frequency processing, etc. Similarly, audio processing effects have also been classified by the perceptual attributes (Amatrian and et al. 2003) which they modify timbre, e.g. delay, pitch, positions or quality. Although these classifications tend to be accurate in many contexts, they are not optimal for understanding the signal processing control architectures of some more complex effects. More recently, an adaptive digital audio effect class was proposed (Verfaille 2006).This class uses features extracted from the signals to control the signal processing process. In terms of their parameter control properties, digital audio effects may be distinguished as follows:

Direct user control Features are not extracted from input signals so these are non-adaptive. A multi-source extension of this approach is the result of linking the user interface, for example when linking a stereo equalizer. This provides exactly the same equalization for the left and right channel using a single user panel. Although the user interface is linked, the output signal processing is independent of the signal content. Such implementation has been depicted on Figure 2.

Auto-adaptive Control parameters are based on a feature extracted from the input source. These include, for example, auto tuning, harmonizers, simple single channel noise gates and compressors. This type of signal processing device can also use feedback from the output signal. A generic auto-adaptive processing device has been depicted in Figure 4.

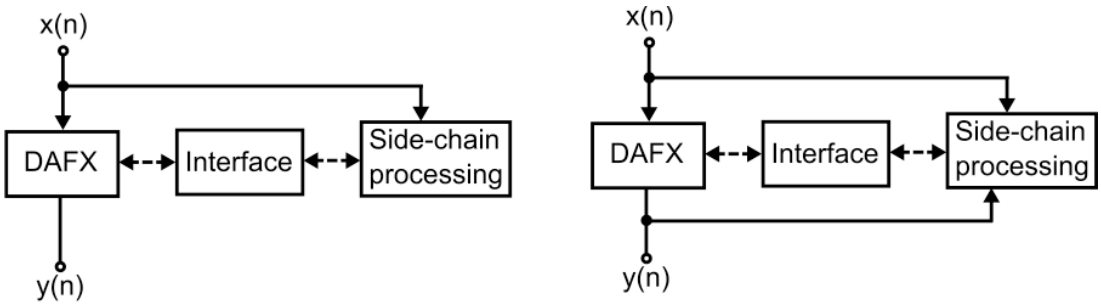

Figure 4 Diagram of an auto-adaptive processing device without feedback (left) and diagram of an auto-adaptive processing device with feedback (right).

External-adaptive The system takes its control processing variable from a deferent source to the one on which it has been applied. It is a feed-forward external adaptive effect if it takes its control variable from the input, and it is called a feedback external adaptive effect if it takes its control feature from the output. This is the case for ducking effects, side chain gates and side chain compressors. Feedback implementations of this approach can also be implemented. A block diagram of an external-adaptive is presented in Figure 5.

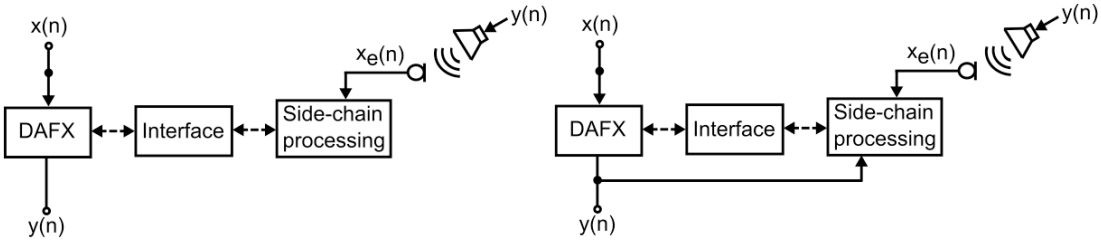

Figure 5 Diagram of an external-adaptive processing device without feedback (left) and diagram of an external-adaptive processing device with feedback (right). Where $x_e(n)$ is the external source.

Cross-adaptive effects Signal process is the direct result of the analysis of the content of each individual channel with respect to the other channels. The signal processing in such devices is accomplished by inter-source dependency. This type of signal processing device can also be enhanced using a feedback loop and the use of external inputs. This control approach to audio processing gives the greatest design flexibility and it generalizes the adaptive processing control of effects given that by removing or adding sections it can conform to any of the previous adaptive topologies previously explained. Therefore we will have a deeper look at them in the next subsection.

3.2 Cross-adaptive methods

When mixing audio, the user tends to perform signal processing changes on a given signal source not only because of the source content but also because there is a simultaneous need to blend it with the content of other sources, so that an overall mix balance is achieved. There is a need to be aware of the relationship between all the sources involved in the audio mix. Thus, a cross-adaptive effect processing architecture is ideal for automatic mixing.

Due to the importance of the source inter-relationships in audio mixing for music, we can add another design objective to be performed by the automatic mixing tool:

4. The signal processing of an individual source is the result of the interdependent relationships between all involved sources.

This objective could be met by the use of cross-adaptive processing. A cross-adaptive process is characterized by the use of a multi-input multi-output (MIMO) architecture. This thesis will make use of MIMO systems that have the same number of input and outputs unless stated. We will identify inputs as $x_m(n)$ and outputs as $y_m(n)$, where m has a valid range from 0 to $M-1$ given that M is the maximum number of input sources involved in the signal processing section of the automatic mixing tool. External sources are denoted $x_e(n)$. The general block diagram of a cross-adaptive device is depicted in Figure 6.

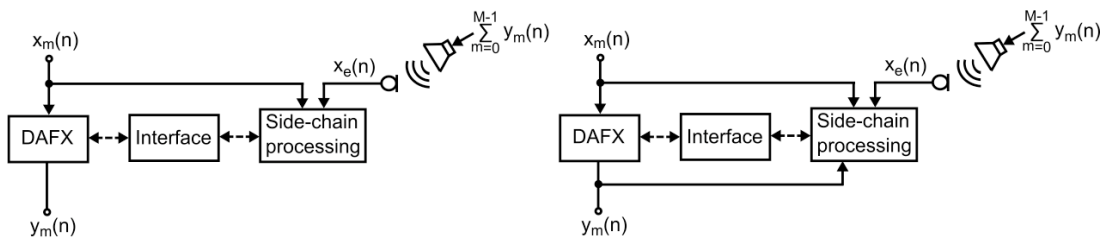

Figure 6 General diagram of a cross-adaptive processing device without feedback and external input (left) and diagram of a cross-adaptive processing device with feedback and external input (right). Notice how the index *m* denotes multiple channels involved in the process.

During this thesis we will use an architecture that does not make use of feedback. Therefore the side chain processing inputs will be taken only from the input of the signal processing section of the automatic mixing tool. Feedback structures remain a field of future research exploration.

3.3 Side chain processing

Due to the complexity of the signal processing involved in most expert systems, it would be almost impossible to think that true real time processing can be achieved. On the other hand for a live performance application a real time signal processing flow is required. For this reason, side chain processing can be performed. What this means is that the audio signal flow can happen in a normal manner in the signal-processing device, such as a digital recallable mixer, while the required analysis, such as feature extraction and classification of the running signal, is performed in a separate analysis instance. Once the correct amount of certainty is achieved on the analysis side then we can proceed to send a control signal to the signal processing side in order to trigger the desired parameter control change command. In a cross-adaptive automatic mixing tool the side chain consists of two main sections:

1. A feature extraction processing section.

2. A cross-adaptive feature-processing block.

In the side chain the feature extraction vector for all sources, obtained from the feature extraction processing section, will be denoted $fv_m(n)$. The control data vectors for all sources, obtained from the cross-adaptive feature processing block, will be denoted as $cv_m(n)$. The detailed block diagram of a cross-adaptive device is depicted in Figure 7.

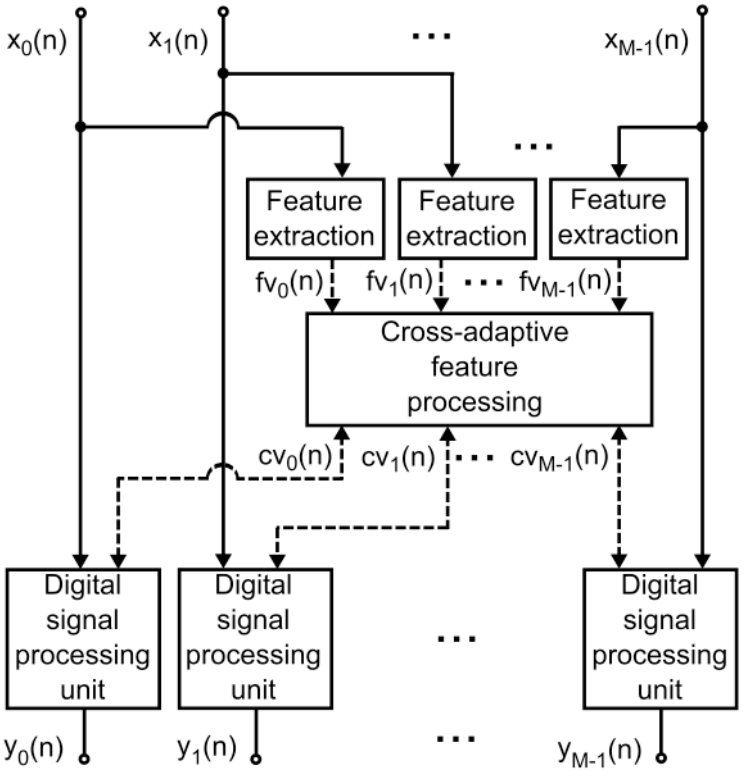

Figure 7 Detailed general diagram of a cross-adaptive device using side chain processing.

It is of crucial importance for the rest of this thesis to understand that the feature vector $fv_m(n)$ will correspond to different features in each chapter, for example in chapter 6 it will correspond to a time delay value. While in chapter 7 it will consist of a spectral decomposition classification feature. In the same manner the control vector $cv_m(n)$ will correspond to different parameter according to the chapters control parameters, for example in chapter 6 it is a multidimensional vector which contains a polarity and delay control parameter while in chapter 7 it consists of an attenuation parameter.

3.4 Feature extraction processing

The feature extraction-processing block is in charge of extracting a series of features per input channel. The ability to extract the features fast and accurately will determine the ability of the system to perform appropriately in real time. The better the model for extracting a feature, the better the algorithm will perform. For example if perceptual loudness is the feature to be extracted, the

model of loudness chosen to extract the feature will have a direct impact on the performance of the system. According to their feature usage, automatic mixing tools can be in one of two forms.

Accumulative This type of automatic mixing tool aims to achieve a converging data value which improves in accuracy with time in proportion to the amount and distribution of data received. The system has no need to continuously update the data control stream, which means that the accumulative automatic mixing tools can operate on systems that are performing real time signal processing operations, even if the feature extraction process can be non-real time. The main idea behind accumulative automatic mixing tools, as implemented herein, is to obtain the probability mass function (Johnson et al. 1993) of the feature under study and use the most probable solution as the driving feature of the system. In other words we derive the mode, which corresponds to the peak value of the probability density of the accumulated extracted feature.

Dynamic This type of automatic-mixing tool makes use of fast extractable features to drive data control processing parameters in real time. An example of such a dynamic system can be a system that uses an RMS feature to ride vocals against background music. Another example can be gain sharing algorithms for controlling microphones such as the one originally implemented in (Dugan 1975). Dynamic automatic-mixing tools do not tend to converge to a static value.

A compromise between dynamic and accumulative feature extraction can be achieved by using relatively small accumulative windows with weighted averages.

3.5 Feature extraction

A feature of a signal is a characteristic measured or extracted from a signal. These features can be low level, meaning they have little or no correlation with human perceptual relation, such as a RMS measurement. Such low level

attributes dictate the electronic constraints and technical limitations of an audio device. Higher-level features are those that represent a human perceptual concept.

3.5.1 Feature extraction with noise

An important consideration to be taken into account during the feature extraction process is noise. The existence of bleed, crosstalk, self-noise and ambient noise will influence the reliability of the feature extraction. Common methods for obtaining more reliable features include averaging, coherence validation and gating.

One of the most common methods used for automatic mixing is adaptive gating, where the gating threshold adapts according to the existing noise. This method was introduced to automatic mixing applications by (Dugan 1975; Dugan 1989). It requires an input noise source that is representative of the noise in the system. In the case of a live system a microphone outside of the input source capture area is a good representation of ambient noise. Therefore this microphone signal can be used to derive the adaptive threshold needed to validate the feature. This gating process has been depicted next, in Figure 8.

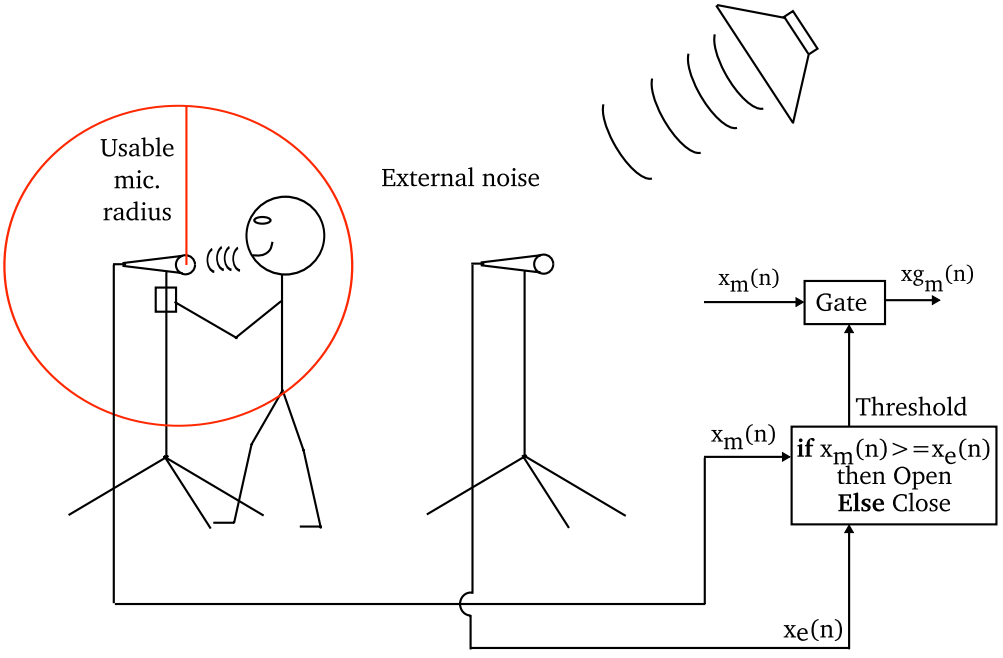

Figure 8 Diagram of an adaptive gated system.

Although automatic gating is generally applied to gate an audio signal it can also be implemented on the data features extracted from the signal as opposed to directly applied to the signal. This has the advantage of being less processing intensive.

For accumulative automatic mixing tools, variance threshold measures can be used to validate the accuracy of the probability mass function peak value. The choice of feature extraction model will influence the convergence times in order to achieve the desired variance. For this to work appropriately in a system that is receiving an unknown input signal, in real time, some rescaling operations must be undertaken. If the maximum dynamic range of the feature is unknown the probability mass function must be rescaled. In such a case, the axis range should be normalized continuously to unity by dividing all received feature magnitudes by the magnitude of the maximum received input value. An example of the effect of adaptive gating and rescaling in an accumulative feature extraction block is shown in Figure 9. In this example the feature under study is loudness, which has been extracted from a musical test signal. If no re-scaling and no adaptive gating is used to optimise the loudness probability mass function, the resulting most probable feature value is always 0, as shown in Figure 9A. This is because there is a large amount of low-level noise that biases the loudness measurement. A second test with rescaling and no adaptive gating is shown in Figure 9B. It can be see that although a Gaussian shape corresponding to the actual loudness can be seen, there are still a large number of data points in the lowest bin of the histogram, causing an erroneous null measurement. When adaptive gating is performed without rescaling, Figure 9C, the number of zero- bin occurrences is dramatically reduced. Finally, a test consisting of both rescaling and adaptive gating is depicted in Figure 9D. It can be seen that the algorithm is able to correctly identify the most probable feature value. This means that that both adaptive re-scaling and gating must be performed in order to achieve accurate extraction of the most probable feature value.

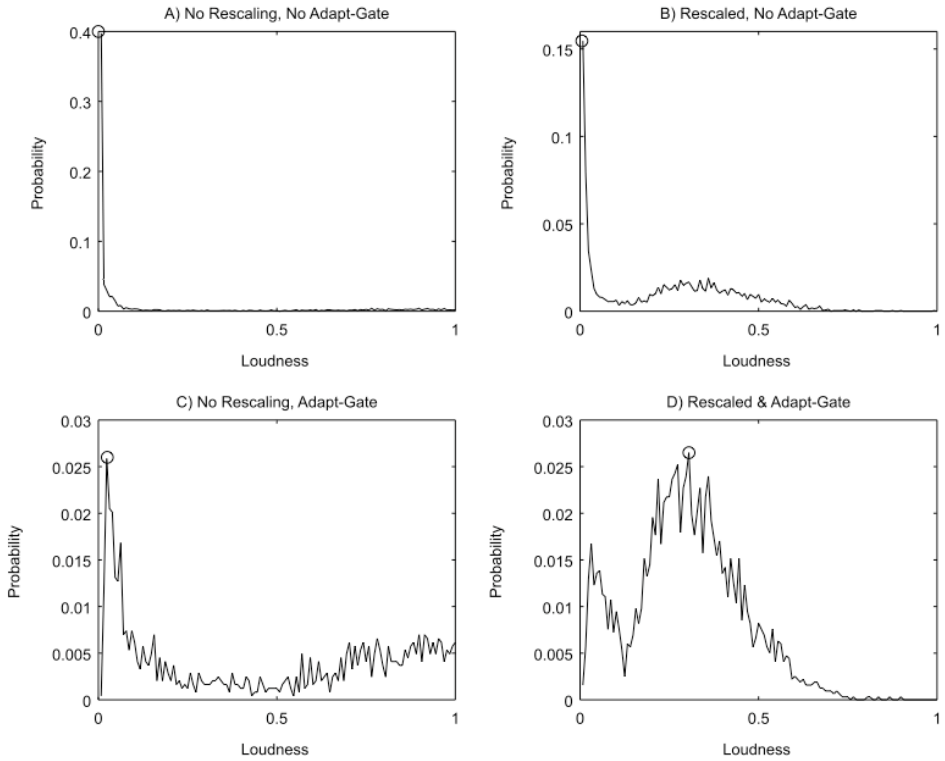

Figure 9 Accumulated histograms. The circular marker denotes the resulting accumulated peak loudness value.

3.6 Cross-adaptive processing

The cross-adaptive processing section of the automatic mixing tools is in charge of determining the interdependence of the input features in order to output the appropriate control data. This data control parameters in the signal processing section of the automatic mixing tools. The obtained control parameters are usually interpolated before being sent to the signal-processing portion of the automatic mixing tools. This can be achieved using a low-pass filter that will ensure a smooth interpolation between control data points. The cross-adaptive feature processing can be implemented by a mathematical function that maps the interdependence between channels. In many cases constraint rules can be used to narrow the interdependency between channels. In order to keep the cross-adaptive processing system stability the overall gain contribution of the

resulting control signals can be normalized so that the overall addition of all source control gains is equal to unity. The cross-adaptive function is unique for every design, and has to be individually derived according to the aim of the automatic mixing tools. For example if our mixing objective was to achieve equal level for all channels, where $fv_m(n)$ is the level per channel, and the processing per channel is given by $y_m(n) = cv_m(n)x_m(n)$. Where $y_m(n)$ is the scaled output per channel, $x_m(n)$ is the input per channel and the control variables we are looking for are given by $cv_m(n)$. A simple cross-adaptive process could be given by $cva_m(n)$=mean($fv(n)$)$/fv_m(n)$ where $cva_m(n)$ would give us the control variables, which ensures equal level for all channels. We can further normalize the control factor by $cv_m(n) = cva_m(n)/$sum($cva(n)$).

3.7 System stability

In the case of a system used for live performance mixing, the automatic system must avoid undesired acoustic feedback artefacts at all costs. For this reason several stability solutions have been developed, for example gain sharing (Dugan 1975; Dugan 1989) and self normalization techniques. In most cases these techniques try to prevent acoustic feedback by ensuring a maximum electronic transfer function gain no bigger than unity. This ensures that regardless of the changes in signal processing parameters the system remains stable. Given the importance to stability of the automatic mixing system the next chapter will be dedicated solely to stability of the automatic mixing tools.

3.8 Perceptual processing and technical constraints

In standard signal processing design devices are developed mainly using low level features. This is because low level features such as RMS and peak levels characterize the technical limitation of electronic devices. For mixing musical signals the audio engineer's tries to make the right balance of technical and perceptual attributes. Therefore, achieving a perceptual goal within a technically constrained universe is the goal of the audio mixer. A graphic explanation of such a phenomena has been depicted in Figure 10.

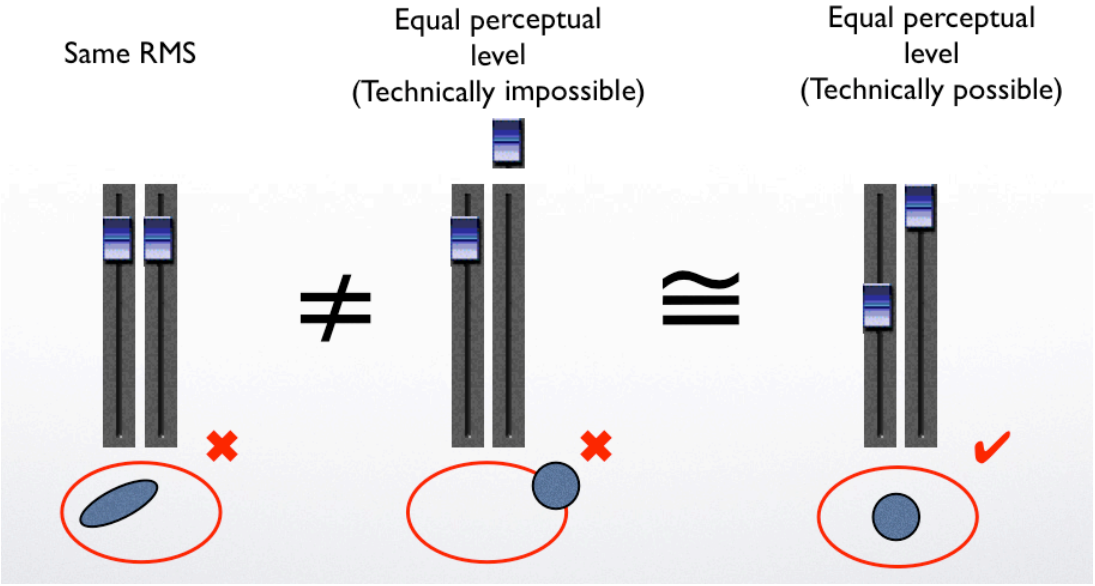

Figure 10 Technical limits of the device in red oval circle. Blue circle represents position of the perceptual attribute. Left image is an example of a non-perceptual equal low-level feature. Middle image is an equivalent perceptual setting but technically impossible. Right image is a perceptually balanced and technically possible solution.

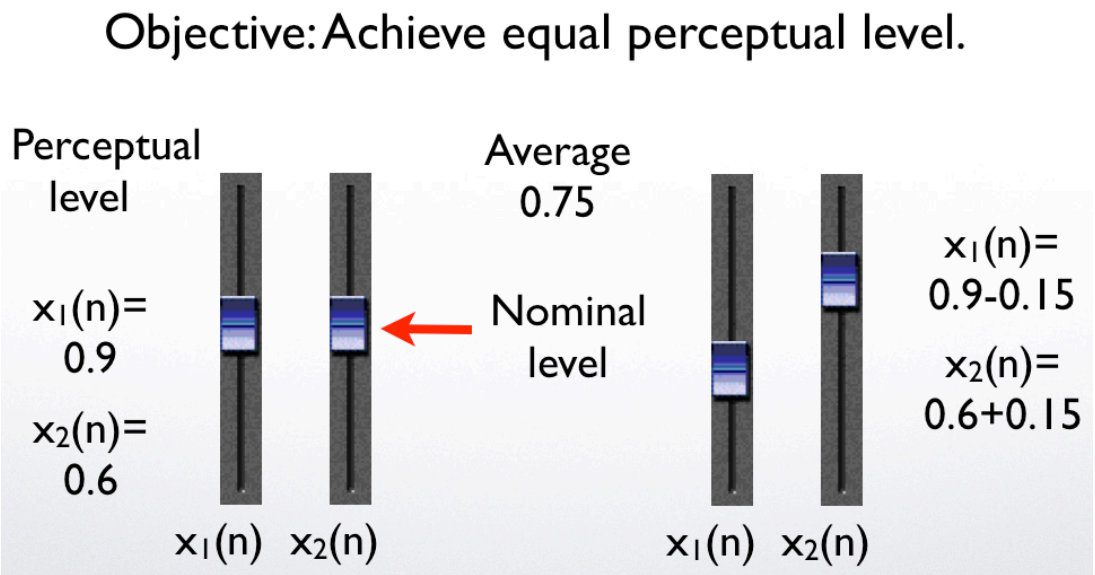

Figure 11 Achieving equal perceptual loudness by average normalization. Left, perceptually unbalanced mix. Right, perceptually balance mixed kept within technical possible range thanks to the average normalization technique.

Given that we have a reliable set of perceptual features extracted from the input signal, a simple but yet elegant solution, for achieving a balance between both the technical and perceptual space can be achieved by average normalization. This basic average normalization process has been depicted on Figure 11. When average normalization is used with a perceptual feature it can be used to balance the ratios of the perceptual feature evenly. Such a method can also be used with low level features to maintains unity gain of the feature being normalized. Therefore, when used with a low level feature such as gain it can be used to keep a system under stability therefore avoiding unwanted acoustic feedback artefacts. A more advance normalization method such as the one presented here will be discuss on chapter 4.

3.9 Summary

The automatic mixing tools, described in this thesis, aim to take objective technical decisions. This is useful for improving the audio engineers work flow and allowing him to achieve a well balanced mix in a shorter period of time. The automatic mixing concepts described in chapter 3 are not designed to take into account any uncommon mixing practices or to be able to take subjective mixing decisions. In order to optimize the design of automatic mixing tools the use of common mixing practices can be used as constraints. Given that the task normally performed by an expert user also involves perceptual considerations, perceptual rules can improve the performance of the algorithms. When combining several basic automatic mixing tools to emulate the signal path of a standard mixer, we can achieve a mix in which the signal processing flow is comparable to the one performed in a standard mixing situation.

A set of mixing tools, which make extensive use of the concepts presented on chapter 3, will be presented next. We will start by presenting a normalization concept, which allows the automatic mixing tools to perform changes in the signal processing without the fear of introducing undesired

artifacts. The automatic mixing tools are presented in the order they would be found on a standard multichannel audio mixer, following the natural signal flow from input to output. With the notable exception of the automatic equalizer, which is presented last due to its methodology relationship with the automatic faders, presented second to last. Therefore, we proceed to introduce a method for automatically maintaining system stability of audio mixers by using automatic gain normalization.

Chapter 4
Automatic gain normalization

4.1 Introduction

Public addressing systems that use a microphone amplifier speaker chain to transmit sound through the air towards the listener are essentially a feedback system. Using the air as a propagation medium has the inevitable effect of turning the sound reinforcement system into an endless feedback loop, and it is the air itself that acts as a feedback path. This is an inherent property of a sound system, and must be taken into consideration when designing or interacting with the system. The design aim is to reduce audio artefacts due to the feedback path. With this goal in mind, it is the purpose of this chapter to introduce a normalization technique that prevents feedback when interacting with an audio system. The proposed method automates the engineering task of continually revising the system gain structure in order to avoid undesired feedback artefacts. This method permits one to achieve maximum gain before feedback while realizing the technical constraints of the mixing engineer, thus permitting him to concentrate more on the aesthetic contributions of the mix. The method permits the audio mixing engineer to interact with the system without the fear of introducing feedback. The algorithm uses an impulse measurement of a mathematical model of the system to automatically calculate the appropriate gain compensation to avoid undesired artefacts due to feedback.

4.2 Feedback background

Feedback is the return to the input of a part of the output of a machine system or process (Davis and Patronis Jr. 2006). In an audio environment where active system is consists of a microphone, amplifier and speaker there is always feedback occurring through an acoustic path, the problem occurs when this feedback causes a growth in gain which causes undesired distortion or "ringing". A simplified model of an acoustic feedback system is presented next in Figure 12.

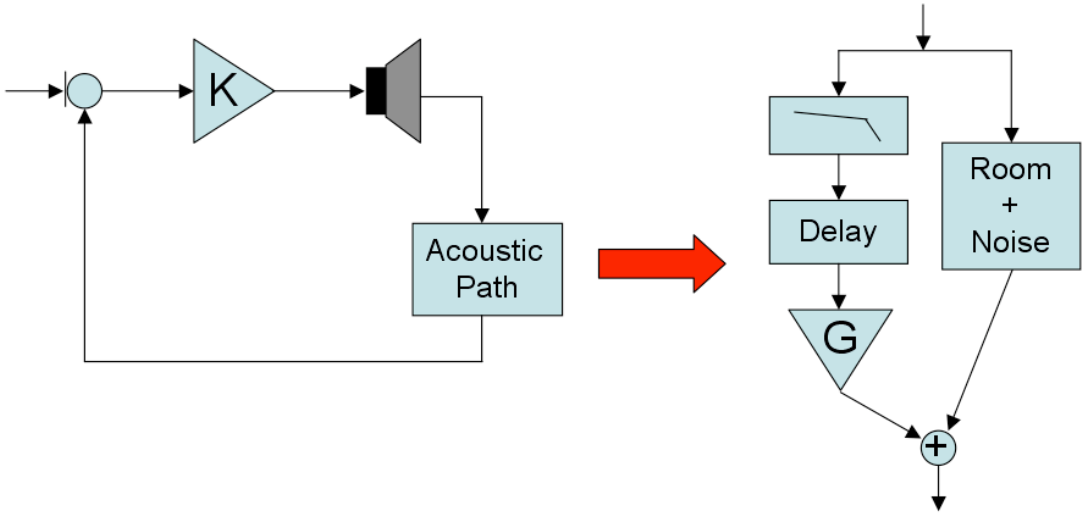

Figure 12 Acoustic feedback systems, and an equivalent acoustic path model.

The acoustic path can be modeled as a Low Pass Filter (LPF), a delay and an attenuation factor. The LPF is the result of high frequency attenuations due to atmospheric absorption; the further the sound travels the more high frequencies are attenuated. The delay represents the distance from the speaker to the microphone. The delay path is dependent on the speed of sound, and thus on temperature and humidity. This means that a change in temperature will change the delay time. Next is presented the equation for calculating the speed of sound, *c*, for a given temperature in Celsius (T_c).

$$c = 332.4 + 0.6T_c \qquad (4)$$

Finally the gain term represents the amplitude attenuation due to the inverse square law of sound. The inverse square law states that Sound Pressure Level decreases by half for every doubling of the distance. The equation for calculating SPL in dB_{spl} is as follows:

$$dB_{SPL} = 20\log\left(\frac{R_1}{R_2}\right) \qquad (5)$$

Where R_1 is the reference distance and R_2 is the distance in meters to where we want to calculate the SPL drop.

The instability in the feedback loop occurs when the amplitude of the feedback is equal or greater than the nominal amplitude of the input. The phenomenon is commonly known in the audio industry as “acoustic feedback”, correctly named howlback (Antman 1965) or Larsen effect (Rombouts et al. 2006; Dacht 2008). Feedback is not only an amplitude problem but it is a phase problem as well (Troxel 2005). If we model the delay (τ) in the acoustic path as a pure delay we would introduce a tilt in the phase of the system $(\Delta\phi)$, which will reflect accordingly on each frequency (f) of the spectrum.

$$\tau = \frac{-\Delta\phi}{360 f} \qquad (6)$$

Based on this we can affirm that only the frequencies that add in a constructive manner, between the input and the feedback path will produce feedback ringing. This means that every time the phase of the transfer function of the system is a multiple of 360 degrees or 0 degrees the summation of the input and the feedback signal is a totally constructive interference. For this reason we can say that the potential frequency spacing of the feedback is given by:

$$f = \frac{1}{\tau} \qquad (7)$$

This means that a change in temperature, which causes a delay change, will cause a change in the frequencies that are susceptible to producing feedback.

A change in delay also alters the rate (ρ) at which the feedback grows and decays. For a system where unity gain is 0dBs, the following equation can be used to calculate the rate at which the feedback will grow or decay in response to the transfer gain ($|H|$) increment or decrement.

$$\rho = \frac{|H|}{\tau} \qquad (8)$$

If the system gain is raised above 0dBs, for a potential feedback frequency, the feedback will grow proportionally to the gain applied. If the gain is put back to a value below nominal the system feedback will begin to decay in a proportional manner to the transfer function gain.

4.2.1 Current feedback elimination approaches

To maximize the acoustic gain while avoiding feedback, the system should have a flat frequency response that falls below the threshold for acoustic feedback. Figure 13 shows the acoustic measurement of the frequency response of an audio system before and after optimization. The 0dB mark represents the threshold before feedback. The area between the frequency response and the 0dB mark represent unused system gain. It is the goal of an audio system engineer to minimize this unused area by flattening the frequency response of the system. This ensures a system with no coloration with the added benefit of maximizing gain before feedback. To achieve maximum gain before feedback audio operators have relied mainly on equalizers (Davis and Patronis Jr. 2006), delay and feedback cancellation techniques.

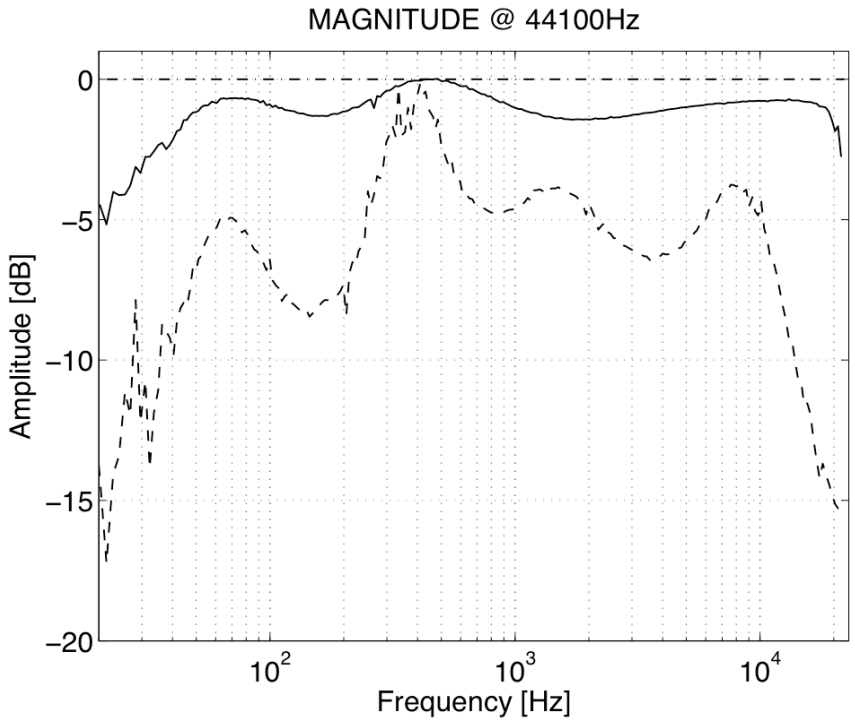

Figure 13 Acoustic measurement of the frequency response of a audio system. The dash-dotted (-·-·-) line represents the threshold for maximum gain before feedback, the dashed line (- - -) represents the frequency response of a non-optimised acoustic system and the full line (——) is the frequency response of an optimized quasi-flat system.

Measurement techniques like, time delay spectrometry (Cable and Hilliard 1980) and source independent measurements (Meyer 1984) have become more widely available, making the use of equalizers and delay lines more of a technique rather than a matter of skill. Also current design techniques and modern electronics, acoustics and speaker technology make a flatter frequency response a reality, such as the corrected response presented in Figure 13. Although the proper design of audio systems still requires a great amount of knowledge from the system engineer, a close to flat frequency response system which maximizes the gain before feedback is now a reality. The process of achieving this is commonly known in the industry as aligning, in time and frequency, a system. The full details of this process are beyond the scope of this thesis but more on this can be found in (McCarthy 2007).

The other important method for achieving maximum gain before feedback is by the use of feedback cancellation. Currently there are four main feedback-controlling techniques (Dacht 2008). The first one consists of slightly frequency shifting the output signal so that the electronic transfer function is out of alignment with the acoustic transfer function. This causes a destructive interaction between the input and the acoustic feedback path, which effectively reduces feedback. In practice it can achieve up to 3 dBs increase in gain before feedback. This method is effective for speech applications but is not suitable for music. This is due to the simple reason that it modifies pitch, which would result in undesired atonal music.

The second feedback control technique is the all-pass filter approach. This is used to invert the phase of a potential feedback frequency. Unfortunately this technique is only useful with low delay systems with a prominent resonance. When applied to a system with flat frequency response it causes the feedback to jump endlessly from one section of the spectrum to other. For this reason its use is very limited.

Third is the adaptive filter modeling (Kamerling and et al. 1998). This uses technology based on echo-cancellation, aimed at telecommunication applications. The main idea is to subtract the far end speech from the near end speech. When the model is accurate it can achieve up to 10dBs of added gain before feedback. Due to the closed loop nature of the acoustic audio system the residual error of this process is highly correlated to the signals involved, and this can cause noticeable artefacts. When the model deviates it can introduce unwanted distortion. It can even cause undesired acoustic feedback, which should not have been there. For this reason it has mainly been applied for speech systems where conditions are controlled. It is currently not consider a good candidate for sound reinforcement.

Finally, there is the adaptive notch filter method (Troxel 2005), which consists of a series of fixed and non-fixed notch filters, which filter out feedback frequencies when detected. The system performance is a trade-off between speed of detection and accuracy, and can notch out program material if a feedback discrimination system is not implemented properly or the system is overused. This method is highly effective and is widely used on sound reinforcement applications. Unfortunately it does not offer any extra gain before feedback for a flat frequency response system.

Currently, there is no optimal feedback cancellation method for music which offers a substantial improvement in gain before feedback without dangerous side effects. Therefore, it is our belief that if system alignment and an acoustic flat frequency response are currently achievable, then there is little need for feedback cancellation techniques. For this reason, we present in this chapter a normalization technique that helps preserves system stability rather than another feedback cancellation technique. The aim is to prevent howling before it happens rather than suppress it after it has happened.

4.3 Understanding feedback from a transfer function perspective

Feedback is the result of a retro-alimentation of the output signal of a system to its input. In an acoustic system these artefacts are introduced due to the feedback path and can be positive and negative feedback contributions. A simplified diagram of an acoustic feedback system is presented in Figure 14. The source signal is picked up by a microphone, transformed by equalization, amplified and played back through a speaker at the output. This is then attenuated and delayed as the output is transmitted through air, and summed with the input signal. $H_E(n)$ is the electronic feed-forward transfer function of the system, and it is the result of the product of the individual transfer functions of the signal chain given by the microphone equalizer amplifier and speaker. $H_A(n)$ is the acoustic transfer function of the system.

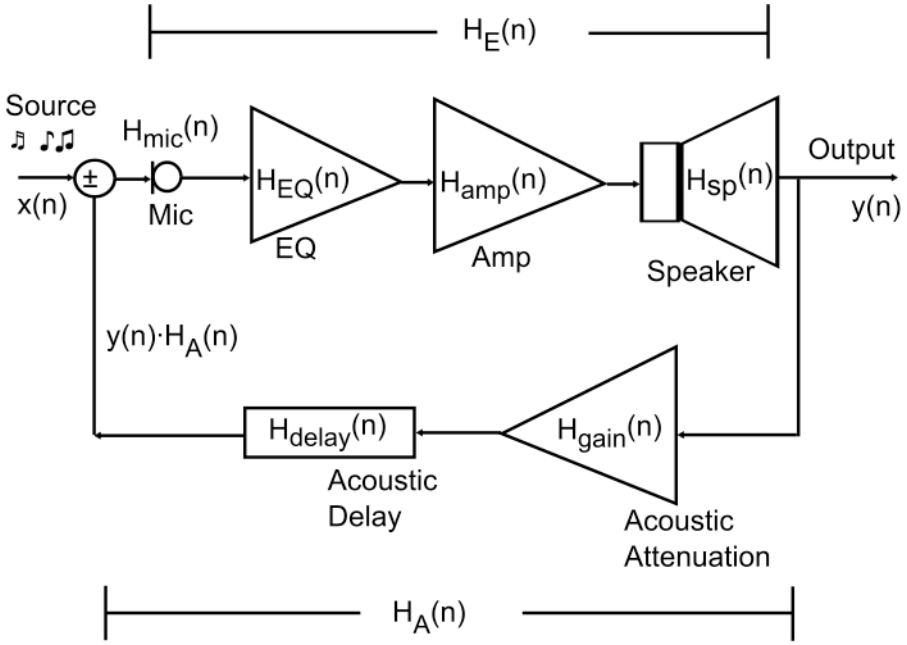

Figure 14 Model of a sound reinforcement feedback system.

For this thesis we will only be concerned with undesired feedback phenomena. This is a state in which system gain exponentially increments out of control, causing an undesired audible pitch. The feedback causes the audio system to behave in an unstable manner. Therefore, this condition must be avoided at all cost. Given the acoustic model in Figure 14, the system will introduce undesired howling artefacts if equation 9 is satisfied.

$$H_E(n)H_A(n) >= 1 \qquad (9)$$

If, for example, the equalizer transfer function gain, $H_{EQ}(n)$, is 0dBs when flat and the overall electronic transfer function of the system $H_E(n)$ is on the marginal condition before howling, then boosting the equalizer will introduce an undesired feedback artefact, and performing a cut on the equalizer will permit the system to remain stable. Therefore, a normalization technique that enables relative gain changes while forcing the transfer function of a linear system to have a maximum peak of 0dBs will preserve the stability of the system.

4.4 Real time transfer function normalization

Normalization of a signal consists in dividing the output by a given constant. In our case we are interested in normalizing the output signal of a linear system with the aim of keeping its overall maximum gain to be one, or 0 decibels full scale (dBFs). For this the normalization constant will correspond to the inverse of the maximum of the transfer function of the system under study. Such a normalization system has a power reduction proportional to the normalization constant. The goal of the methods presented in this and the following subsections is to find the maximum of the transfer function in order to normalize the system. In subsections 4.4.1 and 4.4.2, two normalization methods will be discussed and their advantages and disadvantages will be analyzed. In section 4.5, we will propose an alternative normalization technique.

Finding the maximum value of a transfer function composed of multiple elements, such as a parametric equalizer composed of multiple varying filters, is not a trivial task. Even if one knows the individual maxima of each component of the transfer function (such as through a parallel or series decomposition), their interaction can result in a maximum located at a completely different location. Given that the user can change the coefficients at any time to adjust the processing system, for example to modify an equalization filter, it becomes an even more challenging problem. In fact, the location and magnitude of the maximum of the transfer function is the result of the complex interaction of simpler transfer functions with each other. Therefore this involves both phase and amplitude interactions.

4.4.1 Mathematical normalization approach

Given that the coefficients of the transfer function can be changed by the user at all times, a familiar approach to finding the maximum is finding the analytical

solution of the roots of the first derivative of the transfer function. This approach requires a discrimination process in order to separate the local maxima from the global maximum. The steps for performing such an approach are presented next:

Given a Laplace domain transfer function:

1) Substitute terms so that the transfer function is in terms of the frequency.
2) Calculate the derivative with respect to the frequency.
3) Find the roots for the result obtained on step two.
4) Solve the roots and discard all results but the largest number.

Once the maximum has been found, the input is then divided by this maximum amplitude in order to maintain the system under unity gain. This method has the advantage that it can be implemented at clock speed rather than at sampling rate speed. It is highly effective for simple transfer functions, but unfortunately for most complicated cases, such as a transfer function representing a six filter parametric equalizer, it becomes practically impossible to find the exact analytical result for the roots. Thus, this approach is limited to static coefficients or to a more elaborate mathematical approximation. Such advanced mathematical approaches must be tailored to each particular case of linear system under study. In many cases, this means re-implementing the complete normalization design.

4.4.2 Real time transfer function measurement normalization

A more general solution to the normalization problem is to measure the transfer function of a linear system such as the one depicted in Figure 15 using a source independent measurement algorithm. This approach has the advantage of working for all linear systems without the need of re-implementation for more complex systems.

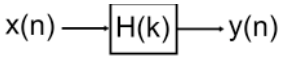

Figure 15 Model of a linear system.

The exact transfer function of the system in Figure 15 is given by dividing the Laplace transform of the output by the Laplace transform of the input. In digital source independent measurement, the transfer function *H(k)* is approximated by dividing the Fast Fourier Transform (FFT) of the output by the FFT of the input, equation 10. The approximation is due to the finite size of the FFT frame. Further improvements to this approximation are presented in (Meyer 1992).

$$H(k) \cong \mathrm{FFT}[y(n)] / \mathrm{FFT}[x(n)] \qquad (10)$$

To use such measurement an algorithm implementation such as the one shown in Figure 16 is needed. In this implementation, the source independent measurement algorithm performs a continual reading of the input and the output and performs a division of its corresponding FFT frames synchronized in time. The result is post processed to improve accuracy and finally a maximum peak detector is used to determine the transfer function maximum. The inverse of this maximum value is then used to multiply the input in order to maintain the system under unity gain.

Unfortunately this approach has to be implemented at a sample rate speed, which makes the algorithm slower than a purely mathematical implementation. Also in order for this algorithm to give a precise measurement a number of frames must be averaged, and coherence and threshold techniques are required before calculating the maximum peak. All of this can be overcome, to some extent, by compromising precision and by algorithm optimization. Lack of precision will translate into a peak measurement that is non-stable and will cause the input to be modulated, introducing undesired audible artefacts. On the other hand a slow performance may cause the system level to go beyond 0dBfs for small periods of time, which can introduce temporary undesired feedback artefacts.

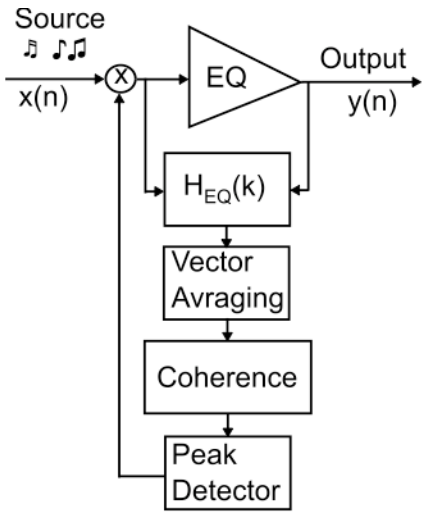

Figure 16 Real time transfer function normalization using source independent measurements.

4.5 Automatic gain normalization

The main idea of this normalization technique is to combine the strengths of a mathematical model normalization together with a transfer function measurement normalization technique. Therefore the system uses an unsolved sampled mathematical model, for example a *Z* domain mathematical model as a target measurement system. The measurement is performed by inputting an impulse to the mathematical model and obtaining its maximum through the realization of a measurement on its output. It is known from Fourier theory and linear system theory that $\delta(n) = FFT^{-1}(H(k))$, where $\delta(n)$ is the output impulse response of the system, $FFT^{-1}(n)$ is the inverse Fourier transform and $H(k)$ is the transfer function of the system under study. By applying the following identity, where $f(n)$ represents an arbitrary time domain function, $f(n) = FFT^{-1}(FFT(f(n)))$, and given that the input $x(n)=1$ *for* t_0 and $x(n)=0$ for any other time, then we can say that the output $y(n)$ is equal to $\delta(n)$. Therefore for such an input:

$$H(k) = FFT(y(n)), \qquad (11)$$

thus the transfer function of a complex system whose input is an impulse is given by performing the FFT of the output.

In other words the normalization constant can be found by applying an impulse to a mathematical model of a system, such as a *Z* domain function. Then a simple FFT is applied to the output. The resulting output can now be searched for the maximum value. In practice, only searching half the FFT data is necessary. The inverse of the obtained value is the normalization constant to be applied to the input.

The algorithm for implementing the automatic maximum gain normalization technique is presented in Figure 17. In a standard system, the user interface would be connected directly to the audio processing device. For demonstrating the algorithm, we have detached the user interface and stored the corresponding coefficients coming from the interface in a memory block called the fade-in parameters block. This memory block sends the coefficients to the audio processing device once the normalization constant has been found. The coefficients together with the normalization constant are transferred using a linear interpolation algorithm that ensures a soft, modulation-free transition to the next system state. An advantage of the user interface detachment is that the method can be implemented on analogue systems by interfacing the analogue user interface with analogue to digital converters and by transferring the results to the audio device using digital to analogue converters.

The algorithm sends an impulse to the mathematical model every time a change in the user interface has been detected. This ensures a correct normalization every time the linear system state has changed. Thus it is possible to calculate correctly the normalization value even if the transfer function order changes, for example when bypassing certain sections of an equalizer or even if the system design has changed, such as changing a filter in real time from a peak/notch to a shelf filter.

One of the advantages of this method is that it can be implemented either at clock speed or at sample rate speed. It also offers a more general solution to linear system normalization. The only section of the algorithm that needs to be revised if the linear system is changed is the memory sector containing the mathematical model. This gives the automatic maximum gain normalization technique the capability of being implemented as a solid-state chip, which can be interconnected to memory containing the model. A block diagram of such normalization algorithm is presented in Figure 17.

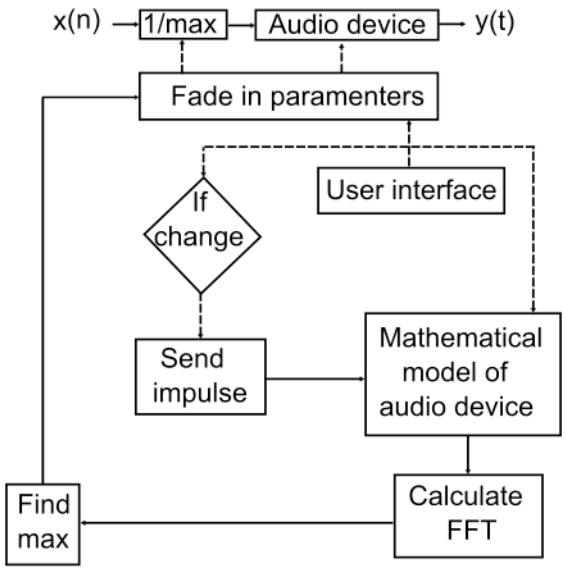

Figure 17 Algorithm of the proposed normalization technique using a truncated impulse response.

4.6 Research and implementation

This technique has been implemented on a full parametric equalizer, Figure 18. The implementation uses six biquadratic filters. One of them is a low pass filter, another is a high pass filter and four of them are full parametric filters. The low and high pass filters have user frequency selectivity and the last four have frequency gain and quality factor (Q) user parameters. Also, the two outer parametric filters can be swapped between a peak/notch filter or a shelving filter. Every time a filter is modified, the coefficients driving the transfer

function of the system change. Therefore a new normalization value is derived for every parameter change. The equalizer has the possibility of individually bypassing the high pass filter, the low pass filter, and the parametric filters. The compensated gain in dBfs is displayed at all times. A bypass button prevents the automatic maximum normalization technique for comparison purposes.

The mathematical model is given by equation 12. It is simply the unsolved *Z* domain transfer function of six biquadratic filtes in series, one per filter in the implemented equalizer, where the coefficients can be positive or negative. The FFT frame size used to implement the algorithm was of N=1024 samples.

$$H(z)=\frac{(a_1+b_1z^{-1}+c_1z^{-2})}{(1+d_1z^{-1}+e_1z^{-2})}\cdot\frac{(a_2+b_2z^{-1}+c_2z^{-2})}{(1+d_2z^{-1}+e_2z^{-2})}\cdot\frac{(a_3+b_3z^{-1}+c_3z^{-2})}{(1+d_3z^{-1}+e_3z^{-2})}\cdot\frac{(a_4+b_4z^{-1}+c_4z^{-2})}{(1+d_4z^{-1}+e_4z^{-2})}\cdot\frac{(a_5+b_5z^{-1}+c_5z^{-2})}{(1+d_5z^{-1}+e_5z^{-2})}\cdot\frac{(a_6+b_6z^{-1}+c_6z^{-2})}{(1+d_6z^{-1}+e_6z^{-2})} \quad (12)$$

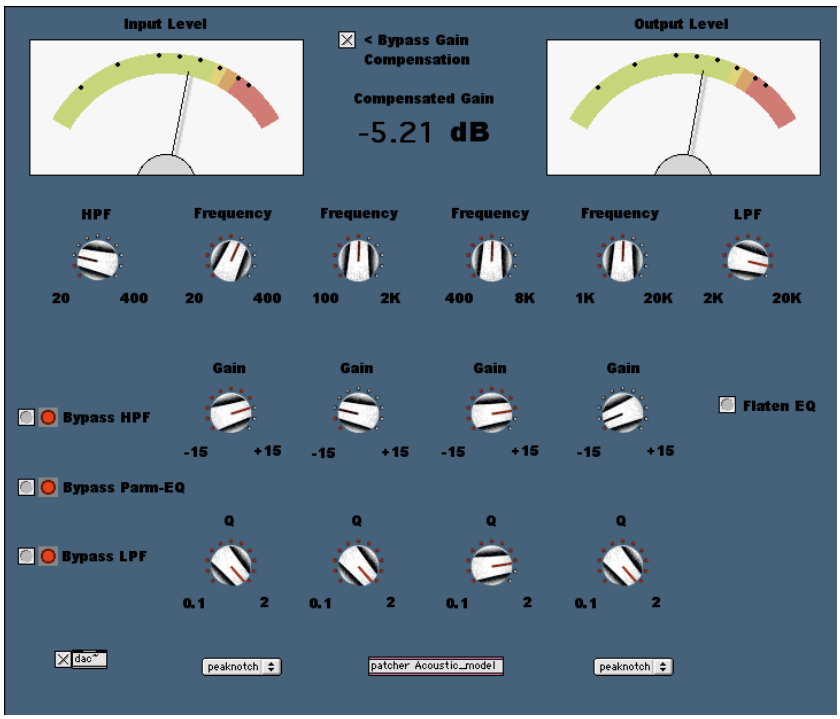

Figure 18 User interface of the implementation of the proposed normalization technique on a six biquadratic filter.

4.7 Test and results

Open loop source independent measurements were performed for the implementation of the method on a six biquadratic parametric filter implementation. Measurements of the resulting transfer function were made using a sample rate of 44100 with a fixed point per octave FFT with a frequency resolution of 24 points per octave with a Hanning window with 32 vector averages.

Several boost and cuts corresponding to the equalizer user settings presented in Figure 18 have been plotted on Figure 19. The dashed line represents the non-normalized response of the equalizer while the solid line represents the normalized transfer function. The solid line has been successfully normalized below the 0dB threshold line. This means that boost functionality on the equalizer is still available relative to the normalization value and does not contribute by adding gain to the overall transfer function of the system. The overall compensation applied to the equalizer for these settings was -5.21dB.

It was also found that for low frequencies the lower frequency resolution below 400Hz could be affected if the Q of the filter is high. This is because the frame size truncates the impulse response of the system under study, causing loss of low frequency information. The error plot of gain normalization vs. Q is presented in Figure 20. It can be seen that the higher the Q, the higher the error. It can also be seen that the error changes in an exponential manner with respect to Q. This means that the error in estimation of the maximum of the transfer function is only significant for very strong filtering of very low frequency content.

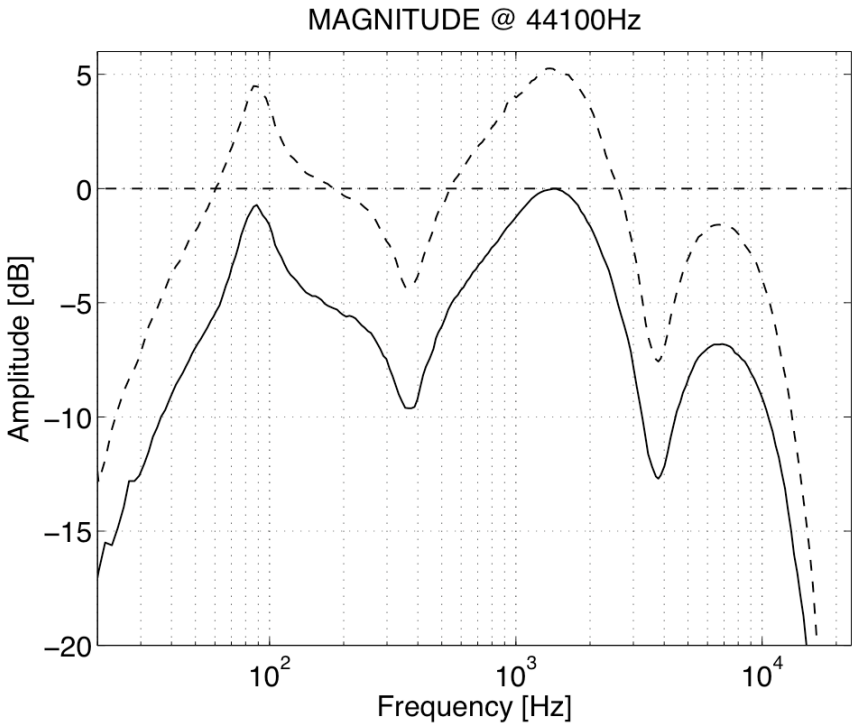

Figure 19 Transfer function of an un-normalized and a normalized response. The dash-dotted (-·-·-) line represents the threshold for maximum gain before feedback, the dashed line (- - -) represents the transfer function of a non-normalized acoustic system and the full line (——) is the transfer function after applying the normalization method.

This particular low frequency error can be counteracted by using an inverted multiplying mask which matches the error plots presented Figure 20. On the other hand, using a constant-Q transform (Brown 1992) might offer a more generalized solution. This remains a subject of future research.

Software simulation based on a single feedback path model like the one shown in Figure 14 was implemented. The model takes into account temperature to calculate the speed of sound and uses the inverse square law to determine the delay and amplitude of the feedback path contribution to the system. Under this condition the system behaved as expected, avoiding howling, for frequencies above 400Hz. After diminishing the overall electronic transfer function gain by 6dB the system performed as expected for all frequencies. This was attributed to the error associated with the use of high Qs in the low frequency range.

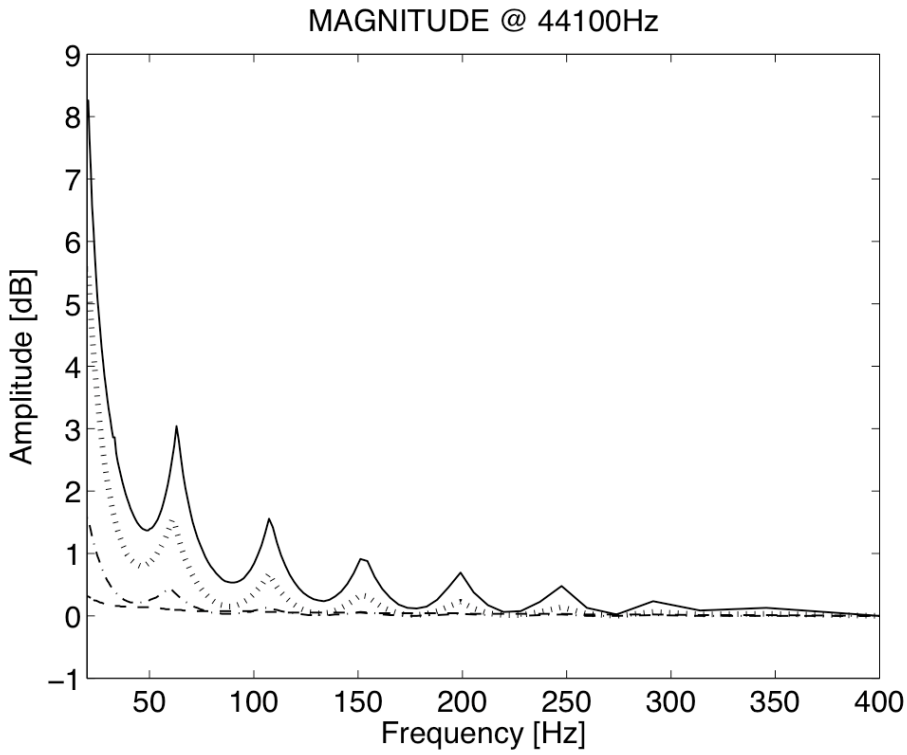

Figure 20 Error due to filter Q for a frequency range of 20Hz to 400Hz. The full line (——) is error for Q=2 (knob at full right position), dotted line (⋯) is error for Q=0.996, dash-dotted (-·-·-) line is error for Q=0.371 (knob at center position) and dashed line (- - -) is error for Q=0.1 (knob at full left position).

Laboratory tests on a real acoustic system were also performed. The experimental set-up and recording environment are shown in **Figure 21**. The room consisted of an acoustically isolated room-in-room construction with no parallel wall design for listening tests. The room has an approximate volume of 24m^3 and reverberation time of 0.2s according to a RT30 derived from a measurement of the impulse response of the room. A self-powered studio monitor playing wideband-recorded music was used as a source. The speaker was placed 10cm away from an omni-directional flat frequency response microphone. Care was taken to keep the source level set such that microphone diaphragm distortions are avoided. The microphone was then connected to a soundcard interfaced to the software containing the automatic normalization parametric equalizer implementation. The output of the system was connected to a line driver to control the overall amplification gain of the system. Finally a

self-power studio monitor was placed at 160cm from the microphone capsule. This speaker was used as the main sound reinforcement speaker. Care was also taken to avoid electronic and acoustic distortion over system.

Figure 21 Acoustic measurement setup.

While the equalizer remained flat, the system was driven to the marginal state of maximum gain before feedback. Afterwards, numerous boost and cuts were applied to the equalizer. Compensations of up to -50dBs were achieved without howling. It was observed that a 3dB margin was required for avoiding howlback due to artefacts introduced by high Qs on the low frequency range. This is better than expected by simulation. It is thought that this is due to the room acoustics, which caused a 3dB destructive contribution to the feedback effect compared to an ideal constructive 6dB contribution achieved during the single path simulation using software. In a room with high reverberation time the compensation is likely to be large, while for a less reverberant room such as an open space there is likely to be little need for compensation.

4.8 Summary

A normalization technique that prevents feedback by maintaining loop gain below unit has been introduced. The method performs real time normalization of the gain of a changing linear system to stop it from going beyond the maximum gain before feedback threshold. Although the linear implementation of the method suffers from low resolution in the low frequencies, simulations and acoustic tests implemented on a six biquadratic parametric filter implementation have shown its suitability for use in live sound reinforcement applications and music mixing. A series of automatic mixing tools that can beneficiate from the use of the presented normalization methodology will be presented in the following chapters.

Chapter 5
Automatic head-amplifier gain

5.1 Introduction

A method has been researched for the purpose of optimizing the head-amplifier input gain levels of a live audio mix. This method is capable of optimizing the analogue to digital conversion by automatically setting the optimal amount of analogue-input head-amplifier hardware gain. The system has applications in automatic mixing of live music, dynamic mixing of game audio, studio recording, music production and studio post-production.

5.2 Automatic gain

Given that our aim is to research automatic signal processing methods for live mixing we must ensure that the analogue to digital conversion is not only optimal and maximizes the use of the digital dynamic range while minimizing distortion, but we must also ensure that we get a normalized maximum signal reference level so that any automatic algorithm will have a known standard starting reference between channels.

The electronic components of a mixer have amplitude limits before distortion. The input signals may be said to have a maximum value of 1. Normalizing all inputs to have maximum peak value of 1 would optimize the dynamic range of the system while giving a common reference for all inputs. Unfortunately, in a live system we do not know the maximum level of the incoming signal. Therefore, adaptive gain compensation should be implemented.

5.3 Research and implementation

Consider a sound mix which is comprised of *M* channels with a valid range from 0 to *M-1*. Each individual channel input $y_m(t)$ has a scaling factor control vector $cv_m(t)$, also known as head-amplifier gain. We can say that the channel overall gain in the time domain is given by $cv_m(t)\ x_m(t)$, where $cv_m(t)$ and $x_m(t)$ can take a maximum value of 1 and this operation happens inside the VCA.

In order to avoid distortion on the inputs, $x_m(t)$ must be scaled continuously by an input gain factor $cv_m(t)$, where $cv_m(t)$ is the adaptive rescaling factor of $x_m(t)$ in order to avoid distortion. $cv_m(t)$ is equal to 1 at t_0 and $cv_m(t)= cv_m(t-1)-r$, where *r* is the amount of decrement applied every time $x_m(t)$ is greater than 1. This method was implemented on a recallable head-amplifier digital to analogue converter. The implementation is depicted in Figure 22.

The implementation in Figure 22 consists of a hybrid circuit in which the system is capable of inputting an analog signal while outputting a digital one. Based on such an implementation we can use the overflow flag of the ADC, $|y_m(n)|$, for determining if there is distortion on the ADC due to excessive head-amplifier gain. Since the head-amplifier is a Voltage Controlled Amplifier, VCA, we can control the amount of signal coming into the ADC by driving it with $cv_m(t)$ as its adaptive rescaling factor. Given that the ADC is reporting the overflow signal in the form of a digital bit, and a bit is equal to 6dBFs where the bit has two states each comprising a 3dBFs, then the optimal value for *r* is +3dBFs. This design will ensure correct digitization of the signal while continually adapting the gain in the case of signal distortion. Most importantly, this will give the system a normalized set of input signals $y_m(n)$, for which the reference limits are the same. The user interface of such a device is very simple and has been implemented in Figure 23.

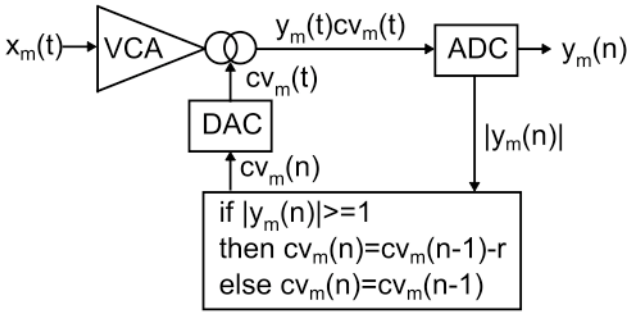

Figure 22 Adaptive Gain Signal Acquisition block diagram.

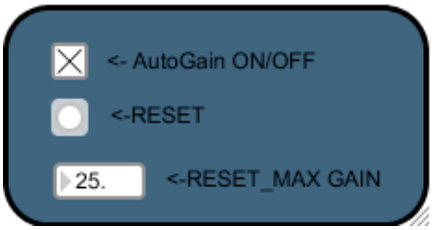

Figure 23 User interface of the implementation of the proposed automatic head-amplifier controller.

5.4 Test and results

A high gain musical test signal, $x_m(t)$, shown in Figure 24B was used as an input for a simulation based on the block diagram depicted in Figure 22. The amount of decrement, *r*, used was equal to a 0.5dB step. The self adjusting gain factor, $cv_m(n)$, was updated every time $|y_m(n)|$ had an amplitude higher than one. This is depicted in Figure 24A. The algorithm has a total convergence in the range of 8.75sec, but given its quasi-exponential rate convergence, the system is close to stability after only 2.73sec. The resulting $y_m(n)$, which is within the maximum range of +/- one, is depicted in Figure 24C.

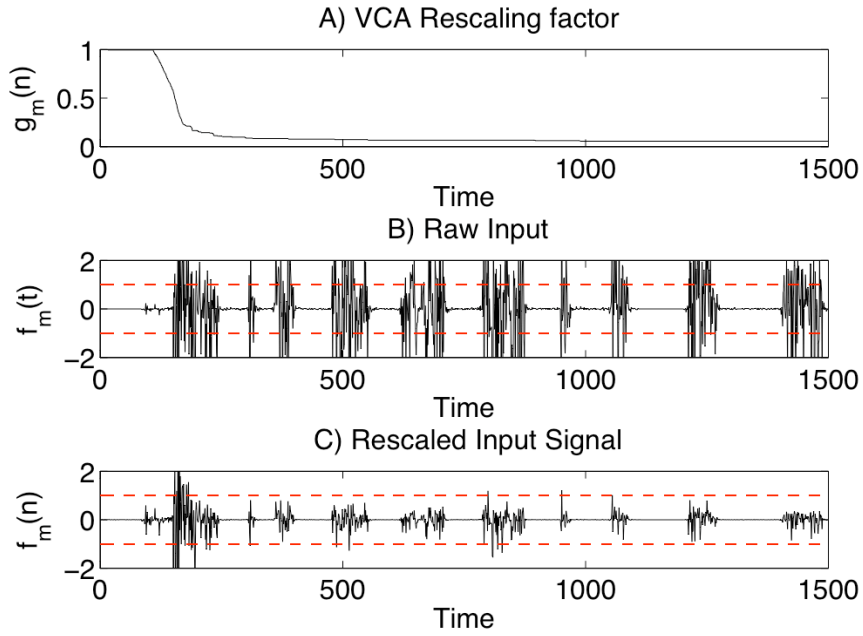

Figure 24 Simulation of automatic input gain normalization, [Time in units of 10ms].

It can be observed that the system is capable of automatically reducing head-amplifier gain when the peak power exceeds the electronic limits of the head-amplifier.

5.5 Summary

The automatic head-amplifier tool research has proven to be capable of setting up gain to an optimal level for real time signals. In principle such a system is simple to implement it requires the use of high quality VCA controlled head-amplifiers capable of working with wide dynamic range signals while maintaining a low noise floor. This step is paramount in the implementation of any automatic signal processing system; so all algorithms presented in this thesis will assume a digital signal that complies with a normal dynamic range and are free of undesired distortion. All automatic mixing tools presented next will assume their input has been previously normalized using the automatic head-amplifier normalization technique presented here.

Chapter 6

Automatic polarity and time offset correction

6.1 Introduction

A method for reducing comb-filtering effects due to delay time differences between audio signals in a sound mixer has been implemented. The method uses a multi-channel cross-adaptive effect topology to automatically determine the minimal delay and polarity contributions required to optimize the sound mix. The system uses real time, time domain transfer function measurements to determine and correct the individual channel offset for every signal involved in the audio mix. The method has applications in live and recorded audio mixing where recording a single sound source with more than one signal path is required, for example when recording a piano with multiple microphones. Results are reported which determine the effectiveness of the proposed method.

6.2 Automatic polarity and time offset correction

It is common in recording or live mixing to use more than one microphone or signal path to record or reproduce a single source (Shure Brothers Inc. 2007). Although using multiple microphones can in some cases improve the sound characteristics of the source, it can also introduce artefacts in the form of destructive interference. For this reason it is of paramount importance to

ensure all signal paths involved are synchronized while sharing compatible polarity. The reason for this is to avoid any undesired audible cancellation artefacts in the audio signals. Common examples of mixing practices that can introduce audible interference due to differences in time arrival and polarity errors are:

-Using more than one microphone to record a drum set or recording a piano with more than one microphone.
-Recording an electric guitar / bass using a direct box together with a microphone placed at the amplifier.
-Using a wireless signal, while simultaneously using a microphone to record the amplifier.
-Using a parallel digital sound effect or digital device next to an analogue or direct feed. This is a common practice in live sound when sending the digital effect return through a stereo channel.
-When using implementations of digital mixers or workstations that do not compensate for plug-in processing latency (SSL 2008).
-Use of more than one microphone on a podium or stage.

All previous examples are common audio practice procedures that have destructive interference in the form of comb-filtering. It is the aim of this chapter to present a method that corrects these artefacts therefore we will begin with a review of the relevant concepts underlying comb-filtering.

6.2.1 The comb-filter

It is well known from signal processing theory that the summation of two signals, which are highly correlated and have different time arrivals, when added together, results in a spectrum artefact known as comb-filtering. Comb-filtering is a time domain problem that affects the spectrum in a perceptible manner (Ballow et al. 2002). Figure 25 shows the comb-filtering effect of the

addition of white noise to audio signals with the same amplitude and a 1ms delay between each other.

The comb-filter minima and maxima points are directly related to the delay between signals. Given that d is the mutual delay time between signals, the first cancellation frequency notch, F_c, is located at

$$F_c = 1/2d \qquad (13)$$

and each successive minimum will be located at odd multiples of F_c , while each successive maximum will be located at even multiples of F_c.

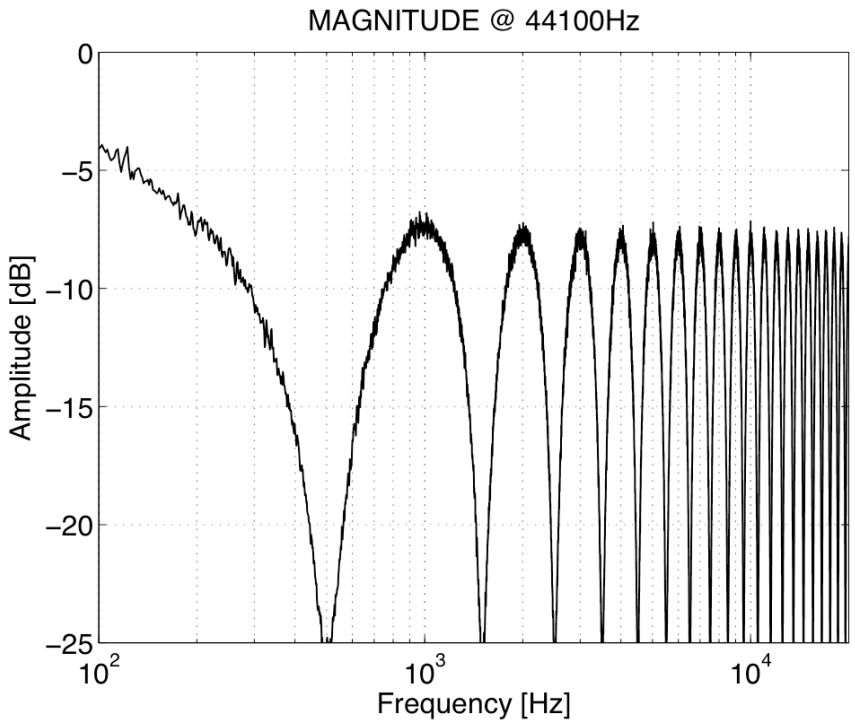

Figure 25 Comb-filtering of two white noise signals, both having the same amplitude, with a 1ms delay between them.

The existence of comb-filtering spectral artefacts in the audio signals is audible, and can make an audio engineer erroneously equalize the signal to improve its spectral texture. Unfortunately, due to its time domain nature,

comb-filtering is not equalizable and requires a time delay compensation to remove it. Finding the right amount of delay in a multi-channel mix that will minimize the comb-filtering between tracks is not an easy task. For this reason a method that automatically detects the relationship between channels by determining the impulse response has been devised and investigated. The proposed method has the aim of obtaining the minimal delay per channel required to minimize comb-filtering.

6.3 Research and implementation

The transfer function of a system is the Fourier transform of the impulse response of the system. The transfer function can be computed by dividing the Fourier transform of the output of the system by the Fourier transform of the input of the system. The impulse response can then be computed by the inverse Fourier transform. The impulse response of a system determines its dynamic characteristics. If we derive the impulse response of a reference signal with respect to another, given that they are correlated, we can determine the delay between them. The polarity of the maxima of the resultant impulse response can be used to determine the polarity relationship between the two signals with a common source.

In this implementation, $x_{\mu}(n)$ is denoted as the reference measurement and $x_m(n)$ as the measured signal. μ can take any value, from $0, \ldots, M-1$ given that m has a valid range from $0, \ldots, M-1$. At the beginning of the process μ must take an arbitrary initial value. The inputs and reference signals must be weighted by $w(n)$, a Hanning window, in order to reduce *FFT* artefacts. Their *FFT*s are expressed by $X_m(k) = FFT[w_{HN}(n) \cdot xg_m(n)]$ and $X_{\mu}(k) = FFT[w_{HN}(n) \cdot xg_{\mu}(n)]$, and therefore we can approximate the transfer function of the reference channel against an input signal, $Ha_m(k)$, by equation 14

$$Ha_m(k) = \frac{X_m(k)}{X_{\mu}(k)} \qquad (14)$$

The previous calculation has the advantage of making the system able to perform the calculation of the transfer function independent of the source content. Unfortunately reverberation and noise will adversely affect the result of the calculation and several procedures for improving the robustness of the method will be explained next. In order to obtain an unbiased estimate of the transfer function when the measurement channel has been contaminated with uncorrelated noise, we must divide the auto-spectrum of the measured channel against the cross-spectrum of the reference channel,

$$Ha_m(k) = \frac{X_m(k)X_m(k)}{X_m(k)X_\mu(k)} = \frac{X_{mm}(k)}{X_{m\mu}(k)}. \qquad (15)$$

Therefore when the measurement is contaminated by noise, the transfer function may be improved given that the noise is averaged out when performing the cross spectrum (Meyer 1992).

The transfer function measurement can also be made more resilient to noise by performing complex averaging. This is achieved by averaging its complex components frames, such that random noise being added to the complex vector is averaged out. The vector averaging is described by:

$$Hv_m(k) = \sum_{i=1}^{S} \frac{(HR_m(k))_i}{S} + j\sum_{i=1}^{S} \frac{(HI_m(k))_i}{S}, \qquad (16)$$

where *S* is a constant, greater than cero, and represents a number of iterations over which the frequency vectors are to be averaged. The larger the value of *S* the longer the system will take to compute. A typical value for *S* under normal conditions is 4 and under very noisy conditions can go up to 256 averages. The total time it takes to compute the full average is a function of the frame size. For example for a 1024 frame at a sample rate of 4410Hz a vector average for *S*=4 will take 92.88ms to compute.

Once we obtain $Hv_m(k)$ we could proceed to apply an Inverse Fourier Transform (IFFT) in order to obtain the impulse response. Unfortunately the system might be contaminated by reverberation. Reverberation can be treated as noise, which is correlated to some degree to the measurement channel, and can still have some undesired effects over the transfer function measurement. Therefore we borrow a technique commonly used for speech correlation that is known as the Phase Transform or PHAT (Knapp and Carter 1976). This is a weighting procedure in which equal emphasis is placed on each frequency. In other words, all frequency components are neglected and forced to have a unity value, while taking into account only the phase information of the transfer function. This type of weighting tends to be sub-optimal under ideal conditions, but tends to be less susceptible to anomalous conditions, particularly to reverberation (Brandstein and Silverman 1997). The resulting equation for obtaining the phase dependent impulse response, $\delta_{PHATm}(n)$, is given by

$$\delta_{PHATm}(n) = \mathrm{IFFT}[Hvm(k) \cdot |Hvm(k)|^{-1}], \qquad (17)$$

In order to determine correctly the location of the signed magnitude of the impulse response, it is necessary to obtain its amplitude and position in time. This is done by a peak finder, which searches for the largest absolute value inside a buffer of 1024 samples. Then the algorithm proceeds to store the corresponding signed magnitude for that value and the position where it was found, which is used to determine the delay time between the reference and the measured signal. Due to the fact that the impulse has been truncated by using finite length FFTs, the impulse obtained is a noisy signal in itself, and it is necessary to accumulate the signed amplitude using the following equation:

$$\delta_{Bm}(n) = \sum_{i=1}^{B} \frac{(\delta_{PHATm}(n))_i}{B} \qquad (18)$$

Where B goes from 0 at t_0 up to infinity at t_∞ and $\delta_{Bm}(n)$ corresponds to the accumulated signed magnitude of the impulse. Given that i is the frame index.

A similar accumulative approach was initially used to determine the delay position but unfortunately these proved slow. This was because for low amplitudes it was impossible to determine the impulse position accurately without having long averaging times in the order of 100 averages. This was due to the fact that the peak finder will continuously accumulate noise peaks that were confused with the impulse response peaks. In other words, the smaller the amplitude of the impulse the smaller the signal to noise ratio and therefore the more corrupted data gets stored into the time delay position accumulator. For this reason an adaptive-accumulative method for determining the delay position was devised.

The delay time calculation is adaptive because the amount of accumulations needed in order to output a valid number is adaptively increasing or decreasing in inverse proportion to the absolute magnitude of the impulse response. In other words if the signal to noise ratio is large, a small amount of accumulations are needed and if the signal to noise ratio is small, more accumulations are needed before an accurate valid time delay, which truly reflects the delay misalignment between signals, is output. Once valid data has been output then it can be sent into an accumulator similar to the one presented in equation 19. This adaptive accumulation is shown next:

$$\delta_m(n) = \sum_{i=1}^{B_m} \frac{(\delta_{PHATm}(n))_i}{B_m}, \qquad (19)$$

where $\delta_m(n)$, is the resulting impulse response of the adaptive accumulation with respect to its amplitude and B_m is a function of the amplitude of the absolute maxima of the impulse response,

$$B_m = \text{int}\left(\frac{\alpha}{\left|\max(abs(\delta_{PHATm}(n)))\right|} \right). \qquad (20)$$

Where for the purpose of this thesis α has been chosen to be 2 in order to duplicate the number of minimum operations to validate the calculated delay time. The formula implies that a high amplitude in the delay estimation pick is the result of having a high SNR, and a low level in the delay estimation pick results in a low SNR. So for a delay estimation pick equal to one, with a=2 a convergence time equal to the length of 2 FFT frames is expected. Similarly, for a delay estimation pick equal to 0.25, a convergence time equal to the length of 8 FFT frames is expected. B_m was chosen to be reset every time $\delta_{PHATm}(n)$ changed in magnitude by a factor of $+/-10^{-2}$. This approach permitted the system to work at levels where the impulse response was practically buried in the noise while still being able to correctly determine its position; it also converged faster than pure accumulation. Figure 26, top, depicts the determination of the delay time with no accumulation. The lower plot in Figure 26 shows the comparison between pure accumulation and adaptive accumulation. Notice that variability is reduced and the adaptive accumulation tends to avoid to accumulated error in a faster manner. It also manages to converge faster than the pure accumulation approach.

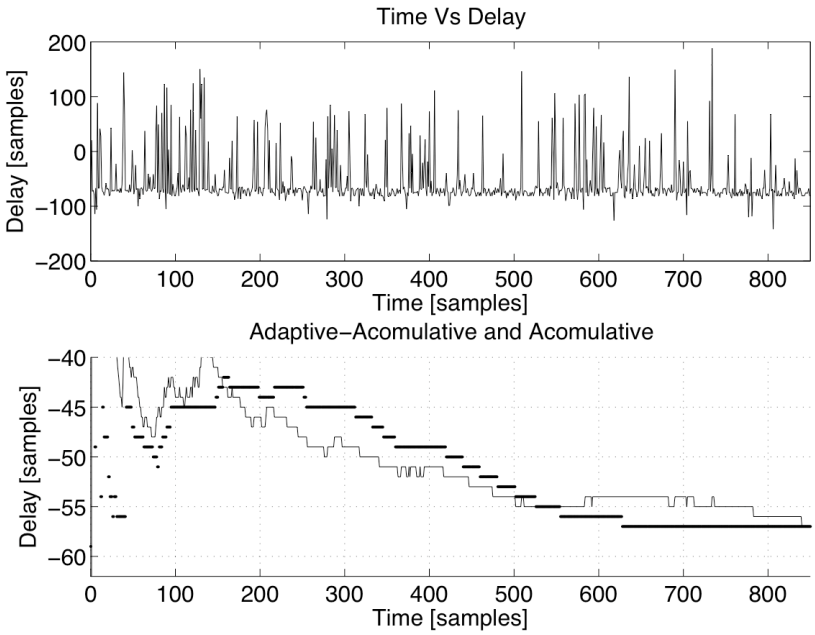

Figure 26 Non-accumulated delay times, (top). Comparison of accumulated delay time in gray, vs. accumulative adaptive delay time in black, (Bottom).

Once we have a stable valid impulse response we can determine its delay with respect to the reference signal by calculating the maximum argument of the impulse response function. This is given by the following equation 21

$$\tau_{\mu m}(n) = \arg\max_{n}(abs(\delta_m(n))) \qquad (21)$$

By evaluating the impulse response by $\tau_{\mu m}(n)$ and extracting the sign we can derive the polarity of the measured signal with respect to the measurement signal, given by equation 22,

$$\rho_{\mu m}(n) = \mathrm{sgn}(\delta_m(\tau_{\mu m}(n))) \qquad (22)$$

$\boldsymbol{fv}_m(n)$ is the output vector containing the delay and polarity information for every $x_m(n)$ with respect to the reference channel $x_\mu(n)$. The feature extraction method for finding the delay and polarity between the input signals $x_m(n)$ and the references signal, $x_\mu(n)$, is depicted in Figure 27.

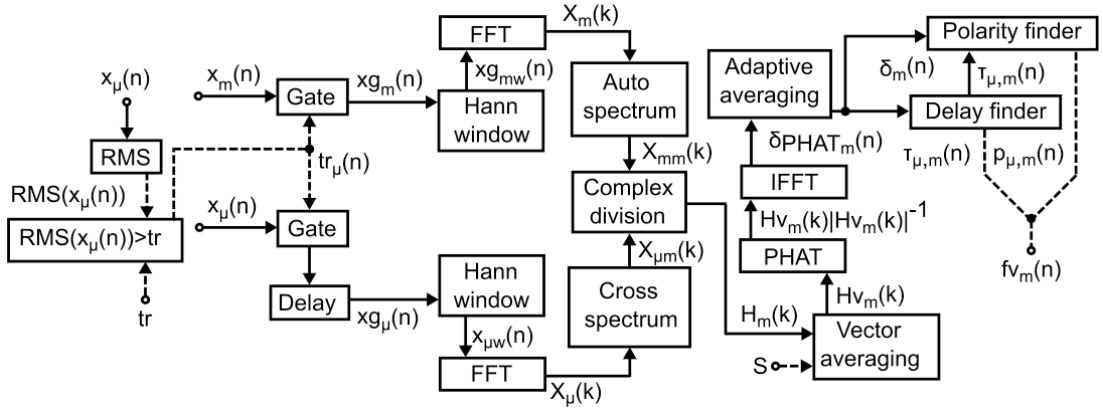

Figure 27 Feature extraction of a time polarity offset corrector.

In Figure 27 it can be seen that an extra delay block is applied to the reference signal. This delay allows the feature extraction to see negative delays. This is useful since the initial reference signal may be selected arbitrarily, and some of the measured signals may contain a negative delay. At the end of the process the reference will be optimized, therefore, all delay reading will be positive, but as an initial condition negative delays can exist. The applied delay is *N/4* samples long, where the FFT resolution is equal to *N*. The current researched implementation uses 1024 point FFTs with a Hanning window with no overlap. The system currently runs at a 44.1K sample rate. This means one sample is equivalent to 0.023ms.

6.3.1 Cross-adaptive processing

The method makes use of a cross-adaptive processing topology in order to measures the features, delay and polarity, and established the interaction between channels with respect to a user specified reference signal. Thus the cross-adaptive feature processing can establish the optimal solution to minimize the amount of delay added to synchronize all channels involved.

During the cross-adaptive processing a minimization solution is obtained from the impulse response relationships of the channels involved with respect to the other channels. This gives the optimal delay time to reduce the comb-filtering between the channels to be mixed. The algorithm calculates the impulse response for every channel with respect to the reference channel. The cross-adaptive algorithm scans the delay times for every channel and finds out if there are any negative delay values. If there are no negative delay time magnitudes the algorithm sends the delay compensation values to all individual channel-processing units, Figure 28.

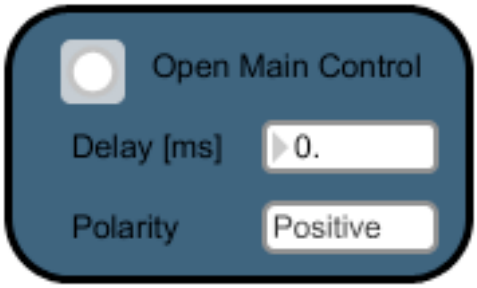

Figure 28 Individual channel processing unit user interface. The implemented automatic mixing tool drives the processing unit control parameters.

In the case where negative delay times exist, the algorithm scans for the most negative delay value and finds the channel responsible for it. Once the channel responsible for the most negative delay has been found the algorithm sets it as the new reference channel and the whole process starts again, until all delays are positive. In this manner the algorithm is capable of offering an optimal delay solution for all inter-channel delay dependencies. In the case of polarity issues, the cross-adaptive effect uses the signed magnitude of the amplitude of the reference impulse response in order to match the polarity of it to all other dependent channels. This means that if a channel has an inverted polarity with respect to the reference the algorithm will reverse its polarity in order to obtain a constructive interaction between all channels.

The signal processing part of this algorithm consists of individual delay and polarity inverter units inserted on each channel. A control vector controls each of these units. The control vector is derived inside the cross-adaptive feature device, in this case, a delay polarity optimizer. The control vectors are derived by processing the cross relation between feature vectors. The control vectors are obtained from the interrelationship between the user selected reference channel and the other channels. Therefore the system aims to determine the optimal polarity and delay times to avoid comb-filtering between channels.

If $\boldsymbol{fv}_{\tau\mu} \neq max(\boldsymbol{fv}_{\tau\,(0\ldots M-1)}(n))$ we must start by reassigning $x_\mu(n)$ such that the delay added to all $x_m(n)$ is minimum. We then reset the feature extraction process and start a recalculation of the feature vector $\boldsymbol{fv}_m(n)$ given the new assignation of $x_\mu\,(n)$. The components of $\boldsymbol{cv}_m(n)$ are given by

$$\mathbf{cv}_{\tau m}(n) = \mathbf{fv}_{\mu}(n) - \mathbf{fv}_{m}(n) \qquad (23)$$

and

$$\mathbf{cv}_{pm}(n) = \begin{cases} 1 & \mathbf{fv}_{p\mu}(n) = \mathbf{fv}_{pm}(n) \\ -1 & \mathbf{fv}_{p\mu}(n) \neq \mathbf{fv}_{pm}(n) \end{cases}, \qquad (24)$$

where $\boldsymbol{cv}_{\tau m}(n)$ corresponds to the delay control data value and $\boldsymbol{cv}_{pm}(n)$ corresponds to the polarity control data value per signal. Such cross-adaptive processing implementation has been depicted in Figure 29.

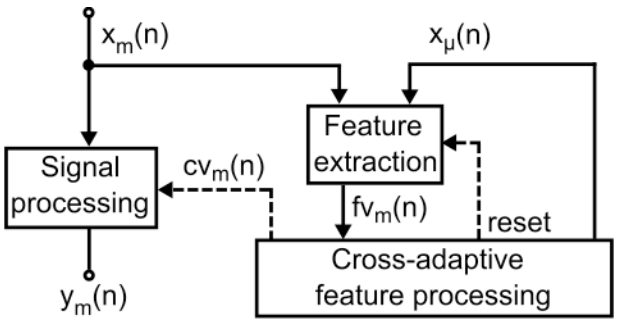

Figure 29 General algorithm flow diagram for an automatic mix cross-adaptive time offset corrector.

The Interface implementation of the cross-adaptive automatic mix time offset corrector is presented in Figure 30. The user has the ability to select the reference channel and the channels involved in the cross-adaptive procedure. The user has the ability to bypass individual or overall corrections. A manual accumulator reset is also available. The top window shows the impulse response of the chosen reference channel against the chosen measured channel before correction. The lower window shows the impulse response of the chosen reference channel against the chosen measured channel after correction. The chosen measured channel field is a form of visual aid. The rest of the measurement channels involved in the process are synchronized simultaneously.

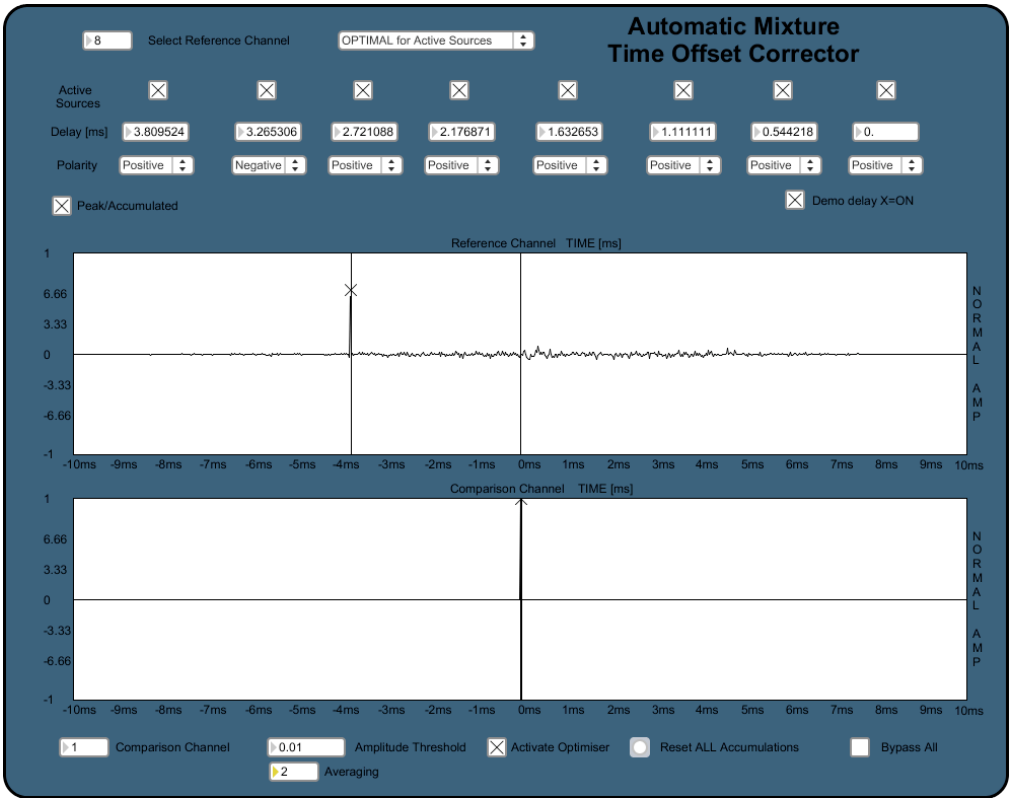

Figure 30 Master user interface of the implemented cross-adaptive time offset corrector.

6.4 Test and results

In order to evaluate the robustness of the algorithm against noise and reverberation the following experiment was performed. Given a reference signal and a measurement signal with the same amplitude and content and synchronized at t_0, thus having ideal impulse amplitude of one, we proceeded to add pink noise to the measurement channel, Figure 31 top. The noise was added in increments of 0.5 dB. Although the amplitude of the impulse response decreased when the noise was added, the system was able to keep track of the signal delay time at t_0 without a single sample error for pink noise up to a value smaller than 6dB. It was also found that for additive pink noise below -40 dB the effect on the measurements is negligible. Adding noise of amplitude 6dB greater than the signal proved impossible to track as the impulse completely disappeared in the background noise.

Next we proceeded to perform the same test but this time by adding reverberation. The reverberator used is one of the most common implementations of the Schroeder and Moore reverberatior model called freeverb~, implemented by Olaf Matthes (Matthes 2003). The settings for it were the default settings, which are:

Bypass: OFF
Room Size: 0.84
Damping: 50
Width: 100
Wet level: 0dB

The only parameter varied was the wet level. In the case of freeverb~ a wet level of 0dB means no reverberation has been mixed to the signal while a negative value represent a relative ratio of reverberation has been added. This means that a certain amount of relative reverberation has been mixed to the signal with respect to the relative level of the pure signal, Figure 31 bottom. It was found that for added reverberation of up to -26dBs it was possible to track the impulse at t_0 with a +/-1 sample error and for added reverberation of -30 dB it was possible to track the impulse at t_0 with +/-2 sample accuracy.

It was also noticed early during the development of the algorithm, that a "windowing effect" occurred on the impulse response amplitude. This effect consisted in a reduction of the amplitude of the impulse, as the reference signal and the measured signal were pulled apart in time. Given that the two signals are exactly the same the algorithm should show a single impulse with unity amplitude at t_0, and this unit amplitude should be maintained even when the delay between the reference channel and the measured channel changes. Unfortunately this was not true and the rate of change of amplitude against the delay between the reference and the measured channels is depicted in Figure 32 top. The implication of this was that the correct calculation of the delay

would be adversely affected as the delay time between the reference and the measurement channel changed. This is due to the fact that the impulse would be buried in the background noise causing the peak finder to erroneously take some noise peaks into account, Figure 32 bottom. This is the main reason why the adaptive accumulative peak averaging method used for deriving the delay times performs better than standard accumulation. It was found that the system was able to maintain +/-2 sample accuracy for delay times up to 5.31ms with +/-4 sample rate accuracy up to a delay time of 6.4ms.

Once the system was characterized with the above experiments we proceeded to test it with music. It was found that with pitched music where the reference signals and the measured signals are highly correlated the system tends to perform as expected within a +/-2 sample accuracy. The top plot of Figure 33 shows a piano signal that has been delayed with respect to the measurement channel by 4.76ms. Such delay displacement between the reference and the measured channel is extremely noticeable both in level and in spectral texture. Figure 33 bottom shows the impulse of the piano signals once both channels have been corrected by the cross-adaptive system. All highly correlated pitch signals such as this example performed in a similar manner. Polarity measurements were successfully corrected in all tests performed.

A second trial was performed with more difficult musical signals. An electric guitar was recorded directly with an analogue box and simultaneously recorded from the guitar amplifier, while containing a moderate amount of distortion. The performance of the system is shown in the top plot of Figure 34. A 0.54ms delay error was found. But when the system added that amount of delay to correct it was unable to achieve full correction. After correction the system still showed a 6 sample error, equivalent to a 0.14ms error, as depicted in the bottom plot of Figure 34. All polarity corrections were correctly identified in that test.

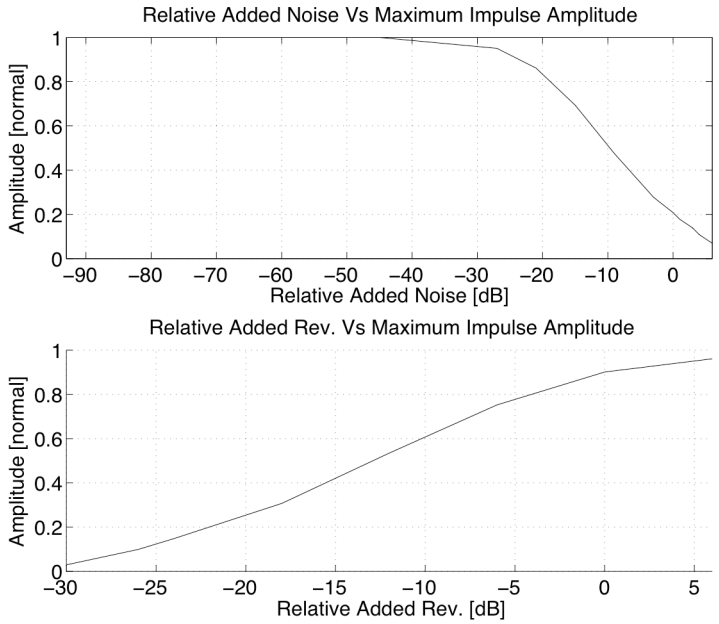

Figure 31 Impulse Response amplitude change due to the addition of noise (top). Impulse Response amplitude change due to the addition of reverberation (bottom). Measurements were performed for an impulse with no delay between reference and measured signal for a 0 sample error. The reverberation and noise were added to the measurement channel only.

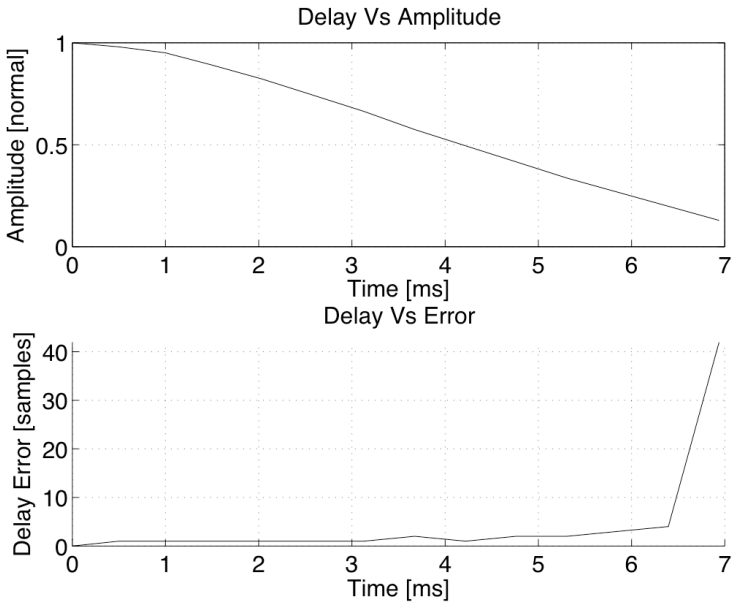

Figure 32 Impulse response amplitude windowing effect as a function of the delay offset between the reference channel and the measured channel (top). Delay calculation error as a function of the delay offset between the reference channel and the measured channel (bottom).

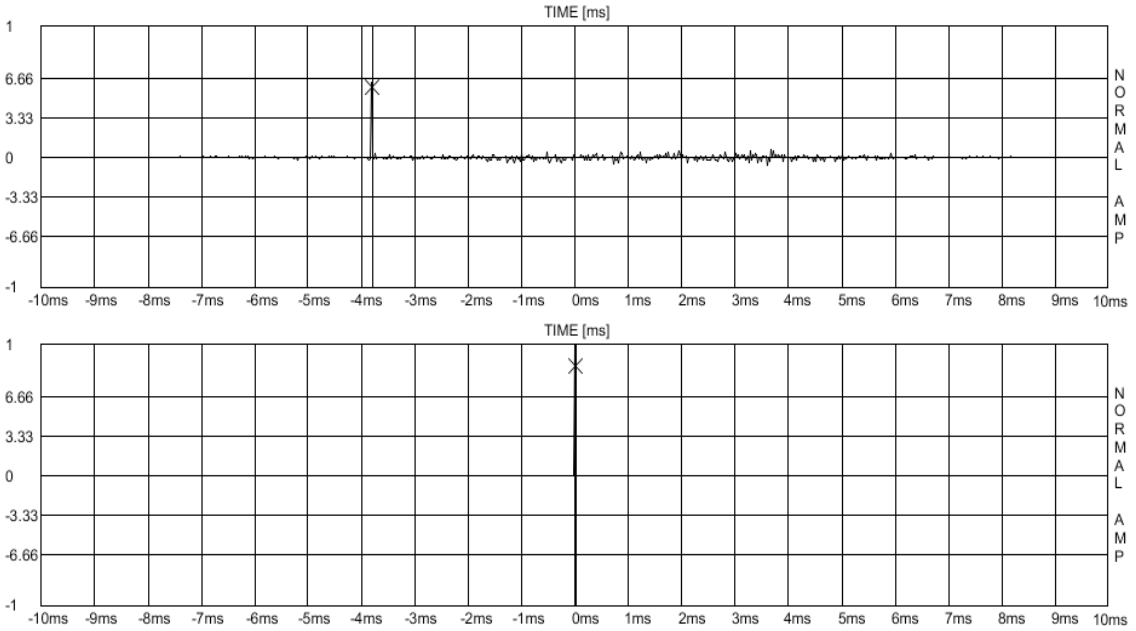

Figure 33 Measurements of impulse response of signal before correction (top) and after the correction (bottom). Measurements were made for a highly correlated signal.

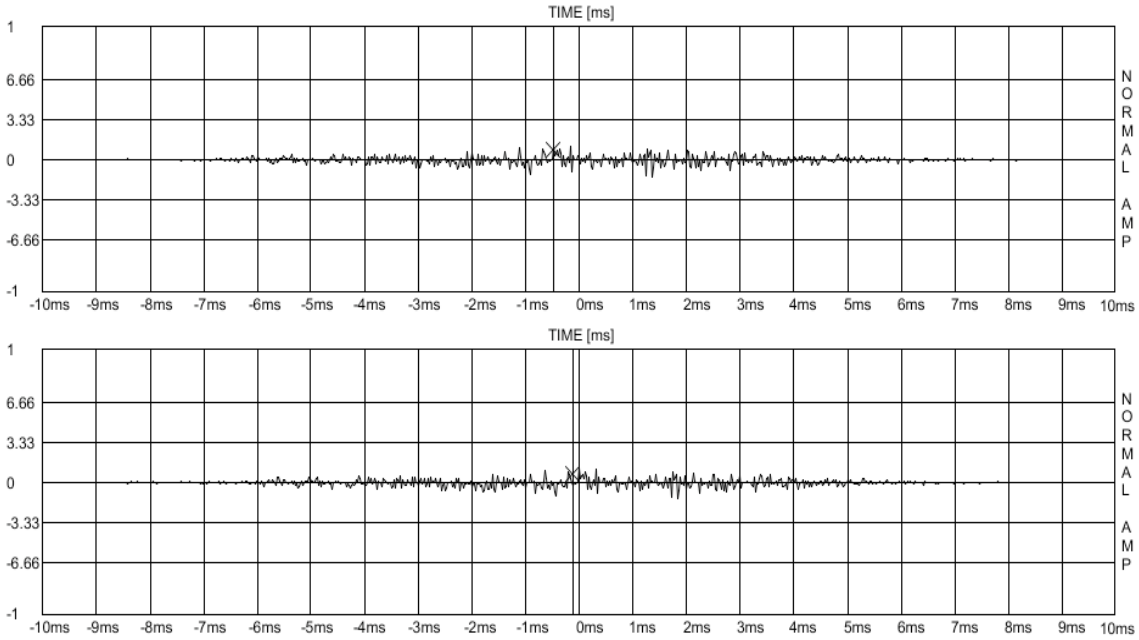

Figure 34 Measurements of impulse response of signal before correction (top) and after the correction (bottom). Measurements were made for a low correlated signal.

The system performed poorly for non-pitched percussive sound such as drums and was unable to find a delay value. On the other hand, it managed to obtain the correct polarity for signals with inverted polarity. This was investigated mainly by using a snare, with one microphone placed on top and one on the bottom of it, thus having one microphone that would require polarity inversion.

6.5 Summary

A method for reducing comb-filtering effects due to delay time differences between audio signals in a sound mixer has been proposed and implemented. The results show that the algorithm is capable of correcting delay errors of +/- 6.4ms with a +/-4 sample accuracy while optimizing the amount of delay to be used in the correction. The algorithm is also capable of optimizing the polarity settings for all channels involved in the cross-adaptive procedure. The system is functional within a +/-1 sample accuracy when the noise applied to one of the channels involved is less than 6dB. The system was capable of maintaining the same accuracy for a reverberation mix of up to -26dB. The research has concluded that the algorithm is suitable for real time live multi-channel mixing and studio applications. The system performs better for highly correlated pitched signals than for impulsive percussive ones. The current system can autonomously correct common mixing problems due to polarity problems while achieving optimal time delay synchronization between channels.

Chapter 7
Automatic spectral enhancer

7.1 Introduction

Spectral masking is a sound artefact that results from the total or partial loss of spectral content perception of one or more channels when they are mixed together. When sources are combined, the content of one source at a given frequency may be low with respect to the other sources in the mix. Thus the listener may not be able to associate that portion of content with its source. Although this obstruction or masking of spectral content has been used as a means of increasing compression ratios of sound files (Painter and Spanias 2000), when creating a sound mix, it is in most cases an undesired artefact because it hides some of the source content, and may leave some musical instruments unheard.

7.2 Automatic spectral enhancer

While performing audio mixing, one of the reasons for setting different relative levels and different equalization curves is to enhance some of the sources of the mix by reducing the spectral masking. This is a complex task and it requires an understanding of the relationship between the spectral content of the sources and the relative levels among channels.

As defined for the purpose of this thesis spectral masking for a given source can be measured by obtaining the amount of the level of spectral overlap between the source and the rest of the mix. In order to quantify masking the author has defined a masking equation based on the following concept: given a set of channel inputs, $x_m(n)$, and a given channel of interest, $x_\mu(n)$, we can define the spectral masking $\Delta S_{\mu,i}(k)$ of the channel with respect to the rest of the mix, as denoted by equation 25

$$\Delta S_{\mu,i}(k) = \left|X_\mu(k)\right|_i - \left|Y_{mix-\mu}(k)\right|_i, \qquad (25)$$

where $X_{\mu,i}(k)$=FFT[$x_\mu(n)$] and $Y_{mix-\mu}(k)$=FFT[$y_{mix}(n)$-$x_\mu(n)$], given i is the frame index and $y_{mix}(n)$ is the overall mix after applying the automatic enhancer. A spectral masking, $\Delta S_{\mu,i}(k)>0$, means the channel is unmasked and $\Delta S_{\mu,i}(k)\leq 0$ means the channel is masked by the rest of the mix. The spectral masking is an amplitude difference measurement; therefore, it is required to compensate for any windowing amplitude artefacts introduced by the FFT, as it might affect the measurements. In order to avoids such artefacts a 50% overlap Hanning window was used.

The accumulated spectral masking index of a source, $Sa_{\mu,i}(k)$, with respect to the rest of the mix can be obtained by accumulating equation 25 over a set of frames, and is given by equation 26

$$Sa_{\mu,I}(k) = \frac{\sum_{i=1}^{I}\left|\sum_{k=0}^{N-1}\Delta S_{\mu,i}(k)\right|}{N \cdot I}, \qquad (26)$$

where there are $N-1$ bins given N is the frame size of the FFT, and I is the total amount of frames to be accumulated, given i is the frame index.

With this in mind in this chapter we will propose a real time cross-adaptive channel enhancer that realizes a selective minimization of spectral masking for control of inter-channel dependency effects. The goal of this effect is to enhance a user selected channel by ensuring it is spectrally unmasked from the rest of the mix. The method uses full range magnitude adjustments to unmask the source instead of equalization techniques. This facilitates the mixing process, both providing support to professional mixing engineers, and providing a method by which musicians and performers without mixing expertise may still create mixes with minimal masking.

7.3 Research and implementation

The cross-adaptive channel enhancement in this chapter allows the user to enhance a selected channel by unmasking it from the rest of the channels. The simple approach to this would be to lower the amplitude levels of all other channels with respect to the channel to enhance. This approach is inefficient, as it affects all channels, even when the channels are not spectrally related to the channel the user wishes to enhance. Also performing such an action could introduce acoustic feedback or distortion if the gain needed is too large. A preferred approach, and the one that has been implemented, is to lower the levels of the other channels in proportion to their spectral relationship to the user-selected channel. In other words, if we aim to enhance a piccolo flute there should be no need to decrease a bass guitar because shares little or no spectral content with the piccolo. This type of complicated frequency dependent enhancement is familiar to audio engineers and it is what we aim to reproduce.

7.3.1 Inter-channel spectral decomposition classification

The first step in the proposed method is the classification of the incoming sources into spectral classes. This process is performed outside of the audible signal-processing path. The implementation is depicted in Figure 35, and is

based on an accumulative spectral decomposition classification method presented in (Perez_Gonzalez and Reiss 2007) and detailed in more depth in chapter 8.

Given that the input signals $x_m(n)$ may contain noise an adaptive gaiting stage must be implemented using an external signal $x_e(n)$ in form of a external microphone. Once the input $x_m(n)$ has been adaptively gated we can obtain a cleaner signal which we will refer as $xg_m(n)$.

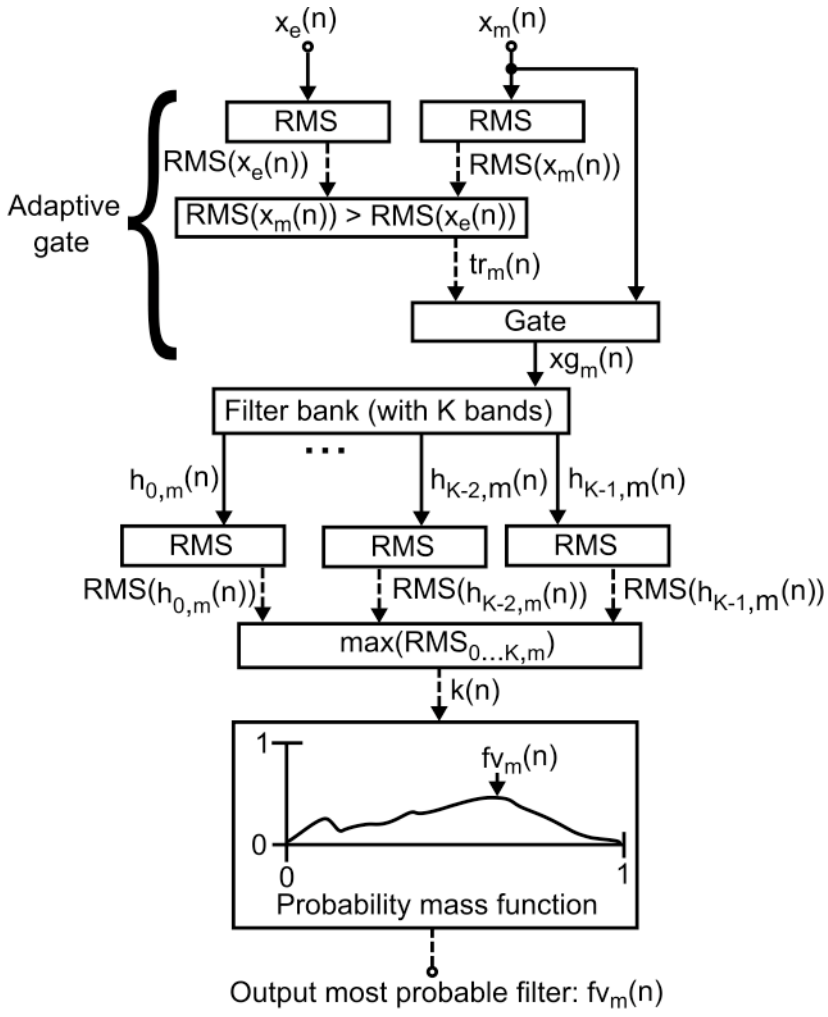

Figure 35 Block diagram of the spectral decomposition channel categorization algorithm.

First we decompose the input $x_m(n)$ after it has been processed by the adaptive gate, $xg_m(n)$ and process it using a filter bank. The filter bank has h_K band-pass filters, in which k has a valid range from 0 to $K-1$ and is equal to the total number of channels m being processed by the algorithm, where m has a valid range from 0 to $M-1$. This means that although the filter bank is working on an individual channel basis it expands or contracts dynamically, in proportion to the total number of source channels involved in the cross-adaptive analysis. Therefore each filter contained within the filter bank can have a corresponding classification feature vector value, $fv_m(n)$, which goes from 0 to $K-1$ where $K-1=M-1$. The filter bank is applied to each input channel and a score related to the maximum peak excitation filter is accumulated and updated every 1ms. The choice of accumulation window of 1ms was a compromise between feature accuracy and available signal processing power. A smaller window could yield better results but would result in an algorithm that can be too intensive to compute. The resulting k filter gets updated as the input signal changes while the accumulation converges into a stable $fv_m(n)$ value. The algorithm has an average conversion time for a musical signal of approximately 3s. The accumulative spectral decomposition algorithm categorises every single channel into a $fv_m(n)$ class, where the higher the value of $fv_m(n)$ the higher the frequency of the $fv_m(n)$ class. Therefore the accumulative spectral decomposition classifier is dependent on the signal content of a channel and outputs a spectral feature corresponding to a filter contained within the filter bank.

In order to test the proposed method an 8ch decomposition classifier was implemented. Measurements of the individual filters comprising the filter bank implemented for an 8 channel mix are presented on Figure 36. The filter bank is comprised of a set of second order low-pass IIR Biquadratic filters with cut off frequencies determined experimentally as follows: 35Hz, 80Hz, 187.5Hz, 375Hz, 750Hz, 1.5kHz, 3kHz and 6kHz. The combined response of the filter bank is presented on attenuating the gain of the higher frequencies. This approach was taken in order to reduce noise associated with high frequencies.

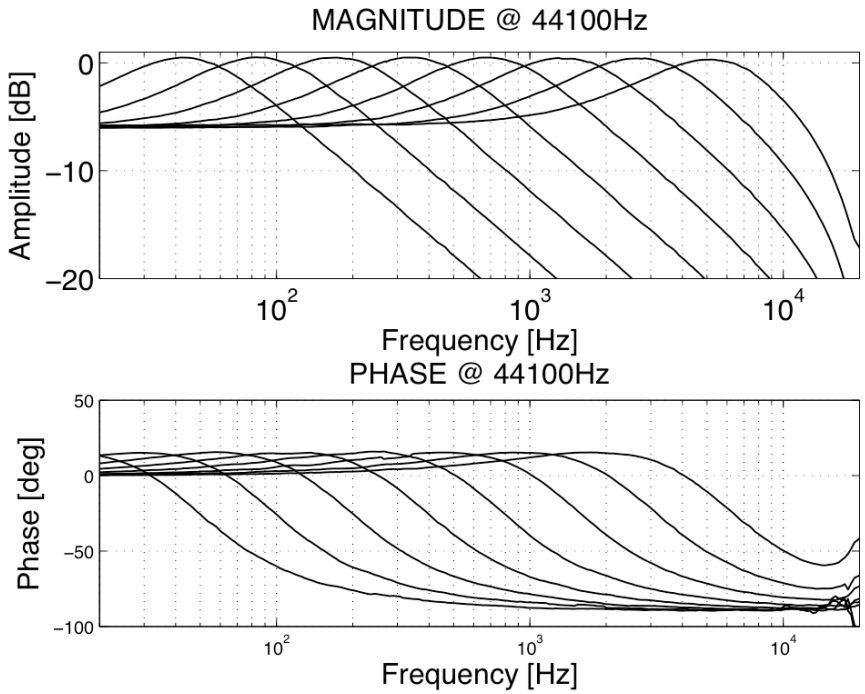

Figure 36 Magnitude vs. frequency and phase vs. frequency of eight individual filters composing the decomposition filter for a source channel.

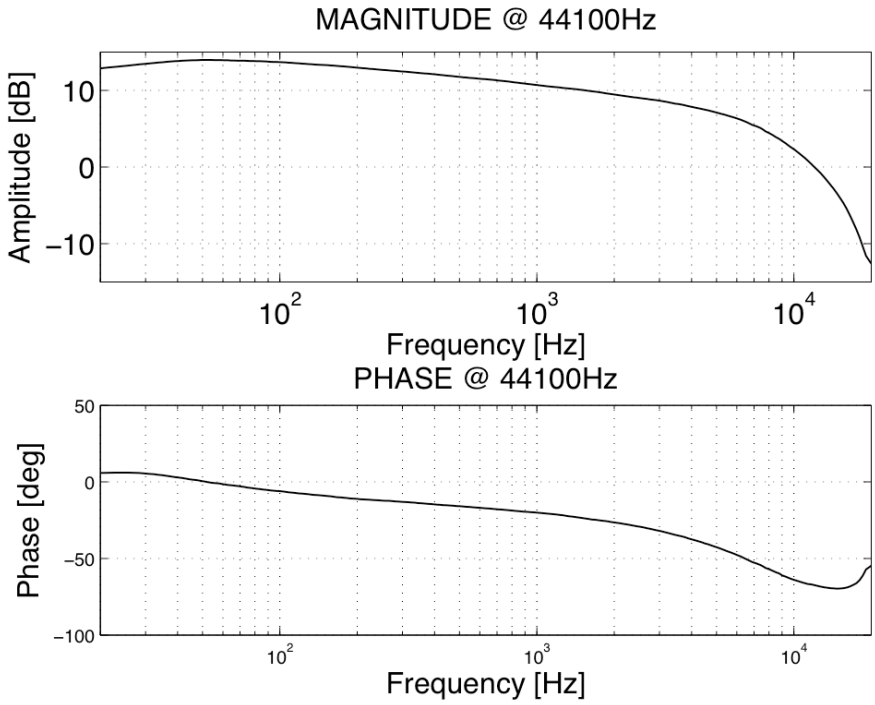

Figure 37 Magnitude vs. frequency and phase vs. frequency of the combined response of a decomposition filter consisting of eight filters.

7.3.2 Gaussian dependency

The second implementation step is to determine a control function which maps the control parameters $fv_m(n)$ to the dependency on other channels. We refer to the enhanced channel as the master reference channel or $fv_\mu(n)$ that corresponds to the classification of the channel the user is willing to enhance. Because $fv_\mu(n)$ has correspondence to a frequency region of the filter bank that was extracted most consistently, we can assume that $fv_\mu(n)$ has most of its spectral content concentrated within that spectral region. Therefore we would like to maximize the attenuation level between the signal level of $xg_\mu(n)$ and the channels which have significant spectral overlap, for example, the channels that have a classification $fv_\mu(n)$ equal to $fv_\mu(n)$. On the other hand, we wish to minimize the amount of attenuation for all $fv_m(n)$ classifications which have little or no spectral relationship to $fv_\mu(n)$. This means that for a non-enhanced channel, the further away the $fv_m(n)$ classification is from $fv_\mu(n)$, the less attenuation is required. This calls for a symmetric function of frequency that provides maximal attenuation at the centre frequency of $fv_\mu(n)$, and smoothly fades to nominal gain as the spectral decomposition classifiers deviate from the value of $fv_\mu(n)$. The attenuation control vector per channel will be given by $cv_m(n)$. This is given by a unitarily normalized Gaussian function

$$fg_m(n) = \frac{1}{Q\sqrt{2\pi}} e^{\frac{-(fr_m(n)-\mu(n))^2}{2Q^2}}, \qquad (27)$$

where $fg_m(n)$ is a cross-adaptive function, and the term $fr_m(n)$ represents frequency, given the function $\mu(n)$ determines the position of the axis of the Gaussian function. Q controls the spread of the Gaussian function and may be given by a user-selected variable that directly controls the rate of attenuation for channels with overlapping frequency content. We then proceed to modify $fg_m(n)$ to fit the design requirements by performing the following steps.

First we normalize $fg_m(n)$, then we obtain the absolute value of its unitary complement, and finally we add a user controllable attenuation variable, *G*. The attenuation variable allows the user to select the amount of attenuation applied at the maximum of the Gaussian function. This is presented in equation 28, where $a_m(n)$ is the inter-channel dependency mapping function corresponding to an enhancement couture curve.

$$a_m(n) = \left| \left(G \cdot fg_m(n) \cdot Q\sqrt{2\pi} \right) - 1 \right| \qquad (28)$$

Given that we require that the axis of the Gaussian to be centered at $fv_\mu(n)$ we must relate $\mu(n)$ to $fv_\mu(n)$. The algorithm has a maximum of *K-1* filters comprising the filter bank, where *K-1* is equal to the total number of channels *M-1*. So $fv_\mu(n)$ must be normalized with respect to *M-1* in order for $\mu(n)$ to be centered exactly at $fv_\mu(n)$. This normalization is presented in equation 29.

$$\mu(n) = \left(\frac{2}{M-1} \left(fv_\mu(n) - 1 \right) \right) - 1 \qquad (29)$$

Recall that our objective is to enhance the master channel, $xg_m(n)$, by reducing the amount of spectral overlap it has with the rest of the mix. So we must keep the master channel gain unchanged while performing a spectrally dependent attenuation to other channels. In other words, the resulting control gain value for the master channel $a_m(n)$ for $m=\mu$ must always be equal to one, while the control gain value for each of the remaining channels, $a_m(n)$ for $m \neq \mu$, must be given by evaluating $fr_m(n)$ in equation 27 with respect to its corresponding $fv_\mu(n)$ spectral classification. Given that the algorithms has a filter bank with the same number of filters as channels filters, we must normalize $fv_\mu(n)$ with respect to *M-1* before evaluating $fr_m(n)$, this normalization Is given by equation 30

$$fr_m(n) = \left(\frac{2}{M-1} \left(fv_m(n) - 1 \right) \right) - 1 \qquad (30)$$

.

Given that our objective is to enhance $x_\mu(n)$ with respect to the rest of sources, we must maintain the gain of $x_\mu(n)$ unchanged. This is expressed by

$$cv_m(n) = \begin{cases} 1 & m = \mu \\ a_m(n) & m \neq \mu \end{cases} \qquad (31)$$

A detail cross-adaptive implementation of such an automatic mixing tool is depicted in Figure 38.

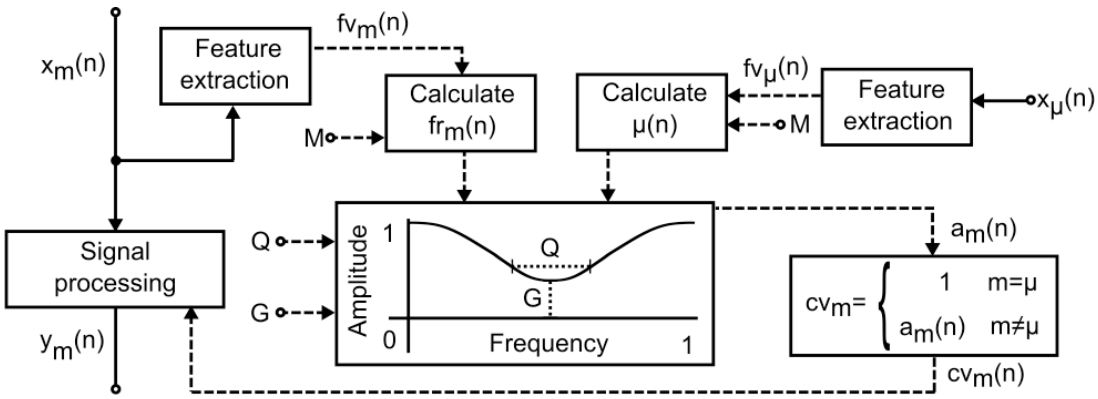

Figure 38 Detailed block diagram of the Gaussian inter-channel dependency algorithm.

It can be seen that the five variables needed by the algorithm are:

A) Channel number location: This is the location of the channel querying a control value and has been depicted in Figure 39. It corresponds to the channel to which the control variable result will have a direct effect. This can be automatically obtained from the host and does not require user input.

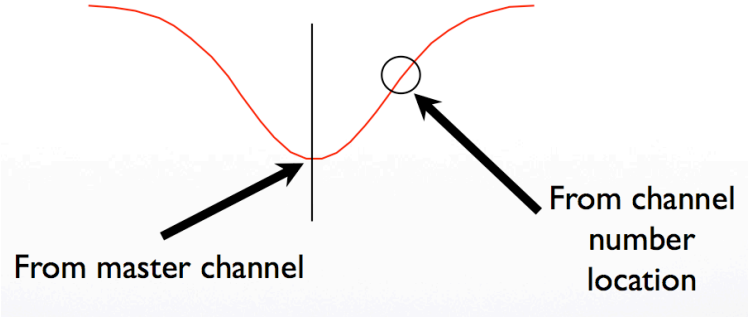

Figure 39 How to read the corresponding $cv_m(n)$ from the enhancement contour according to the Channel number location given by $fv_m(n)$.

B) Master Channel: This is the channel that the user wishes to enhance, depicted in Figure 40. This variable is user selected, and must be selected at the beginning of the process.

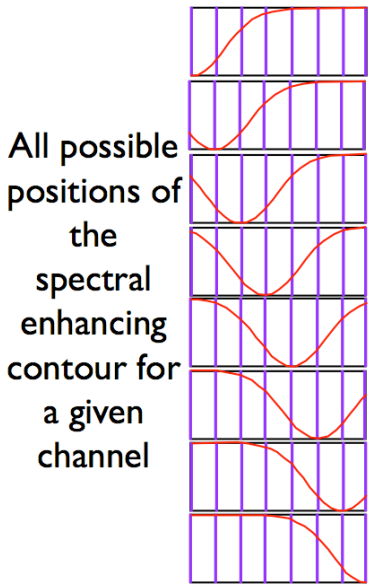

Figure 40 All possible master channel $fv_m(n)$ values for an example filter bank of K=8.

C) Total number of channels: This corresponds to the overall amount of channels involved in the Cross-Adaptive processing, Figure 41. This variable is user selected, and must be selected at the beginning of the process.

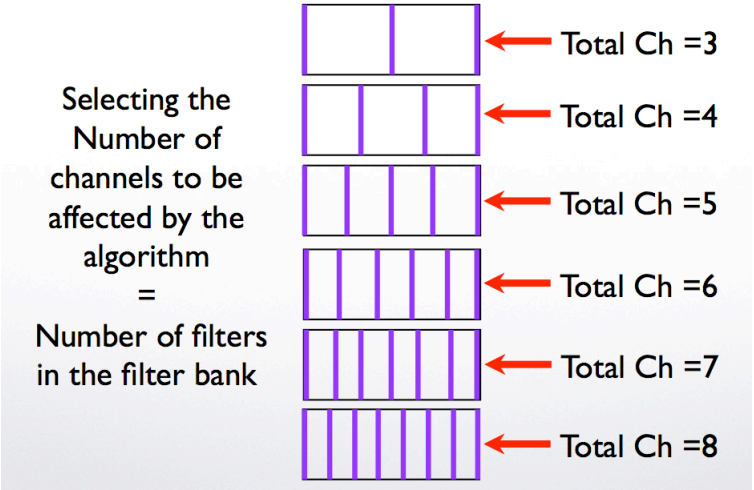

Figure 41 Vertical lines representing for a filter banck of 3 filters up to 8 filters.

D) Attenuation: This is the amount of maximum attenuation applied to sources that are directly related to the spectrum classification of the master channel selected. This variable is user selected. Several possible steps of attenuation have been plotted next in Figure 42.

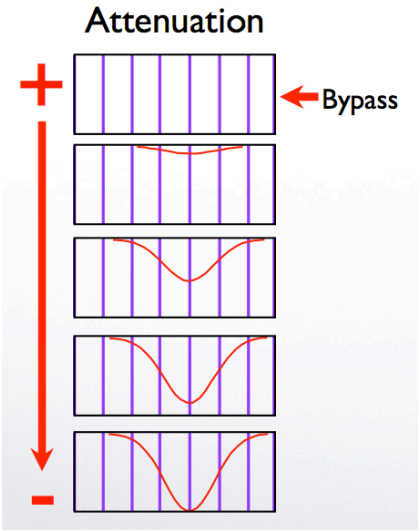

Figure 42 Five different attenuation settings.

E) Q: Corresponds to the smoothness quality factor of the Gaussian curve which controls the attenuation spread over the neighbor classified with a different class than the master channel spectral class. This variable is also user selected. Several possible steps of Q have been plotted next in Figure 43.

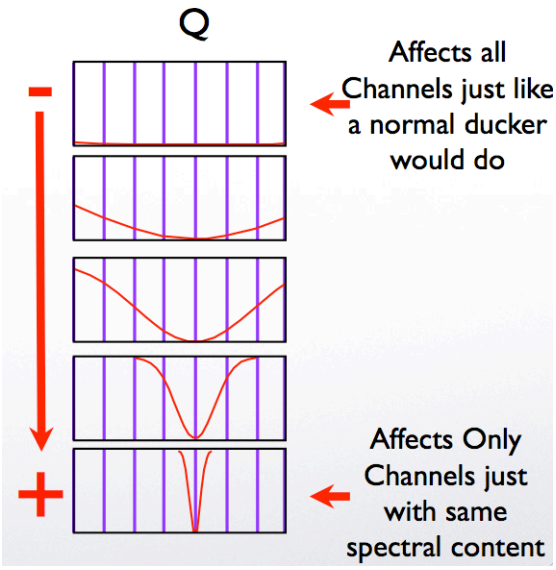

Figure 43 Five different Q settings.

The final graphic interpretation of such a cross-adaptive enhancer is depicted by the following illustration in Figure 44.

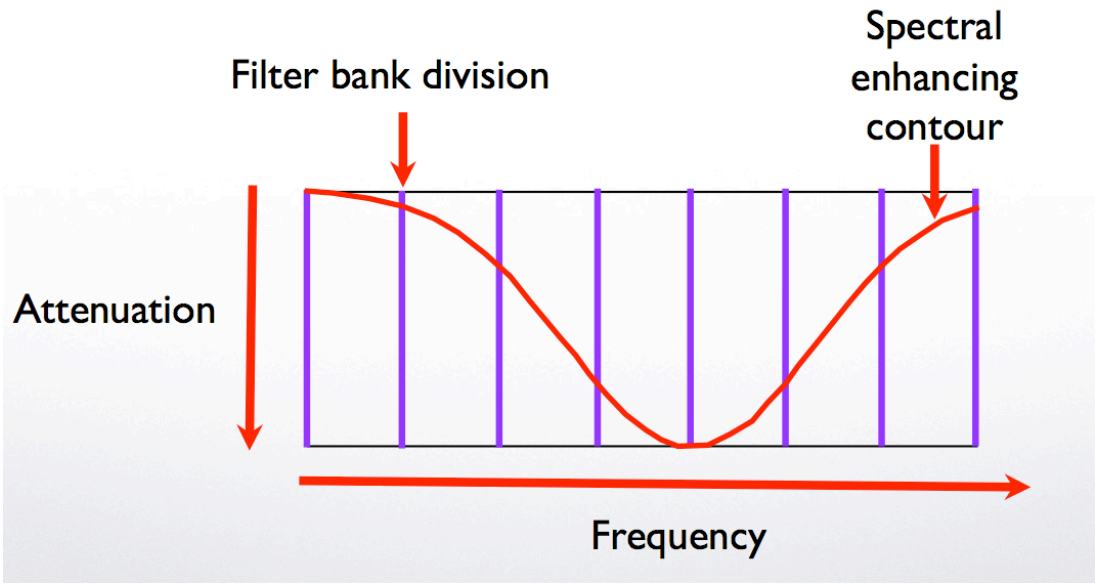

Figure 44 Enhancement contour for a mid Q with a 8 filter bank decomposition algorithm with the reference master channel centered at k=4 and maximum attenuation.

7.3.3 Algorithm applications to enhancement

The algorithm presented in the previous section devises a Gaussian inter-channel dependency value for every channel. It can be used to determine the amount of gain applied to each channel of an audio mix. This approach ensures minimal spectral masking while affecting the level of the mixed sources in proportion to their spectral relation to the master channel. A flow diagram of this algorithm is depicted next in Figure 45.

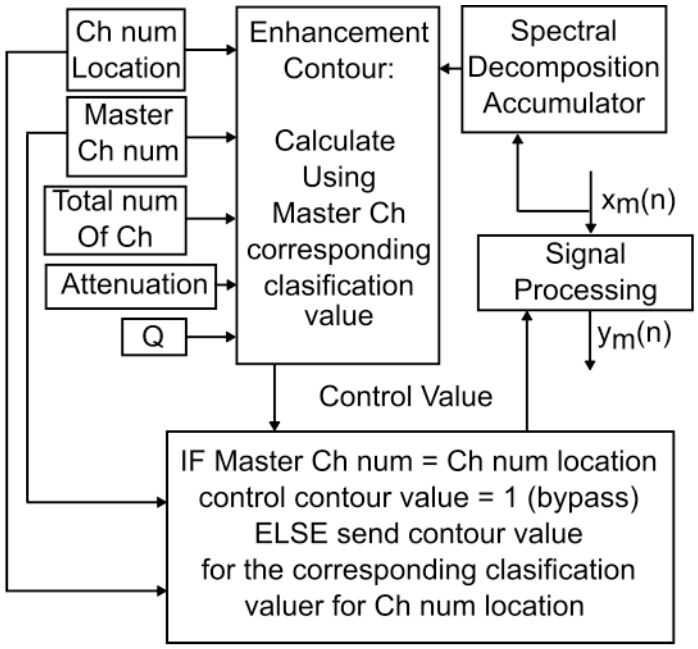

Figure 45 Algorithmic block diagram of the Gaussian inter-channel dependency algorithm.

Such a system would be governed by a cross-adaptive mixing function such as equation 32

$$y_{mix}(n) = \sum_{m=0}^{M-1} cv_m(n) \cdot x_m(n) \quad (32)$$

,

where $y_{mix}(n)$ is the overall mix after applying the cross-adaptive effect, $cv_m(n)$ is the control value for every channel, $x_m(n)$, and $cv_m(n)$ is equal to one for $x_m(n) = x_\mu(n)$. Where m corresponds to every channel involved in the cross-adaptive mix and takes a value from 1 to $M-1$, where M corresponds to the total number of channels involved in the cross-adaptive mix process. Compared to a system that performs a similar task by using equalization filters, the proposed approach has no channel phase distortion.

Another possible implementation of the algorithm for stereo applications is to reduce directional masking. Directional masking is the equivalent of spectral masking but in the phase domain. Directional masking can be reduced by de-correlating the phase information of the right channel against the left channel. Therefore, the greater the de-correlation the more diffuse the sound, and the more correlated the left and the right channels are, the more present the channel is. By using pseudo-stereo techniques proposed in (Gerzon 1992), which split monaural sources and applies all-pass filter networks to the pseudo-left and pseudo-right channels, a stereo effect can be achieved. It is thought that such an effect reduces listening fatigue and enhances the content of the channel to which the pseudo-stereo technique is applied (Mongomery 2007). The all-pass filter network used for such a method is given by $H_L(n)$. We can generate a cross-adaptive effect that enhances a target channel by reducing its directional masking by using

$$y_L(n) = \frac{1}{\sqrt{2}}\left[\begin{pmatrix}\sum_{m=0}^{M-1}\sin(90[1-cv_m(n)])\cdot H_L[x_m(n)] + \cos(90cv_m(n))\cdot x_m(n)\\ -(\sin(90[1-cv_\mu(n)])\cdot H_L[x_\mu(n)] + \cos(90cv_\mu(n))\cdot x_\mu(n))\end{pmatrix} + x_\mu(n)\right] \quad (33)$$

and

$$y_R(n) = \frac{1}{\sqrt{2}}\sum_{m=0}^{M-1} x_m(n) \quad (34)$$

,

where the mix of the left and right channel y is given by $y_L(n)$ and $y_R(n)$ respectably, given that we desire an implementation that only affects the phase and not the gain; care has to be taken to ensure that the operations applied ensure unitary gain. First the inter-dependency control variable $cv_m(n)$ has been scaled to represent a maximum of 90 degrees and integrated to a sine/cosine law (Griesinger 2002) to preserve overall power. Finally a $1/\sqrt{2}$ term has been used to preserve the overall power of the constructive interaction of the left and right channel. For this application we must ensure the enhanced target channel does not suffer any diffusion due to the all-pass filter networks. For this reason when $cv_m(N) = cv_m(n)$, $cv_m(n)$ must be equal to one.

7.3.4 Algorithm interface

In order for the user to have access to the effect, a graphical user interface was implemented and depicted in Figure 46. The user interface is arranged in a standard frequency vs. amplitude plot. The vertical lines show the location of the h_k filters (K=8 on Figure 46), the user has access to changing the number of h_k filters shown by changing the amount of channels to which the cross-adaptive effect is to be applied. A plot of the intersection of the Gaussian dependence function, $cv_m(n)$, with the h_k filters is also depicted. The user also has control access for the attenuation and Q of the algorithm. The user can choose the channel to be enhanced, and this automatically sets it as the master channel. The master control interface must be hosted separately from any individual channel host interface, as it is the interface for cross-controlling all channels.

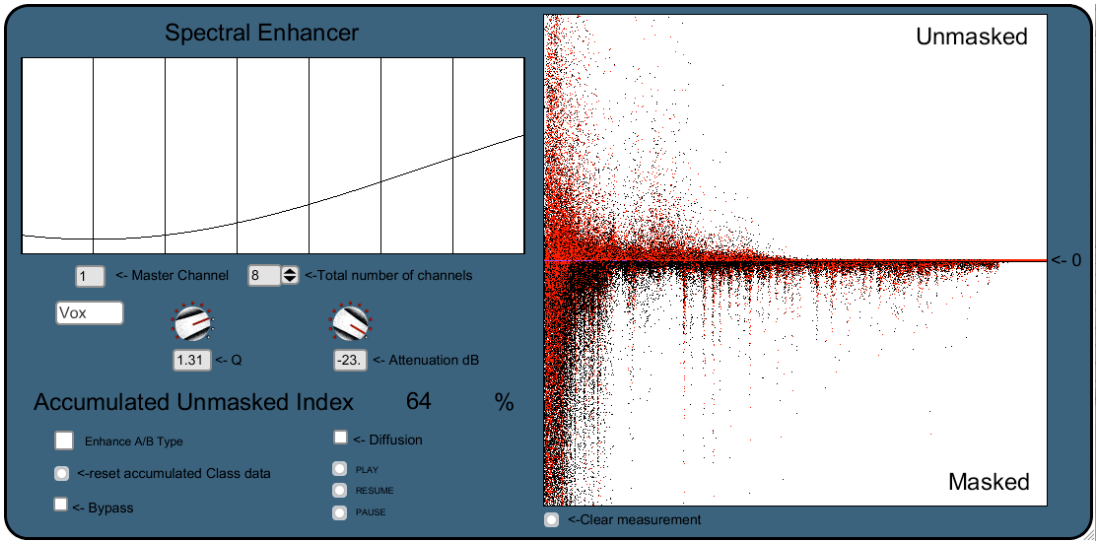

Figure 46 Master user control interface

Since there is an actual signal processing happening on every channel a processing device must be contained within each channel. This signal processing device contains a small host interface. This signal processing device is controlled by the inter-dependent variables $cv_m(n)$ and in the case of the implementation proposed here, it requires a channel location identifier, which can be automatically assigned by the host. The channel also needs to know if it is a master channel or a slave channel, and this is automatically given by the user selected enhanced channel on the master user interface. Finally, for convenience of the user, a button to call the master user interface from any channel has been included. A depiction of the host interface located on every channel is presented next on Figure 47.

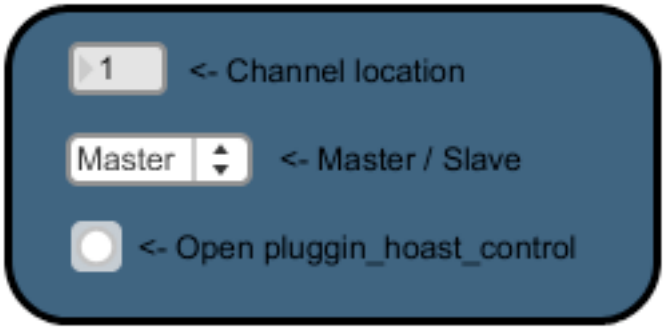

Figure 47 Host channel interface.

7.4 Test and results

To determine the effectiveness of the algorithm, a masking-improvement meter was developed. With the aim of obtaining a perceptual improvement measurement a quantised version of equation 25 was implemented. The quantized implementation was calculated once for the masked index before the effect was applied and once for the masked index after the effect has been applied. All implemented measurements use a FFT frame size of N=1024 samples. In order to measure the reduction in spectral masking due to the technique, a simple quantization function was applied to the frequency bins of each frame. Quantisation was performed for all bins for every given frame. The equations used for implementing such a quantisation are given by equations

$$Qx_{\mu,i}(k) = \begin{cases} 1 \quad if \quad \left|X_\mu(k)\right|_i - \left|x_{mix-\mu}(k)\right|_i > 0 \\ 0 \quad if \quad \left|X_\mu(k)\right|_i - \left|x_{mix-\mu}(k)\right|_i \le 0 \end{cases} \qquad (35)$$

and

$$Qy_{\mu,i}(k) = \begin{cases} 1 \quad if \quad \left|X_\mu(k)\right|_i - \left|Y_{mix-\mu}(k)\right|_i > 0 \\ 0 \quad if \quad \left|X_\mu(k)\right|_i - \left|Y_{mix-\mu}(k)\right|_i \le 0 \end{cases} , \qquad (36)$$

where equation 35 corresponds to the quantised calculation of the masking index before the effect was applied given by $Qx_{\mu,i}(k)$, and equation 36 corresponds to the quantised calculation of the masking index after the effect was applied given by $Qy_{\mu,i}(k)$. Where $X_\mu(k)$=FFT[$x_\mu(n)$], $Y_{mix-\mu}(k)$=FFT[$y_{mix}(n)$-$x_\mu(n)$], and $X_{mix-\mu}(k)$=FFT[$x_{mix}(n)$-$x_\mu(n)$], and $x_{mix}(n)$ is the addition of all inputs before any signal processing, given i is the frame index. A physical implementation of equation 36 and 35 was developed, where each of the resulting quantized bins are plotted, so that any bin greater than zero represents a successful unmasked bin. This implementation is depicted in Figure 48.

Finally a quantitative calculation of the quantised unmasked-rate before and after applying the effect was calculated by equation 37

$$Qp_{\mu,I}(k) = \left(\frac{100 \sum_{i=1}^{I} \left| \sum_{k=0}^{N-1} Qy_{\mu,i}(k) \right|_i}{\sum_{i=1}^{I} \left| \sum_{k=0}^{N-1} Qx_{\mu,i}(k) \right|_i} \right), \qquad (37)$$

where $Qp_{\mu,I}(k)$ is the unmasked-rate percentage, obtained by calculating the ratio of the quantized accumulated spectral masking of channel μ before and after applying the enhancer.

This implementation gives the rate difference between the successfully un-masked bins before and after the cross-adaptive effect has been applied. It represents the percentage of masking improvement of using the effect against not using it.

The accumulated masking spectral index for the mix before and after applying the effect were depicted as a visual aid based on the implementation of equations 35 and 36. The result of this implementation is shown on Figure 48, where all successfully unmasked spectral data has been depicted as falling below the zero crossing threshold. The spectral masking index before applying the effect is depicted in black while the spectral masking index after applying the effect is depicted in grey. A perceptual improvement $Qp_{\mu,I}(k)$ based on equation 37 is also shown. Results on an enhanced guitar show up to an $Qp_{\mu,I}(k)$=310% improvement, making the presence of the enhanced material more tangible. It was found that an excessive enhancement, in the order of $Qp_{\mu,I}(k)$=300% % will practically isolate the enhanced source from the overall mix, so a more moderate enhance of $Qp_{\mu,I}(k)$=60% is recommended. If an excessive $Qp_{\mu,I}(k)$ is required to produce the desired unmasking, pre-equalisation process may be required.

For the purpose of accuracy all measurements are reset, plotted and re-calculated every time the user changes a parameter in the user interface of the cross-adaptive effect.

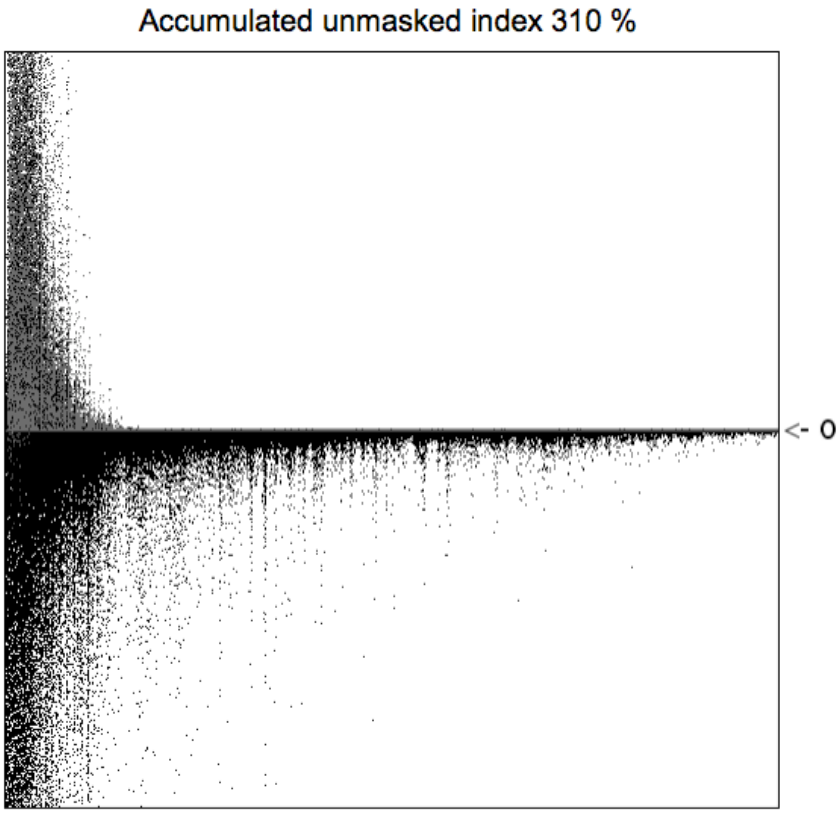

Figure 48 Accumulated masking index visualization interface.

7.5 Summary

An automatic mixing tool that uses channel inter-dependency spectral features for enhancement purposes has been implemented. The effect simplifies the complex mixing task of rescaling the levels of multiple sources with respect to their spectral content in order to enhance a source. The user can control the amount of un-masking by changing the Q and attenuation parameters. This controls the inter-channel dependant characteristics of the effect. The effect has a visual measurement display that permits quantifying the amount of enhancement applied in terms of the spectral masking improvement. This research also underlines the need for a dedicated cross-adaptive, inter-channel dependency effect host.

Chapter 8
Automatic panning

8.1 Introduction

Stereo panning aims to transform a set of monaural signals into a two-channel signal in a pseudo-stereo field (Gerzon 1992). Many methods and panning ratios have been proposed, the most common one being the sine-cosine panning law (Griesinger 2002; Anderson 2008). A common task in live mixing is down mixing a series of mono inputs into a two channel stereo mix. For doing this the input channels get summed into a Left (L) and a Right (R) channel bus. The proportion at which these multiple mono inputs are added to each L and R channels are responsible for the perceived stereo image. Over the years the use of panning on music sources has evolved and some common practices can now be identified.

Previous related work on down mixing for spatial audio coding, from 5.1 surround to 2.0 stereo, has been attempted by (Schick et al. 2005). Processing of multiple channels for real time applications using priority has been attempted by (Tsingos 2005), but this method requires an off-line processing stage which requires pre-processing of the audio channel in order to enhance them with descriptors. This method is suitable for game and simulations but is not optimal for live environments where the signal nature is unknown. Work on up mixing has been researched by (Advendano and Jot 2004; Li and Driessen 2005). In their work, they describe methods to turn a stereo down mix into a multi-channel up mix. Although these methods can prove useful if backtracked, they are more suitable for multi-channel surround format conversion rather than for multiple input mixing. By multiple input channels we refer to the

individual instruments of a live group of musicians or multiple speech inputs as opposed to multi-channel format sub-mixes, as contained in 5.1 surround formats. Currently, there is no known approach to automatic stereo down-mixing multiple inputs channels in a live real time environment.

This chapter presents an expert system capable of characterizing multi-track inputs and autonomously panning sources with panning results comparable to a human mixing engineer. This was achieved by developing cross-adaptive rules that take into account technical constraints and common practices for panning, while minimizing human input. Two different approaches are described and subjective evaluation demonstrates that the automatic panner has equivalent performance to that of a professional mixing engineer.

8.2 Automatic panner

In practice, the placement of sound sources is achieved using a combination of creative choices and technical constraints based on human perception of source localization. It is not the purpose of this automatic mixing tool to emulate the more artistic and subjective decisions in source placement. Rather, we seek to embed the common practices and technical constraints into an algorithm that automatically places sound sources. The idea behind developing an expert automatic panning machine is to use well-established common rules to devise the spatial positioning of a signal. A list of seven common panning practices in music mixing is presented in the following page.

List of seven common panning practices:

1) When the human expert begins to mix, he or she tends to do it from a monaural, all centered position, and gradually moves the pan pots (Self and et al. 2009). During this process, all audio signals are running through the mixer at all times. In other words, source placement is performed in real time based on accumulated knowledge of the sound sources and the resultant mix, and there is no interruption to the signal path during the panning process.

2) Panning is not the result of individual channel decisions; it is the result of an interaction between channels. The audio engineer takes into account the content of all channels, and the interaction between them, in order to devise the correct panning position of every individual channel (Neiman 2002).

3) The sound engineer attempts to maintain balance across the stereo field (Izhaki 2007). This help maintain the overall energy of the mix evenly split over the stereo speakers and maximizes the dynamic use of the stereo channels.

4) In order to minimize spectral masking, channels with similar spectral content are placed apart from each other (Neiman 2002; Bartlett 2009). This results in a mix where individual sources can be clearly distinguished, and this also helps when the listener uses the movement of his or her head to interpret spatial cues.

5) Hard panning of monaural sources is uncommon (Owsinski 2006). It has been established that panning a ratio of 8 to 12 dBs is more than enough to achieve a full left or full right image (Rumsey and McCormick 2006). For this reason, the width of the panning positions is restricted.

6) Low frequency content should not be panned. There are two main reasons for doing this. First, it ensures that the low frequency content remains evenly distributed across speakers (White 2000). This minimizes audible distortions that may occur in the high power reproduction of low frequencies.

Second, the position of a low frequency source is often psycho-acoustically imperceptible. In general, we can not correctly localize frequencies lower than 200Hz (Benjamin 2006). It is thought that this is due to the fact that the use of inter-aural time difference as a perceptual clue for localization of low frequency sources is highly dependent on room acoustics and loudspeaker placement, and Inter-Level Differences are not a useful perceptual cue at low frequencies since the head only provides significant attenuation of high frequency sources (Beament 2001).

7) High priority sources tend to be kept towards the centre, while lower priority sources are more likely to be panned (Izhaki 2007). For instance, the vocalist in a modern pop or rock group (often the lead performer) would often not be panned. This relates to the idea of matching a physical stage setup to the relative positions of the sources.

8.3 Research and implementation

8.3.1 Cross-adaptive implementation

The automatic panner is implemented as a cross-adaptive effect, where the output of each channel is determined from analysis of all input channels (Verfaille 2006). For applications that require a real time signal processing, the signal analysis and feature extraction has been implemented using side chain processing. The audio signal flow remains real time while the required analysis of the input signals is performed in separate instances. The signal analysis involves accumulating a weighted time average of extracted features. Accumulation allows us to quickly converge on an appropriate panning position in the case of a stationary signal, or smoothly adjust the panning position as necessary in the case of changing signals. Once the feature extraction within the analysis side chain is completed, then the features from each channel are analyzed in order to determine new panning positions for each channel. Control signals are sent to the signal processing side in order to trigger the desired panning commands.

8.3.2 Adaptive gating

Because noise on an input microphone channel may trigger undesired readings, the input signals are gated. The threshold of the gate is determined in an adaptive manner. By noise we refer not only to random ambient noise but also to interference due to nearby sources, such as the sound from adjacent instruments that are not meant to be input to a given channel.

Adaptive gating is used to ensure that features are extracted from a channel only when the intended signal is present and significantly stronger than the noise sources. The gating method based on a method implemented in (Dugan 1975; Dugan 1989). A reference microphone may be placed outside of the usable source microphone area to capture a signal representative of the undesired ambient and interference noise. The reference microphone signal is used to derive an adaptive threshold by opening the gate only if the input signal magnitude is greater than the reference microphone magnitude signal. Therefore the input signal is only passed to the side processing chain when its level exceeds that of the reference microphone signal. This process is depicted in section 3.5.

8.3.3 Filter bank implementation

The implementation uses a filter bank to perform spectral decomposition of each individual channel. The filter bank does not affect the audio path since it is only used in the analysis section of the algorithm. It was chosen as opposed to other methods of classifying the dominant frequency or frequency range of a signal (Sethares and et al. 2009) because it does not require Fourier analysis, and hence is more amenable to a real time implementation.

For the purpose of finding the optimal spectral decomposition for performing automatic panning, two different eight band filter banks were designed and tested. The first consisted of a quasi-flat frequency response band-pass filter bank, which for the purposes of this chapter we will call filter bank type A in Figure 49, and the second contained a low-pass filter decomposition filter bank, which we will call filter bank type B in Figure 50. In order to provide an adaptive frequency resolution for each filter bank, the total number of filters, *K*, is equal to the number of input channels that are meant to be panned. The individual gains of each filter were optimized to achieve a quasi-flat frequency response.

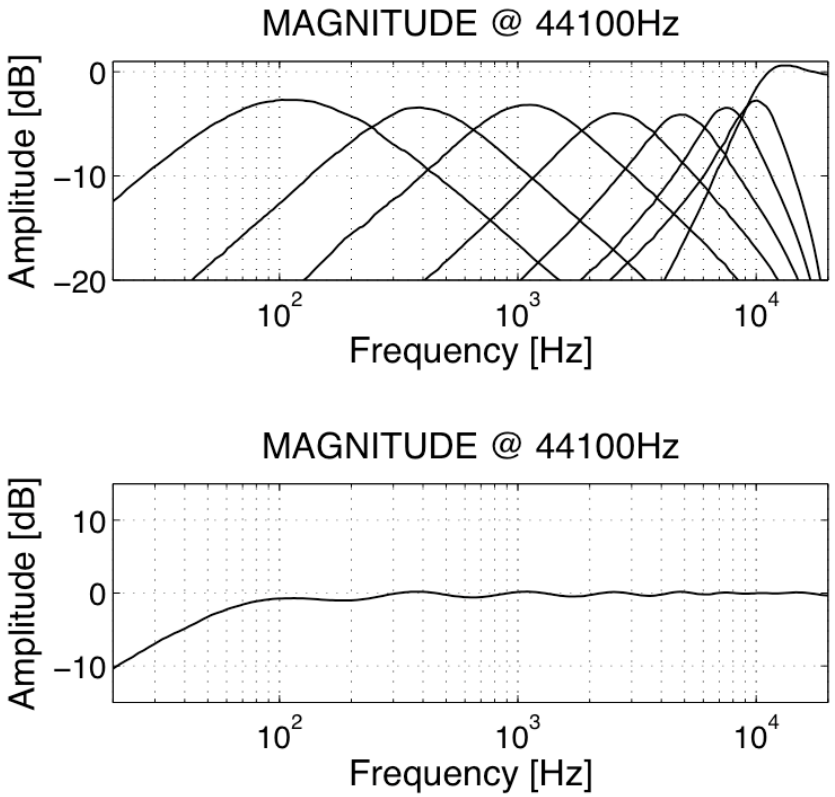

Figure 49 Quasi-flat frequency response band-pass filter bank. (Type A filter bank for *K=8*). Top, filter bank consisting of a set of eight second order band-pass IIR Biquadratic filters with center frequencies as follow: 100Hz, 400Hz, 1kHz, 2.5kHz, 5kHz, 7.5kHz, 10kHz and 15000kHz. Bottom, combined response of the filter bank.

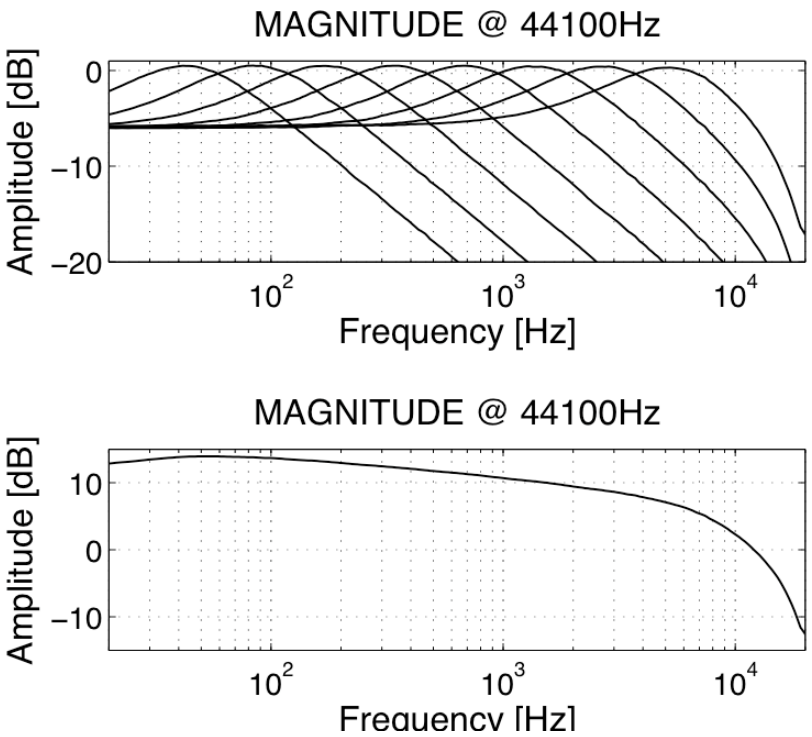

Figure 50 Low-pass filter decomposition filter bank. (Type B filter bank for *K=8*). Top filter bank comprised of a set of second order low-pass IIR Biquadratic filters with cut off frequencies as follows: 35Hz, 80Hz, 187.5Hz, 375Hz, 750Hz, 1.5kHz, 3kHz and 6kHz. Bottom, combined response of the filter bank. All gains have been set to have a maximum peak value of 0dBs.

8.3.4 Determination of dominant frequency range

Once the filter bank has been designed the algorithm uses the band-limited signal in each filter's output to obtain the absolute peak amplitude for each filter. The peak amplitude is measured within a 100ms window. The algorithm uses the spectral output of each filter contained within the filter bank to calculate the peak amplitude of each *k* band. By comparing these peak amplitudes, the filter with the highest peak is found. An accumulated score is maintained for the number of occurrences of the highest peak in each filter contained within the filter bank. This results in a classifier that determines the dominant *k* filter band for an input channel taken from the highest accumulated score. The spectral calcification per channel is denoted by $fv_m(n)$ where $fv_m(n)$ takes a number from 0 to K-1 which corresponds to highest accumulated filter.

The block diagram of the filter bank analysis algorithm is provided in Figure 51. It should be noted that the accumulated score of the signal *k(n)*, that contains the dominant filter band, can be implemented using digital logic operations of comparison. Thus it can be implemented in a single clock cycle implementation, which makes it highly attractive for an efficient digital implementation.

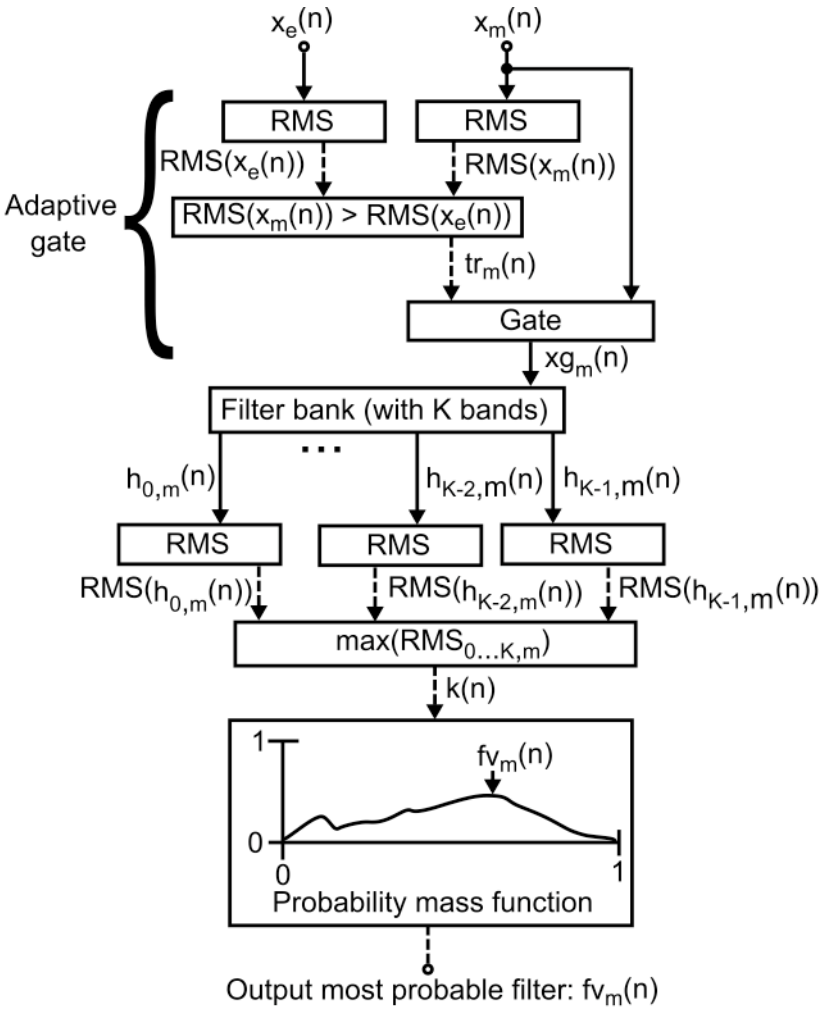

Figure 51 Analysis block diagram for one input channel.

8.3.5 Cross-adaptive mapping panning rules

Now that each input channel has been analyzed and associated with a filter, it remains to define a mapping that results in the panning position of each output channel. The rules that drive this cross-adaptive mapping are presented in the next page.

First rule is what is refer as the user priority rule; in which the user identifies his subjective preference over which musical source is more important by labeling in consecutive order from the most important to the least important. The common practices mentioned earlier would suggest that certain sources, such as lead vocals, would be less likely to be panned to extremes than others, such as incidental percussions. However, the current implementation of our automatic panner does not have access to such information. Thus, in this thesis it has been proposed to use a priority driven system in which the user can label the channels according to importance. In this sense, it is a semi-blind automatic system. Thus, all sources are ordered from highest to lowest priority. In the current implementation this is done by connecting the sources to the mixer in order of importance where the musical source connected to channel one is the more important and the channel connected to the last channel is the least important.

The second rule is the use of spectral content to spread the panning position spreading evenly sources with same spectral content. This means that for the sources residing in the same spectral classification, the first panning step is taken by the highest priority source, the second panning step by the next highest priority source, and so on. This means that if we hade 3 sources classified in the same spectral category the one with highest priority will be paned o the center and the other two wish are lower priority will be evenly paned t the sides but in opposite directions. The procedure for achieving the panning position based on user priority and spectral content is presented next.

In order to assign a panning position per source we must be able to identify the total number of sources in the mix with same spectral classification, denoted as R_m, and the relationship between the user priority and its spectral classification given by P_m. We can then calculate the panning position of a source based on the obtained parameters R_m and P_m. Equation 38 is used to obtain the total number of classification repetitions due to other signals having the same k filter classification, given the initial condition $R_0=0$.

$$R_m = \sum_{j=1}^{M} R_{j-1} + \begin{cases} 1 & fv_m(n) = fv_{j-1}(n) \\ 0 & fv_m(n) \neq fv_{j-1}(n) \end{cases} \quad (38)$$

Now we proceed to calculate P_m, the relationship between the user assigned priority of a source denoted U_m and its spectral classification $fv_m(n)$. The user assigned priority U_m has a unique value from *0, ..., M−1*, the smaller the magnitude of U_m, the higher the priority. The assigned priority due to being a member of the same spectral classification, P_m, has a valid range from 1 to its corresponding R_m value. The lower the value taken by P_m, the lower the probability of the source of being widely panned. P_m is calculated by equation 39

$$p_m = \left| \{U_i : fv_i(n) = fv_m(n)\} \bigcap \{U_i : U_i \leq U_m\} \right| \text{ for } i = \{0 \ldots M-1\}, \quad (39)$$

where the modulus of the intersection of the two sets, $\{ U_i : fv_i(n) = fv_m(n) \}$ and $\{ U_i : U_i \leq U_m \}$ gives us the rank position, which corresponds to the value taken by P_m. Given R_m and P_m we can relate them in order to obtain the panning control parameter with equation 40:

$$cv_m(n) = \begin{cases} 1/2 & R_m = 1 \\ W + \left[(1-2W) \dfrac{R_m - P_m - 1}{2(R_m - 1)} \right] & P_m + R_m \text{ is odd} \\ W + \left[(1-2W) \dfrac{R_m + P_m - 2}{2(R_m - 1)} \right] & P_m + R_m \text{ is even}, R_m \neq 1 \end{cases} \quad (40)$$

by evaluating R_m and P_m the assigned panning position can be derived. The panning position $cv_m(n)$ has a valid control range from *0* to *1* where *0* means fully panned left, 0.5 means centered and 1 means panned fully right. The panning width limit, *W*, can go from wide panning *W=0* to mono *W= 0.5*.

In the current implementation the panning width control has a default value of *W*=0.059. The *W* value is subtracted for all panning positions bigger than 0.5 and added to all panning positions smaller than 0.5. In order to avoid sources originally panned left to cross to the right or sources originally panned right to cross to the left, the panning width algorithm ensures that sources in such cases default to the centre position.

Finally, the third last panning constraint rule implements the constraint that low frequency sources should not be panned. Thus, all sources with accumulated energy contained in a filter with a cutoff frequency below 200Hz are not panned and remain centered at all times (Benjamin 2006). This is based on the principle that we should not pan a source if its spectral category is too low we set $cv_m(n)$ to be centered if the spectral category of the input source $fv_m(n)$ is less than a psychoacoustically established threshold tr_{ps} ,. This can be implemented by using equation 41, presented next:

$$cv_m(n) = \begin{cases} 1/2 & fv_m(n) \le tr_{ps} \\ cv_m(n) & fv_m(n) > tr_{ps} \end{cases} \qquad (41)$$

Such an automatic panning signal processing tool implementation has been depicted in Figure 52.

8.3.6 The panning processing

Once the appropriate panning position was determined, a sine-cosine panning law (Anderson 2008) was used to place sources in the final sound mix.

$$y_L = \sum_{m=0}^{M-1} \sin(cv_m(n)\pi/2) \cdot x_m(n) \qquad (42)$$

$$y_R = \sum_{m=0}^{M-1} \cos(cv_m(n)\pi/2) \cdot x_m(n) \qquad (43)$$

Where $y_L\ (n)$ and $y_R(n)$ correspond to the automatically panned stereo output of the mixing devise, $cv_m(n)$ is the panning factor and $x_m(n)$ represents the input signals.

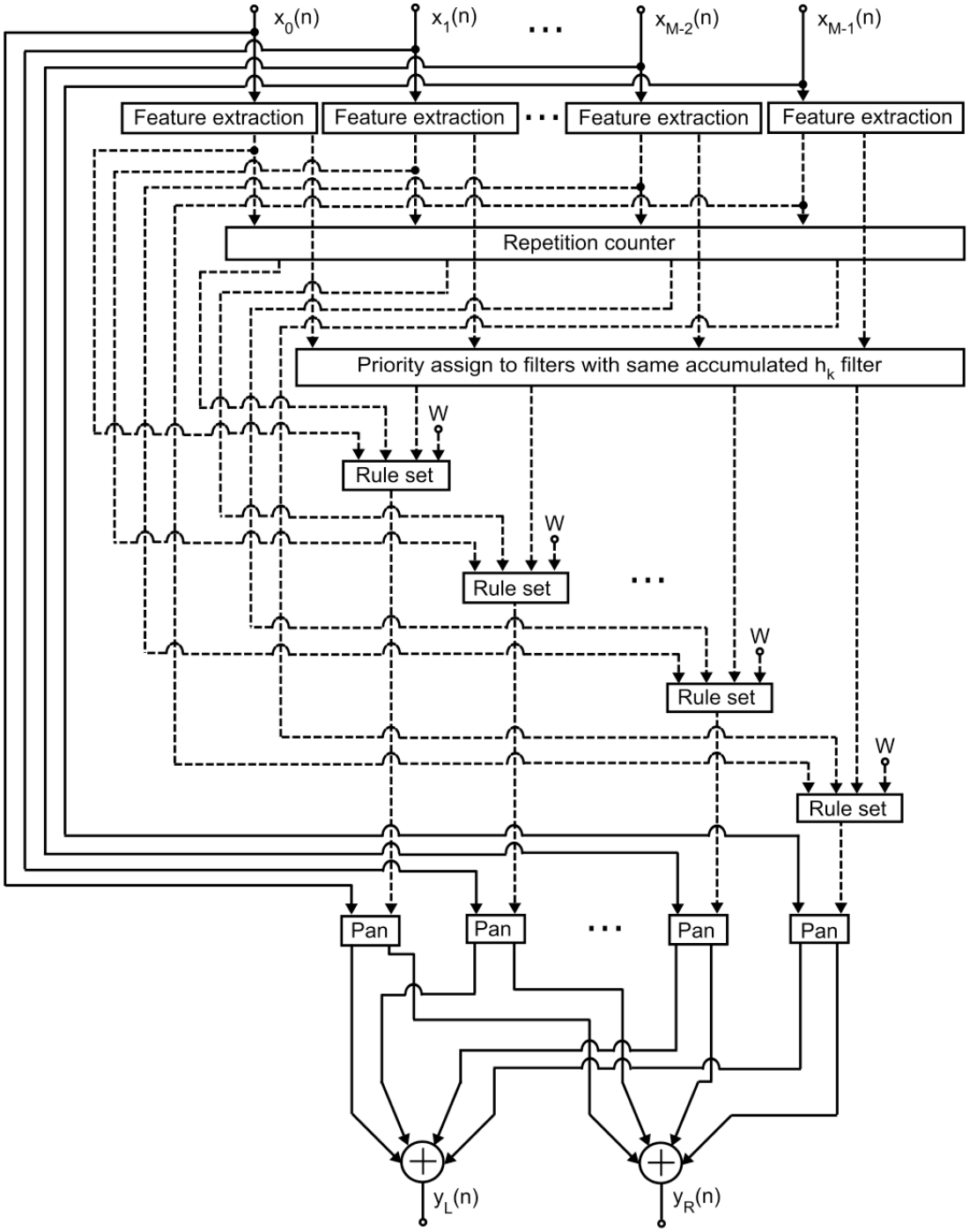

Figure 52 Block diagram of the automatic panner constrained control rules algorithm for M-1 input channels.

In the current implementation an interpolation algorithm has been coded into the panner to avoid rapid changes of signal level. The interpolator has a 22ms fade-in and fade-out, which ensures a smooth natural transition when the panning control step is changed. As a reference the user interface of the auto-panning algorithm is presented in Figure 53.

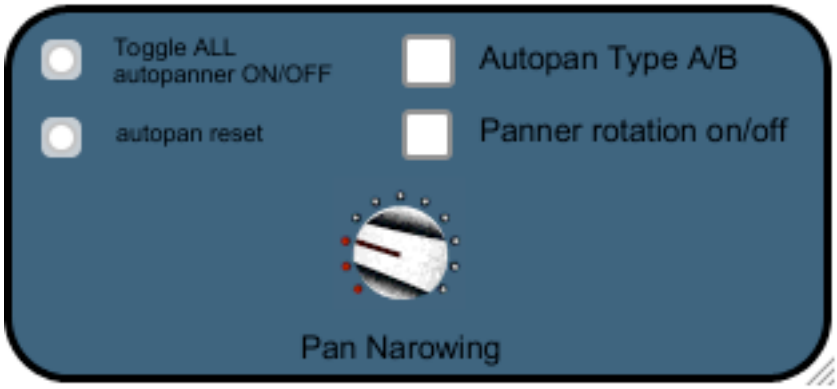

Figure 53 User interface of the auto panning mixing tool.

8.4 Test and results

8.4.1 Objective testing

Several sinusoidal test signals and music tracks simulating a live playing band were used as a mean to test the automatic panning algorithm. The multi-track data used was obtained from the BASS-dB database (Vincent et al. 2006). BASS-dB is the Blind Audio Source Separation evaluation database; it contains links to multi-track recordings which license allows modification and redistribution of the data for non-commercial purposes.

In all studied cases the algorithm was able to converge, this indicated difficult musical signals like drum kits and bass guitar tracks, which contained almost an equal amount of energy in more than one filter, reached a steady

panning position. In Figure 54 we can see the convergence for 4 different sources. The 4 sources were selected from a set of measurements obtained from an 8ch automatic panning downmixer. The plot shows the panning position, $cv_m(n)$, as it approaches stable state, as applied to the input signal.

The doted line, in Figure 54, is the result of plotting the panning position, $cv_m(n)$ of one of 4 tracks that have similar spectral content; the algorithm has spread all four signals equidistantly. $cv_m(n)$ has been smooth using interpolation in order to avoid abrupt changes in panning position. The dotted line corresponds to the highest priority channel out of the four tracks. The dotted line has converged into a panning factor equal to $cv_m(n)=0.33858$. The other 3 sources not shown in this plot converged in accordance to equation 40. Notice that the speed at which the algorithm converges is dependant on the spectral content of the overall input channels. Also a track containing similar spectral content which start later in time than others can cause a panning space reassignation. Others convergence values where the source has been panned fully to the sides have been also plotted for $cv_m(n)=0$ and $cv_m(n)=1$. Finally the solid line represents a drum kit signal, which although the algorithm struggles to decide wheatear its spectral content is of mainly high frequencies (due to the hi-hat) or low frequencies (due to the kick drum) it manages to converge into central position ($cv_m(n)$=0.5), which is technically the most convenient panning position for a signal containing very low frequencies.

In Figure 55 we can see the panning position, $cv_m(n)$, depicted as a solid line in Figure 54 superimposed on the panning position calculated by the algorithm before applying a 2000 sample interpolation to the obtained panning position, $cv_m(n)$. These results show how the interpolation step makes the automatic panner more resilient to panning positioning flutter while achieving a smoother pan displacement.

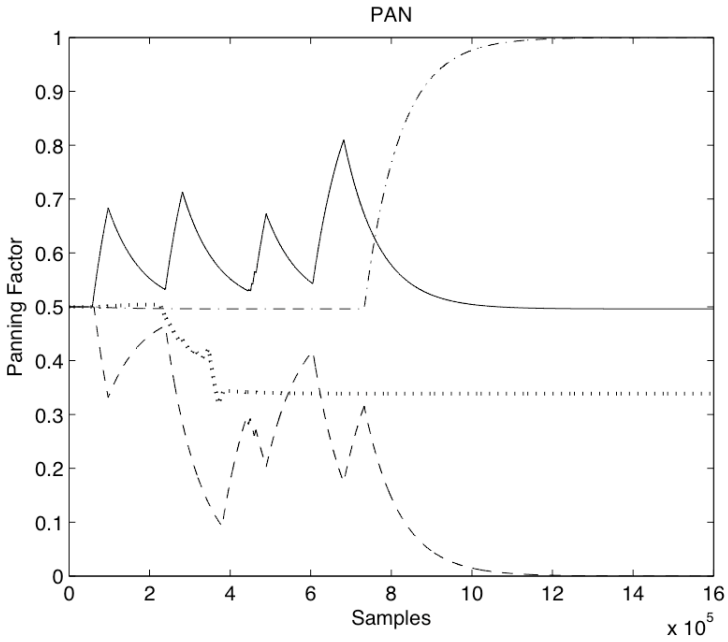

Figure 54 Convergence of automatic panning algorithm for 4 different convergence values. (-) Panning Factor for a drum kit track,(--)panning Factor for a bass guitar, (.-) panning Factor for a vocal track, and (..) panning Factor for a channel input which spectral content is concentrated in the same filter.

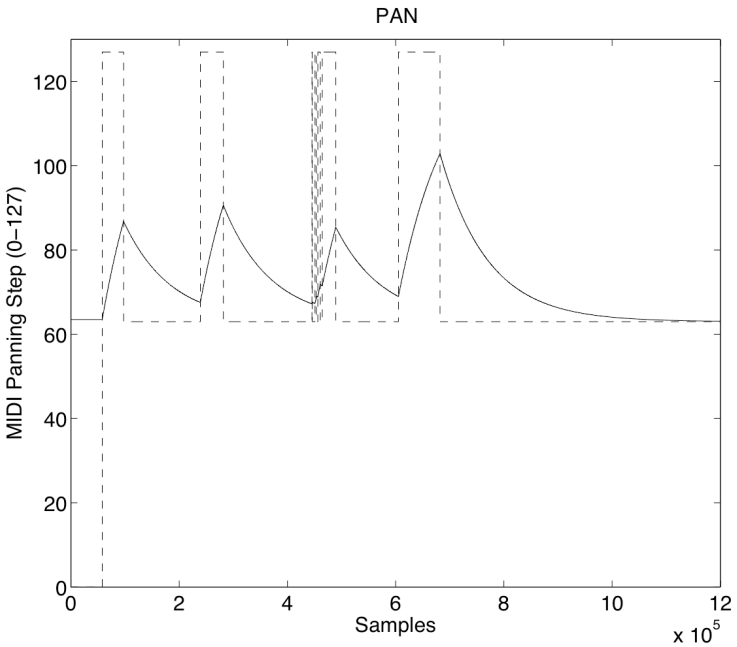

Figure 55 Discrete panning step (- -). Super imposed interpolative panner angle (-) as applied to an input signal consisting of a drum kit recording. A MIDI valid range goes from *0* to *127* therefore the MIDI panning step is given by $127cv_m(n)$.

In Figure 56, the result of down-mixing 12 sinusoidal test signals through the automatic panner is shown. It can be seen that both f_1 and f_{12} are kept centered and added together because their spectral content is below 200Hz. The three sinusoids with a frequency of 5kHz have been evenly spread. f_2 has been allocated to the center due to priority; while f_4 has been send to the left and f_6 has been send to the right, in accordance with equation 40. Because there is no other signal with the same spectral content than f_{11} it has been assigned to the center. The four sinusoids with a spectral content of 15kHz have been evenly spread. Because of priority, f_3 has been assigned a value of 0.33, f_7 has been assigned a value of 0.66, f_9 has been assigned all the way to the left, and f_{10} has been assigned all the way to the right, in accordance with equation 40. Finally, the two sinusoids with a spectral content of 20KHz have been panned to opposite sides. All results prove to be in accordance with the constraint rules proposed for cross-adaptive mapping.

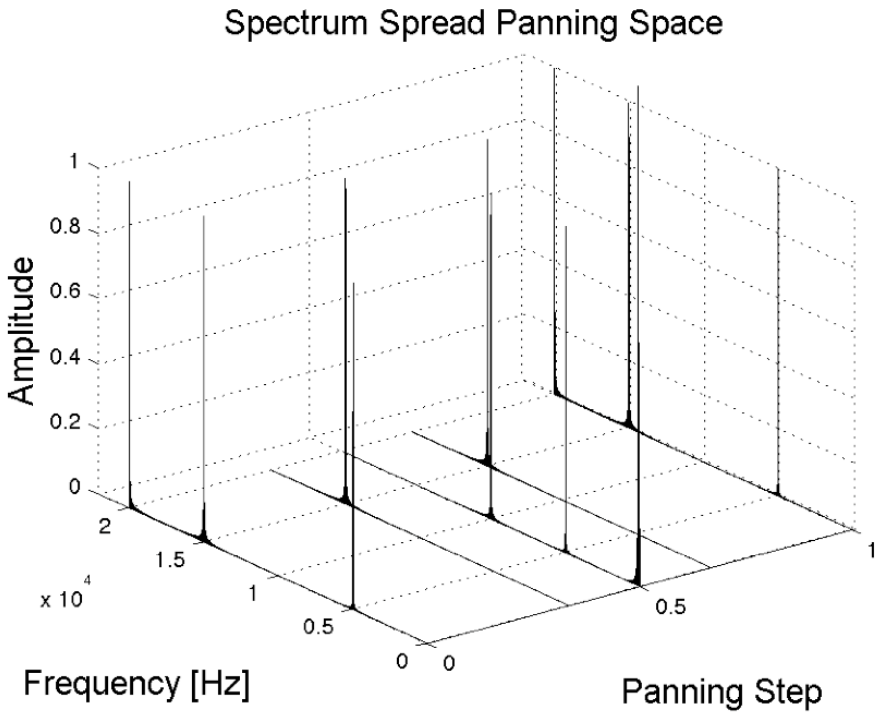

Figure 56 Results of automatic panning based on the proposed design. The test inputs were 12 sinusoids with amplitude equal to one and the following frequencies: f_1=125Hz, f_2=5kHz, f_3=15kHz, f_4=5kHz, f_5=20kHz, f_6=5kHz, f_7=15kHz, f_8=20kHz, f_9=15kHz, f_{10}=15kHz, f_{11}=10kHz, and f_{12}=125 Hz.

A Lissajous curve or stereogram is a two dimensional representation of a stereophonic signal and is usually perform by using an oscilloscope in XY mode or by using a vector oscilloscope. The stereogram can be obtained by plotting in time-synchronicity the left channel against the right channel. This measurement provides detailed information concerning inter-channel phase relationship (Brixen 2007). The data contained in the stereogram of Figure 57 is widely spread in an oval. This means that the phase relation between the left and right channel is close to 90deg. This means we have achieved a wide spread stereo signal. In order to have a reference, a mono signal, which is represented by the diagonal separating the left and right planes of the stereogram has been plotted. The plot also shows a good data equilibrium between the right and left channel.

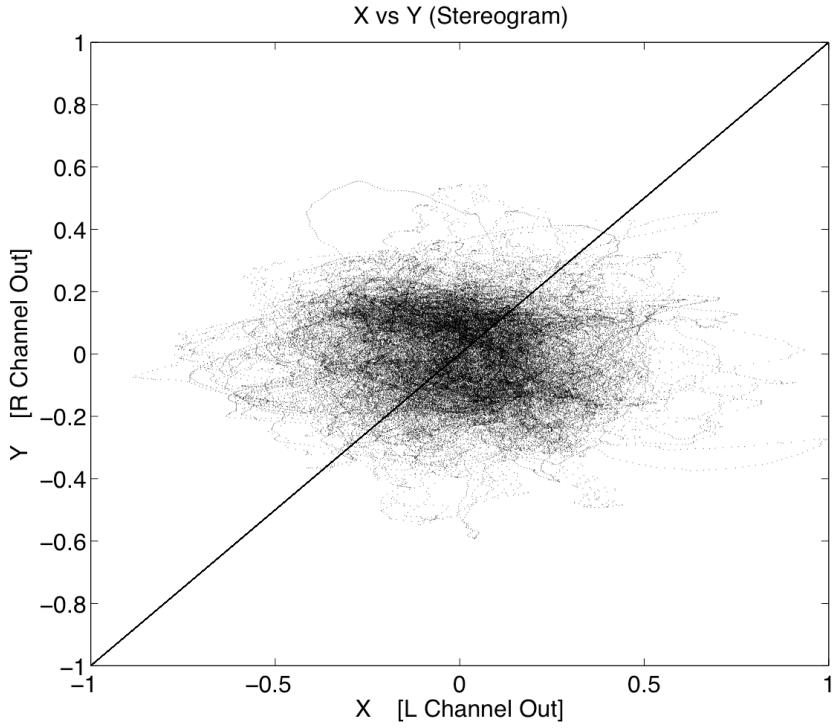

Figure 57 Stereogram of 100,000 samples of a 5CH automatic panner. The samples correspond to a section in time where all 5-channel instruments are interacting simultaneously.

8.4.2 Subjective testing

In order to evaluate the subjective performance of the autonomous panner algorithm against human performance, a double blind test was designed. Both of auto-panning algorithms were tested, the band-pass filter classifier known as algorithm type A, and the low-pass classifier known as algorithm type B. Algorithms were randomly tested in a double blind fashion.

The control group consisted of three professional human experts and one non-expert, who had never panned music before. The test material consisted of 12 multi-track songs of different styles of music. Stereo sources were used in the form of two separate mono tracks. Where acoustic drums were used they would be recorded with multiple microphones and then pre-mixed down into a stereo mix. Humans and algorithms used the same stereo drum and keyboard tracks as separate left and right mono files. All 12 songs were panned by the expert human mixers and by the non-expert human mixer. They were asked to pan the song while listening for the first time. They had the length of the song to determine their definitive panning positions. The same songs were passed through algorithms A and B only once for the entire length of the song. Although the goal was to give the human and machine mixers as close to the same information as possible, human mixers had the advantage of knowing which type of instrument it was. Therefore, they assigned priority according to this prior known knowledge. For this reason a similar priority scheme was chosen to compensate for this. Both A and B algorithms used the same priority schema. Mixes used during the test contain music freely available under creative commons copyright can be located in (Perez_Gonzalez and Reiss 2010).

As shown in Figure 58, the test used two questions to measure the perceived overall quality of the panning for each audio comparison. For the first question, ‘how different is the panning of A compared to B?’, a continuous slider with extremities marked ‘exactly the same’ and ‘completely different’ was used. The answer obtained in this question was used as a weighting factor in order to decide the validity of the next question. The second question, ‘which file, A or B,

has better panning?', used a continuous slider with extremes marked 'A quality is ideal' and, 'B quality is ideal'. For both of these questions, no visible scale was added in order not to influence their decision.

The test subjects were also provided with a comment box that was used for them to justify their answers to the previous two questions. During the test it was observed that expert subjects tend to use the name of the instrument to influence their panning decisions. In other words they would look for the "bass" label to make sure they kept it center. This was an encouraging sign that panning amongst professionals follows constraint rules similar to those that were implemented in the algorithms.

The tested population consisted of 20 professional sound mixing engineers, with an average experience of 6 years work in the professional audio market. The tests were performed over headphones, and both the human mixers and the test subjects used exactly the same headphones. The test lasted an average time of 82 minutes.

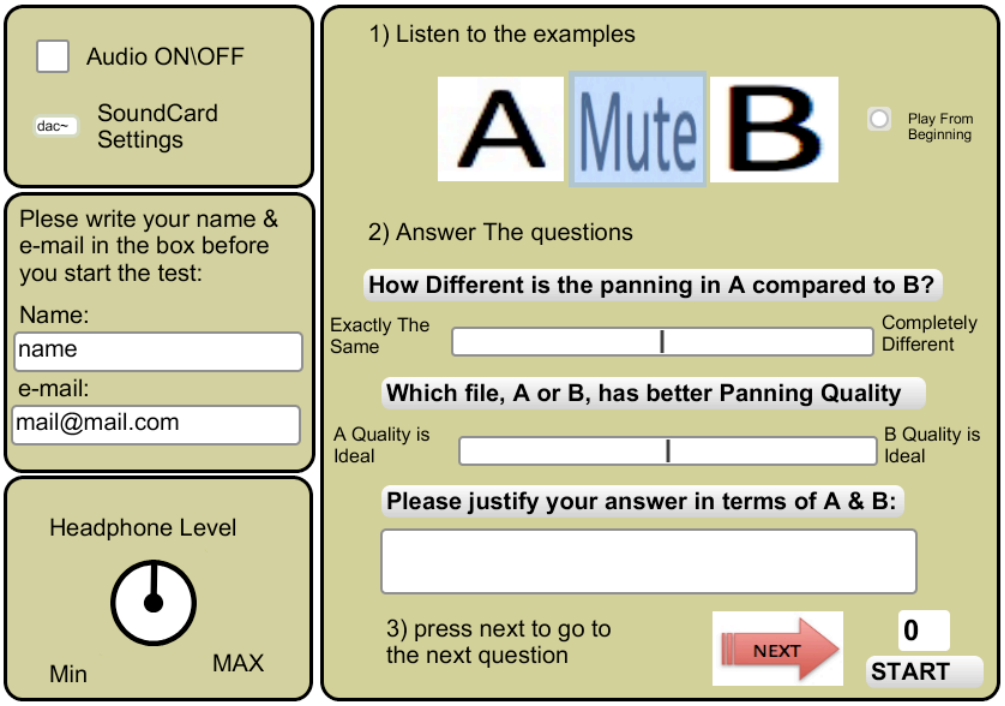

Figure 58 Double blind panning quality evaluation test interface.

Double blind A/B testing was used with all possible permutations of algorithm A, algorithm B, expert and amateur panning. Each tested user answered a total of 32 questions, two of which were control questions, in order to test the subject's ability to identify stereo panning. The first control question consisted of asking the test subjects to rate their preference between a stereo and a monaural signal. During the initial briefing it was stressed to the test subject that stereo is not necessarily better than monaural audio. The second control question compared two stereo signals that had been panned in exactly the same manner.

8.4.3 Result analysis

All resulting permutations were classified into the following categories: monaural versus stereo, same stereo versus same stereo file, method A versus method B, method A versus non-expert mix, method B versus non-expert mix, method A versus expert mix, and method B versus expert mix.

Results obtained on the question 'How different is panning A compared to B?' were used to weight the results obtained for the second question 'Which file, A or B, has better panning quality?'. This is in order to have a form of neglecting incoherent answers such as "I find no difference between files A or B but I find the quality of B to be better compared to A".

Answers to the first question showed that, with at least 95% confidence, the test subjects strongly preferred stereo to monaural mixes. The second question also confirmed with at least 95% confidence that professional audio engineers find no significant difference when asked to compare two identical stereo tracks. The results are summarized in Table 1, and the evaluation results with 95% confidence intervals are depicted in Figure 59.

The remaining tests compared the two panning techniques against each other and against expert and non-expert mixes. The tested audio engineers preferred the expert mixes to panning method A, but this result could only be

given with 80% confidence. On average, non-expert mixes also were preferred to panning method A, but this result could not be considered significant, even with 80% confidence.

In contrast, panning method B was preferred over non-expert mixes with over 90% confidence. With at least 95% confidence, we can also state that method B was preferred over method A. Yet when method B is compared against expert mixes, there is no significant difference.

The preference for panning method B implies that low-pass spectral decomposition is preferred over band-pass spectral decomposition as a means of signal classification for the purpose of automatic panning. Furthermore, the lack of any statistical difference between panning method B and expert mixes, (in contrast to the significant preference for method B over non-expert mixes, and for expert mixes over method A), leads us to conclude that the automatic panning method B performs roughly equivalently to an expert mixing engineer.

Test	Number of Comparisons	Preference	Confidence	Standard Deviation	Mean
Stereo vs. Mono	20	Stereo	95%	0.61545	-0.4561
Stereo vs. Stereo	20	Identify them to be the same	95%	0.0287	0.0064
Human Expert vs. Method A	144	Human	80%	0.5105	-0.666
Method A vs Non-expert	56	No significant difference between algorithms	95%	0.5956	-0.0552
Method B vs Non-expert	56	Method B	90%	0.6631	0.1583
Method B vs Expert	144	No significant difference between algorithms	95%	0.5131	0.0108
Method A vs Method B	200	Method B	95%	0.4474	-0.0962

Table 1 Double blind panning quality evaluation table.

It was found that the band-pass filter bank, method A, tended to assign input channels to less filters than the low-pass filter bank, method B. The distribution of input tracks among filters for an 8-channel song for both methods is depicted in Figure 60. In effect, panning method B is more discriminating as to whether two inputs have overlapping spectral content, and hence is less likely to unnecessarily place sources far from each other. This may account for the preference of panning method B over panning method A.

The subjects justified their answers in accordance with the common practices mentioned previously. They relied heavily on manual instrument recognition to determine the appropriate position of each channel. It was also found that any violation of common practice, such as panning the lead vocals, would result in a significantly low measure of panning quality. One of the most interesting findings was that spatial balance seemed to be not only a significant cue used to determine panning quality, but was also a distinguishing factor between expert and non-expert mixes. Non-expert mixes were often weighted to one side, whereas almost universally, expert mixes had the average source position in the centre. Both panning methods A and B were devised to perform optimal left to right balancing. Histograms of source positions that demonstrate these behaviors are depicted in Figure 61.

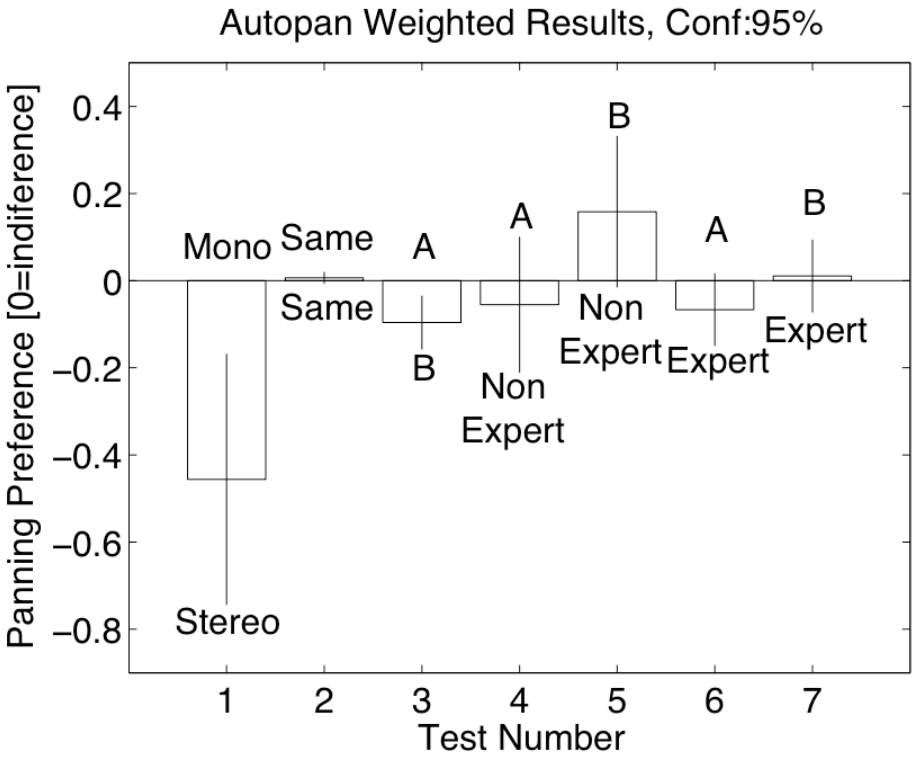

Figure 59 Summarized results for the subjective evaluation. The first two tests consisted of reference tests (comparing stereo against monaural, and comparing identical files). The remaining questions compared the two proposed auto-panning methods against each other and against expert and non-expert mixes. 95% confidence intervals were used.

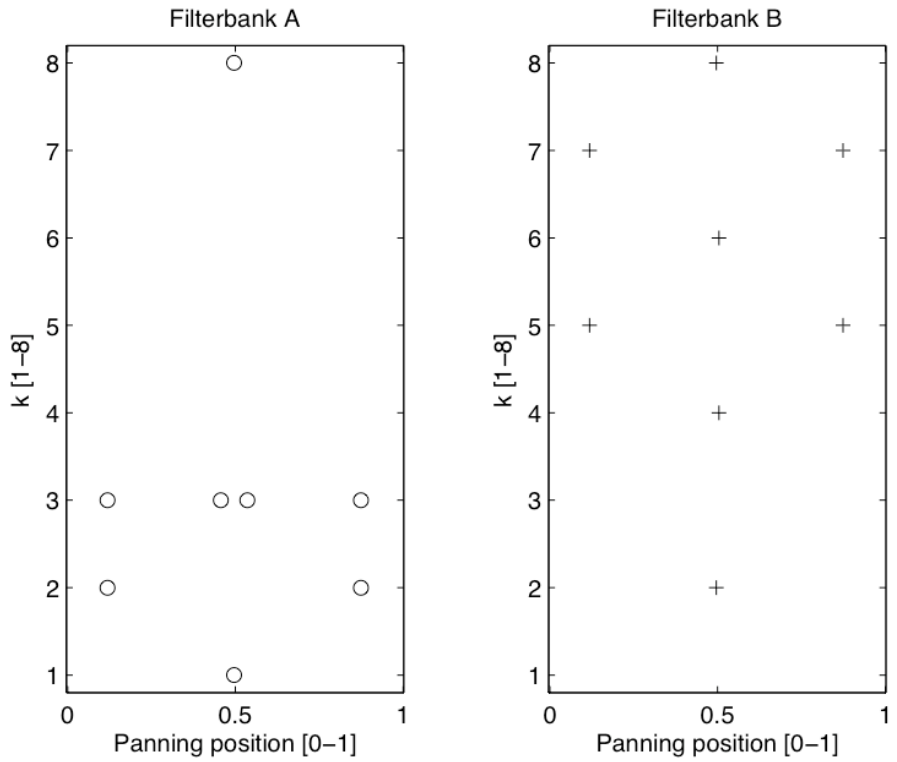

Figure 60 Distribution of sources among filters for methods A and B for the same song (*W=0.059*)

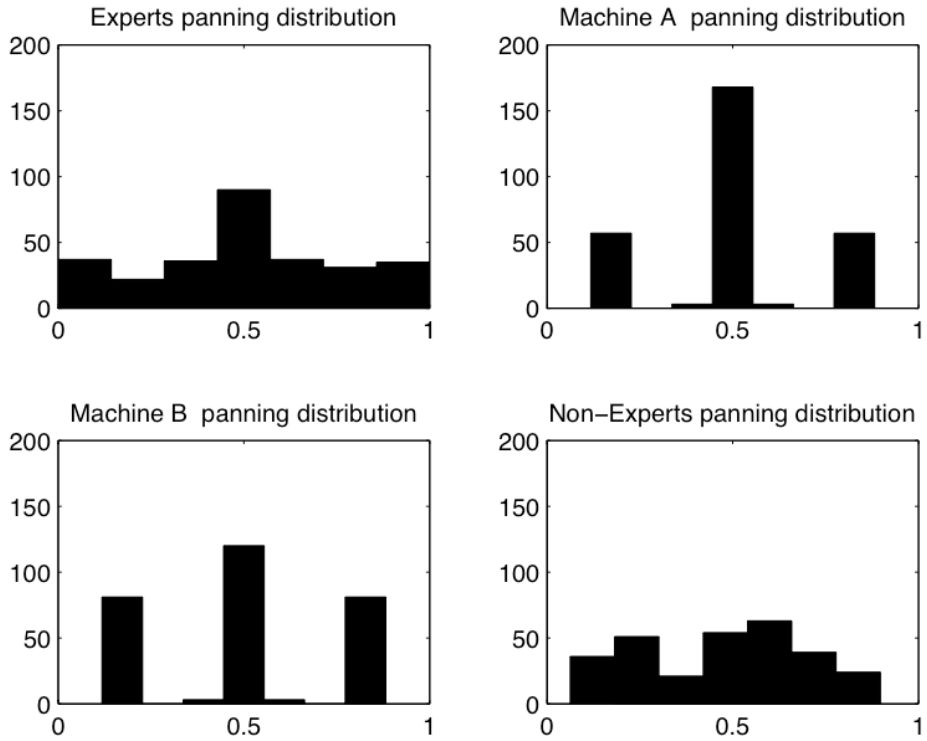

Figure 61 Panning space distribution histograms (*W*=0.059).

8.5 Summary

In terms of generating blind stereo panning up-mixes with minimum human interactions, we can conclude that it is possible to generate intelligent expert systems capable of performing better than a non-expert human while having no statistical difference when compared to a human expert. According to the subjective evaluation, low-pass filter-bank accumulative spectral decomposition features seem to perform significantly better than band-pass decompositions. This is due to its ability to sparse more evenly the spectral classification across the panning space. It was found that power balance between the left and right channel is an important constrains expected from an automatic panner.

More sophisticated forms of performing source priority identification in an unaided manner need to be investigated. To further automate the panning technique, instrument identification and other feature extraction techniques could be employed to identify those channels with high priority. Furthermore, in live sound situations, the sound engineer would have visual cues to aid in panning. For instance, the relative positions of the instruments on stage are often used to map sound sources in the stereo field. Video analysis techniques could be used to incorporate this into the panning constraints.

Chapter 9

Automatic accumulative fader method

9.1 Introduction

A cross-adaptive mixing device has been developed for the purpose of optimizing the gain levels of a live audio mix. The method aims to achieve optimal mixing levels by optimizing the ratios between the loudness of each individual input channel and the overall loudness contained in a stereo mix. In order to evaluate loudness of each channel in real-time, accumulative statistical measurements were performed. The system uses a cross-adaptive algorithm to map the loudness indicators to the channel gain values. It has applications in automatic mixing of live music, live mixing of game audio, and studio recording post-production.

9.2 Automatic fader

In order to create a balanced audio mix, careful scaling of input gains and level faders must be achieved. Several methods for automatically setting levels for speech have been proposed, (Dugan 1975; Peters 1978; Julstrom and Tichy 1984; Dugan 1989). On the other hand, only a few methods for automatically setting the levels for music have been proposed (Dugan 1975; Campbell and Whittemore 1982; Dannenberg 2007). In the case of the methods proposed by Dugan, Campbell and Dannemberg, the systems are based on measuring signal amplitude and adapting the mix according to low-level feature amplitude

indicators. The use of perceptual attributes was suggested by (Dannenberg 2007), but was not implemented. (Dugan 1975) describes a method which works by turning on and off microphones when their input level is greater than an adaptive threshold, while (Campbell and Whittemore 1982) attempt to achieve mix balance by lowering high amplitude signals and increasing the level of low amplitude signals.

In this chapter, we approach the problem by making use of cross-adaptive methods driven by a perceptual indicator. The proposed system attempts to handle the task of weighting the gain between channels by using accumulative loudness measures. We assume that a mix in which loudness per channel tends to the overall average loudness is a well-balanced mix with optimal inter-channel intelligibility. By doing this, each channel has an equal chance of masking other channels, thus optimising the likelihood of each channel being heard. The system adapts its gain according to the relationship of loudness indicators between channels and the overall average loudness of the mix. In order to achieve this we apply the following criteria:

1. Equal loudness probability: By scaling all input signals such that they tend to a common average probability, minimal perceptual masking can be achieved.

2. Minimum gain changes: The algorithm should minimise gain level changes required in order to avoid excessive gains by using the overall average loudness of the mix as a reference.

3. Fader limit control: There must be a mechanism for limiting (the amount of maximum) gain applied to the input signals. This avoids unnaturally high gain values from being introduced.

4. Maintain system stability: The overall contribution of the control gains $cv_m(n)$ should not introduce distortion or acoustic feedback artefacts.

In this chapter, the theory and implementation behind such a system will be presented together with results demonstrating the effectiveness of the technique.

9.3 Research and implementation

9.3.1 Loudness estimation

In the same way that the method introduced in (Dugan 1975) mentions that the implementation is dependent on the accuracy of the implementation of the amplitude envelope feature, our implementation depends on the overall performance of the loudness feature. Loudness is a perceptual attribute of sound and therefore requires a psycho-acoustic model. Our psychoacoustic model implementation has been depicted in Figure 62.

The loudness feature in this thesis utilizes the ISO 226 standard loudness curves (ISO 2003). The curves are used to weight the amplitude of the sampled input signal $x_m(n)$. A loudness weighting curve, $w(SP(n))$, was stored for every 10dB$_{SPL}$ increment, within a range of 10 dB$_{SPL}$ to 130dB$_{SPL}$, where $SP(n)$ states the increment range of the weighting curve. Four biquadratic filters in series were used to approximate the loudness curves. The coefficients for each biquadratic filter corresponding to each loudness curve were stored in a lookup table. The lookup table outputs the appropriate coefficients according to the reading of a sound pressure level meter device, or can be manually fixed to a desired value for non-live applications. The input loudness per channel is calculated and reported to the system by using equation 44

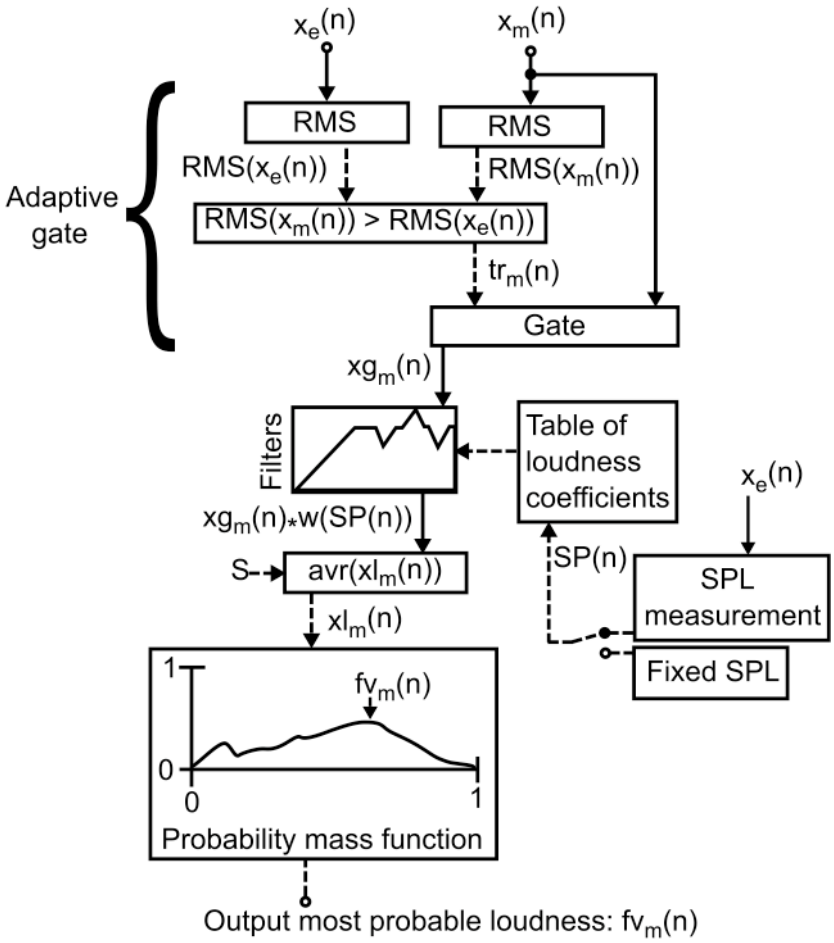

Figure 62 Loudness feature block diagram.

$$xl_m(n) = \frac{\sum_{i=1}^{S} (xg_m(n) * w(SP(n)))_i}{S}, \qquad (44)$$

where *SP(n)* corresponds to the measured sound pressure level, and *S* represents a given number of samples for calculating the mean amplitude loudness, $xl_m(n)$, given that the frame is given by *i.* Our current implementation has average buffer size of *S=200* samples, but this is user selectable. The weighting calculation is performed per channel by using the *SP(n)* value derived from the external microphone $x_e(n)$, thus all channels are weighted with the same loudness curve. The system uses adaptive gating in order to ensure more reliable measurements.

9.3.2 Adaptive gating

In practice, a noisy input distorts the loudness measurement. For this reason, a gate with an adaptive threshold was implemented. Consider the inputs being microphones on a stage. There is a usable distance, where the microphone performs well. If the performer is too far away from the microphone, the signal to noise ratio will be too low and unsuitable for reproduction. In (Dugan 1975), a method of installing a measurement microphone outside the usable distance of the reproduction microphones was proposed as a measurement for noise. Therefore, a microphone far away from stage, and representative of the overall mix, can be used as a noise model from the perspective of each individual channel, while being used for obtaining the sound pressure reference *SP(n)*. The system will only let a signal through if $x_m(n)$ is ≥ $x_e(n)$. This gated signal represents a cleaner, more representative measurement of the loudness of $x_m(n)$, and will be noted as $xg_m(n)$. In the current implementation the adaptive gating is performed on the data stream as opposed to the signal flow. This offers the advantage that the gate performs at a slower rate than the audio stream, thus reducing the overall processing power required in inverse proportion to *S*. When the gate is closed, $xg_m(n)$ is in a state of no signal, which is different from a state of silence. This is important given that a correct loudness measurement can now be achieved which is not biased by silence.

9.3.3 Accumulating the loudness

Once we have a clean measurement of $xl_m(n)$, we can proceed to the analysis. The implementation proposed in this chapter uses accumulative histograms of loudness. Each channel histogram represents the loudness mass probability function, thereby representing its probabilistic behavior from the start of the measurement up to the time of the current measurement. Since the system is to be used in real time, for computing a histogram we must consider the range of the loudness signal. This is done to ensure that the maximum number that can be held by the histogram function is equal to the maximum value taken by the

function $xl_m(n)$. This is analogous to ensuring the level of a head-preamplifier avoids clipping. Given that the system is capable of having a maximum amplitude input of one, the actual maximum measurement of such a signal can generate a loudness measure not linearly related to the maximum amplitude input magnitude. For this reason, a self-adjusting scaling mechanism was implemented to ensure that the values of $xl_m(n)$ were within the range of 0 to 1. The normalization algorithm scans for a probability higher than zero in its highest bin, $B_{max}(n)$. In case this is true for any of the channels, then the rescaling gain of all channels, $r(n)$, should be decreased by a factor d. The process should be repeated recursively until the highest bin in the histogram is equal to zero for all channels. All channels must use the same gain scalar in order to have a common reference, so the gain of the channel with highest input level is used. Such a system is depicted in Figure 63.

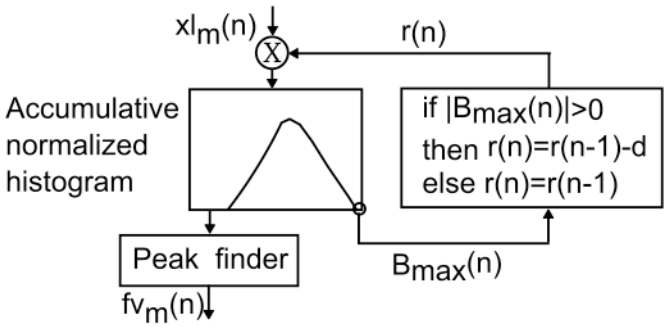

Figure 63 Histogram adaptive rescaling.

The current implementation has a decrement value $d=0.5$ and a rescaling initial gain of $r(0)=100$. These values have been determined experimentally, although the system is robust to parameter changes. Once the accumulative histograms have been correctly rescaled and gated, we can proceed to calculate the highest peak for each channel and use it as the most probable loudness state, $fv_m(n)$.

9.3.4 Cross-adaptive function

The cross-adaptive function consists of mapping the perceptual loudness of each channel to its amplitude level so that, by manipulating its amplitude, we can achieve the desired loudness level. Given that we are aiming to achieve an average loudness value *L(n)*, we must increase the loudness of the channels below this average and decrease the channels above this average. The average loudness *L(n)* is obtained as the arithmetic mean of $fv_m(n)$ for all channels. In turn, our aim is to find a factor $cva_m(n)$, representing a channel fader gain control level, such that we achieve a common average loudness between channels, *L(n)*. A model which approximates the problem is depicted in Figure 64.

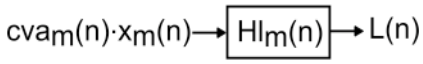

Figure 64 Loudness feature system diagram.

The input output ratio function is given by $Hl_m(n) = L(n)/[cva_m(n)x_m(n)]$, where *L(n)* is the arithmetic average loudness of the system obtained by averaging all most probable accumulated loudness values $fv_m(n)$ from *m=0 to M-1*. Then we can derive the following equation for determining $cva_m(n)$:

$$cva_m(n) = \frac{L(n)}{Hl_m(n)x_m(n)}, \qquad (45)$$

where $cva_m(n)$ represents the fader gain control factor per channel in order to achieve *L(n)*. On the other hand we know that for every channel the input output ratio function of the feature extraction system depicted in Figure 62 is given by $Hl_m(n)=fv_m(n)/x_m(n)$. So $Hl_m(n)x_m(n)=fv_m(n)$, where $fv_m(n)$ corresponds to the most probable loudness state per channel. Therefore, the fader gain control factor per channel is given by $cva_m(n)=L(n)/fv_m(n)$.

This proposed model has the advantage that it is not dependent on the feature used, thus leaving room for future study of better features without the need for a major re-implementation of the system.

9.3.5 Determining the fader headroom of the system

In most cases $cva_m(n)$ represents a physical fader with range limits. The system must ensure that the values of $cva_m(n)$ are within range. The proposed solution is to scale the input to the side chain, $x_m(n)$, before it is measured. This scaling is proportional to the available headroom that the system will have between $L(n)$ and the maximum value that can be taken by $cva_m(n)$. For example, scaling by $0.5x_m(n)$ will give a 6dB headroom to the mix with respect to $L(n)$. This rescaling is currently user selectable, and must be selected according to the type of desired dynamics of the music being mixed. If a channel requires compensation which forces $cva_m(n)$ to go out of range, it should be clamped to its highest possible value. In practice, such a clamping action should indicate to the user the need for compressing this particular signal or re-selecting the microphone position in order to achieve the desired headroom. In order to apply this headroom scaling factor and obtain the corresponding scaled fader control factor $cvr_m(n)$, equation 46, must be updated to give

$$cvr_m(n) = \frac{L(n)}{Hl_m(n)x_m(n)hr}, \qquad (46)$$

where hr correspond to the available headroom fader level and $x_m(n)$ has been scaled by a factor $1/hr$ before the loudness model is applied. The final user interface is depicted in Figure 65; the red markers indicate the need for compression on channels 2 and 3.

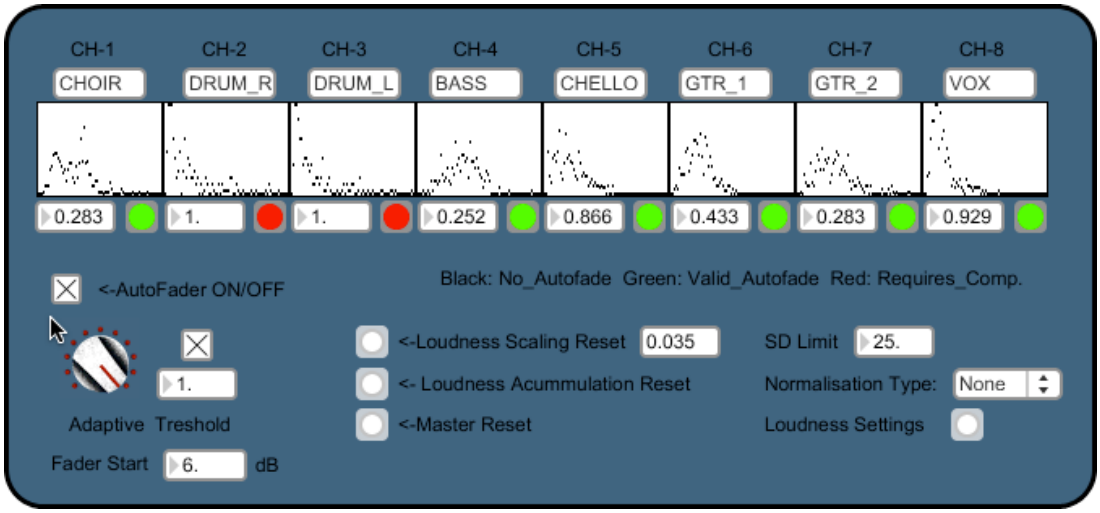

Figure 65 User interface for automatic accumulative fader mixing tool.

9.3.6 Keeping overall system stability

The electronic components of the output of a mixer have amplitude limits before distortion, and the overall mix and individual signals have a maximum amplitude limit of one. In this case, regardless of the values taken by $cvr_m(n)$, the maximum gain before feedback will be maintained (Perez_Gonzalez and Reiss 2008). For implementing such a design we must continuously normalize the gain values to add up to unity gain. Such a method, which we refer as cross normalization, has been suggested by (Dugan 1975). It is important to mention that suitable interpolation methods are required in order to implement this in real time applications. This will ensure continuous gain level changes with no audible artefacts. The equation for achieving such normalization is

$$cv_m(n) = \frac{cvr_m(n)}{\sum_{m=0}^{M-1} cvr_m(n)}, \qquad (47)$$

where $cv_m(n)$ is the normalized version of $cvr_m(n)$. Thus the summation of all $cv_m(n)$ from 0 to $M-1$ is equal to one and the final mix is given by

$$y_{mix}(n) = \sum_{m=0}^{M-1} cv_m(n) x_m(n) \quad (48)$$

,

where $cv_m(n)$ has been obtained by targeting a common loudness average $L(n)$ between all channels in the mix, while doing a cross normalization. An overall system diagram is depicted next in Figure 66.

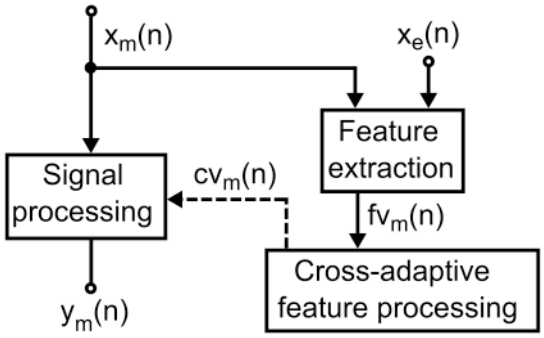

Figure 66 Overall system diagram. Solid line audio path, dotted line data control path.

9.4 Test and results

Using a musical signal, a measurement of the convergence between the channel loudness $fv_m(n)$ and the overall mix average loudness $L(n)$ was performed. This is depicted in Figure 67. The musical signal was of a choir part of a song, It can be seen that the choir starts at *t=10000ms*. A measurement of the average peak loudness before and after the signal had been psychoacousticaly weighted was performed. The resulting channel loudness is expressed by $x_m(n)cv_m(n)$. A user assigned headroom target, *hr=0.5,* was used.

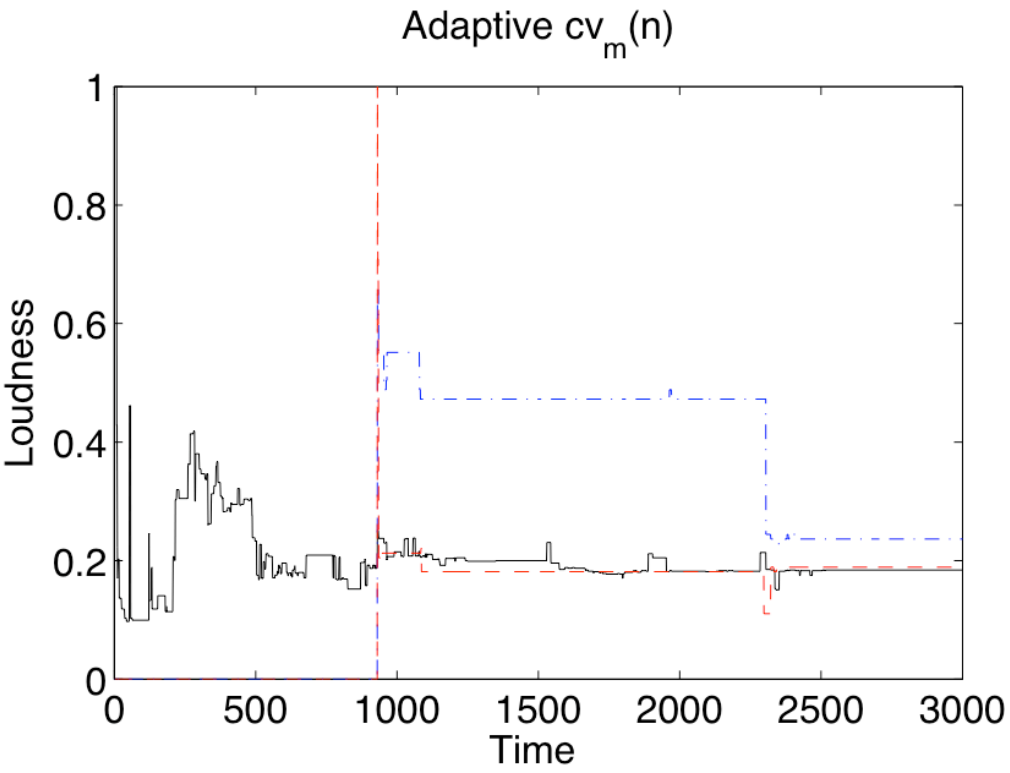

Figure 67 Cross-adaptive target loudness for a single music channel before and after applying the automatic fader algorithm before interpolation. [Time in units of 10ms].

All, three measurements have been depicted on in Figure 67. First, the overall mix average loudness, *L(n),* is depicted using a black solid line (-). Second, the control vector, without loudness compensation, has been plotted in a blue dashed/dotted line (-.-). Finally, the loudness of $x_m(n)cv_m(n)$ is depicted with a red dashed line (--). It can be seen that on average, $cv_m(n)$ is able to follow the magnitude of *L(n)*. Results indicate that the accumulative characteristics of the measurement make it robust to noisy changes. In cases where a sudden change in $cv_m(n)$ is needed, this could prove troublesome.

9.5 Summary

An implementation of a cross-adaptive effect has been developed for the purpose of automatically optimizing the gain levels of an audio mix. A mixing model in which the perpetual loudness per channel tends to the overall average of the mix in order to achieve a well-balanced mixture with optimal intelligibility has been proposed. The system can be used as an indicator of what channel signals need to be compressed based on the user headroom limitations imposed by the user. Tests performed with music signals indicate that system is able to match perceptual average target loudness successfully. In order to improve the performance of the system, better psycho-acoustic models for loudness, such as (Skovenborg 2008), could be implemented.

Chapter 10
Automatic equalizer

10.1 Introduction

Equalizing a sound mix is one of the most complex tasks in live music mixing requiring human expertise. The main problem of determining the amount of equalization to be used is that the perceived equalization is different from the applied equalization. In order to achieve a perceptually pleasant equalization several things should be considered; whether or not the channel needs equalization at all, how many filters should be used, the type of filters and ultimately the amount of boost or cut they should have. Some studies on how the sound engineer performs these decisions have been made by (Bitzer and et al. 2008; Bitzer and LeBeuf 2009).

Automatic mixing of speech and music levels has been attempted by (Dugan 1975; Campbell and Whittemore 1982; Dannenberg 2007; Perez_Gonzalez and Reiss 2009). However, very little has been done to attempt self-equalization of musical signals. The only notable example of research in automatic equalization is (Reed 2000). Here, an off-line machine learning approach was used. In Reed's approach humans need to manually train the machine. Once it is trained, it equalizes using nearest neighbor techniques.

In this chapter a proposed method for use in live mixing situations driven by perceptual indicators will be researched. The proposed system does not require off-line machine learning. Instead it uses a real time cross-adaptive accumulative spectral decomposition approach to the problem based on a multiband implementation of chapter 9. The cross-adaptive algorithm uses the relationship between the perceptual loudness of all input channels to perform the equalization of every individual channel. The system then handles the task of weighting the channel equalization bands by using a perceptual indicator corresponding to a set of spectrally decomposed accumulation measurements of loudness. The spectral decomposition of signals is achieved by the use of a flat response filer bank. In this method we assume that the mix in which loudness per band tends to the overall average loudness of the signal is a well-equalized mix with optimal inter-channel equalization intelligibility. The idea behind this is to achieve an equal chance of masking between channels, thus optimizing the likelihood of each channel being heard. In order to achieve optimization, the system adapts its sub-band equalization gains according to the relationship of loudness indicators between channels and the overall average loudness. In this chapter the theory and implementation behind such a system, and results demonstrating the functionality of the system is presented.

10.2 Automatic equalizer

The proposed system consists of two fundamental parts. The first is the signal processing part of the algorithm consisting of an equalizer. In the context of the presented algorithm, the equalizer under study has fixed frequency bands and the only parameters are the gains of each frequency band. The processing part takes the input signal channels $x_m(n)$ where m correspond to the channel number from $m=0$ to $M-1$, and outputs an equalized version of each channel known as $y_m(n)$. The second part is the cross-analysis, which takes $x_m(n)$ as an input and outputs the control gain parameters corresponding to the equalization bands of each channel, $cv_{km}(n)$, where $k=0$ to $K-1$ is the equalization

band, and K is the maximum number of bands. The system diagram of the overall system is presented in Figure 68. In the following sections we explain the theory and implementation of all the analyses stages required to derive the equalization parameters $cv_{km}(n)$.

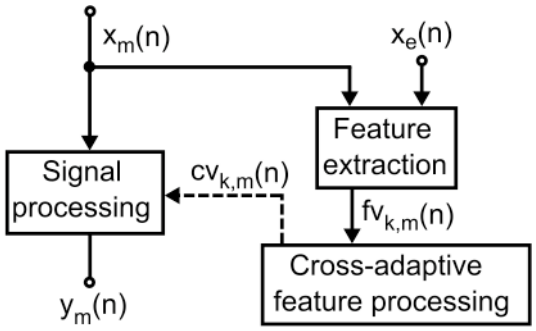

Figure 68 System overview block diagram. Signal flow through the signal processing has no added latency due to the side chain processing.

The implementation of such an automatic equalization tool should comply with the following design requirements:

1. Equal loudness probability per band: All the input signals involved which share spectral content, should tend to the same average loudness per band.

2. Minimum gain changes: The algorithm should perform the most optimal gain changes, therefore it should increment or decrement band gain from a natural starting point such as the overall average of the mix.

3. Overall equal loudness probability: The system must simultaneously achieve equal loudness per band and full bandwidth equal loudness as proposed in chapter 9.

10.3 Research and implementation

10.3.1 Spectral decomposition

The first step of the cross-analysis is the decomposition of each of the channel inputs, $x_m(n)$, into K frequency bands. For each $k=0$ to $K-1$ decomposition band there must be a corresponding equalizer band in the processing side of the algorithm. For this $x_m(n)$ must be processed by a filter bank, a fast Fourier transform, FFT, or a similar transform such as a constant-Q transform (Brown 1992), in order to separate the input into K spectral bands. For the system to perform properly the spectral bands (or bins) must be spread as evenly as possible, and must add up to a flat frequency response. The accuracy of the final performance of the system will be dependent on the number of spectral decomposition bands; the more spectral bands used, the more accurate the system will be.

10.3.2 Adaptive gating for multiband implementations

For ensuring the loudness model uses a clean and valid signal, the use of adaptive gating is recommended at the input of our feature extraction step. In the proposed method the adaptive gating is a multiband implementation based on the adaptive gating approach of (Dugan 1975; Dugan 1989). Therefore, the system requires an external input $x_e(n)$ to derive the adaptive threshold. A spectral decomposition filter bank whose filters $h_{k,m}$ match the cut-off frequencies of the equalizer filters $hq_{k,m}$ is used in the signal processing equalization section. For performing the multi-band adaptive gating, the system takes each of the spectrally decomposed bands of $x_m(n)$ and $x_e(n)$ and outputs a clean version of each of the bands of $x_m(n)$ denoted as $hg_{k,m}(n)$. By determining the relationship between the ambient noise microphone and the individual inputs, it is possible to determine if a signal contains noise or not. Such functionality is given by the following pseudo-code 1.

```
if  [RMS(h_{k,m}(n))>RMS(he_k(n))]
     tr_{k,m}(n) = open_gate;
else
     tr_{k,m}(n) = close_gate;
```

Pseudo-code 1 Multiband adaptive gate implementation

Where $tr_{k,m}(n)$ is the adaptive threshold signal for operating the noise gate. $h_{k,m}(n)$ corresponds to the input signal of the spectrally decomposed input channels, and $he_k(n)$ corresponds to the signal of the spectrally decomposed ambient microphone input. Therefore the signal, $h_{k,m}(n)$, can be gated by the adaptive threshold signal $tr_{k,m}(n)$ in order to obtain a noise-free signal which can be used to extract a valid feature in correspondence to the perceived loudness.

10.3.3 Loudness weighting

For extracting the loudness feature a model containing the ISO 226 standard loudness curves (ISO 2003) is used. Given that a psychoacoustic model is used for weighting the signals, the more accurate the psychoacoustic model of loudness the better the results the system should give. The proposed model, depicted in Figure 69 consists of a look-up table containing all the coefficients necessary for generating the loudness weighting curves, $w(SP(n))$. The table is driven by Sound Pressure Level (SPL) measurement denoted as $SP(n)$. The ISO curves are defined in steps of 10dB. The SPL value of $SP(n)$ can be time varying, as obtained from a single SPL measurement microphone at the mixing position, or can be non-time varying by fixing the value of $SP(n)$ for all values of n. Given that the spectrally decomposed input, $h_{k,m}(n)$ is weighted by $w(SP(n))$ we can perform an averaging of length S in order to include in the model longer and shorter term loudness measures. S is given in samples, and therefore is dependent on the sample. Our psychoacoustic weighting is given by equation 49

$$xl_{k,m}(n) = \frac{\sum_{i=1}^{S} (xg_{k,m}(n) * w(SP(n)))_i}{S}, \quad (49)$$

where $xl_{k,m}(n)$ represents the psychoacousticaly weighed signal derived from the spectrally decomposed and gated signals $xg_{k,m}(n)$.

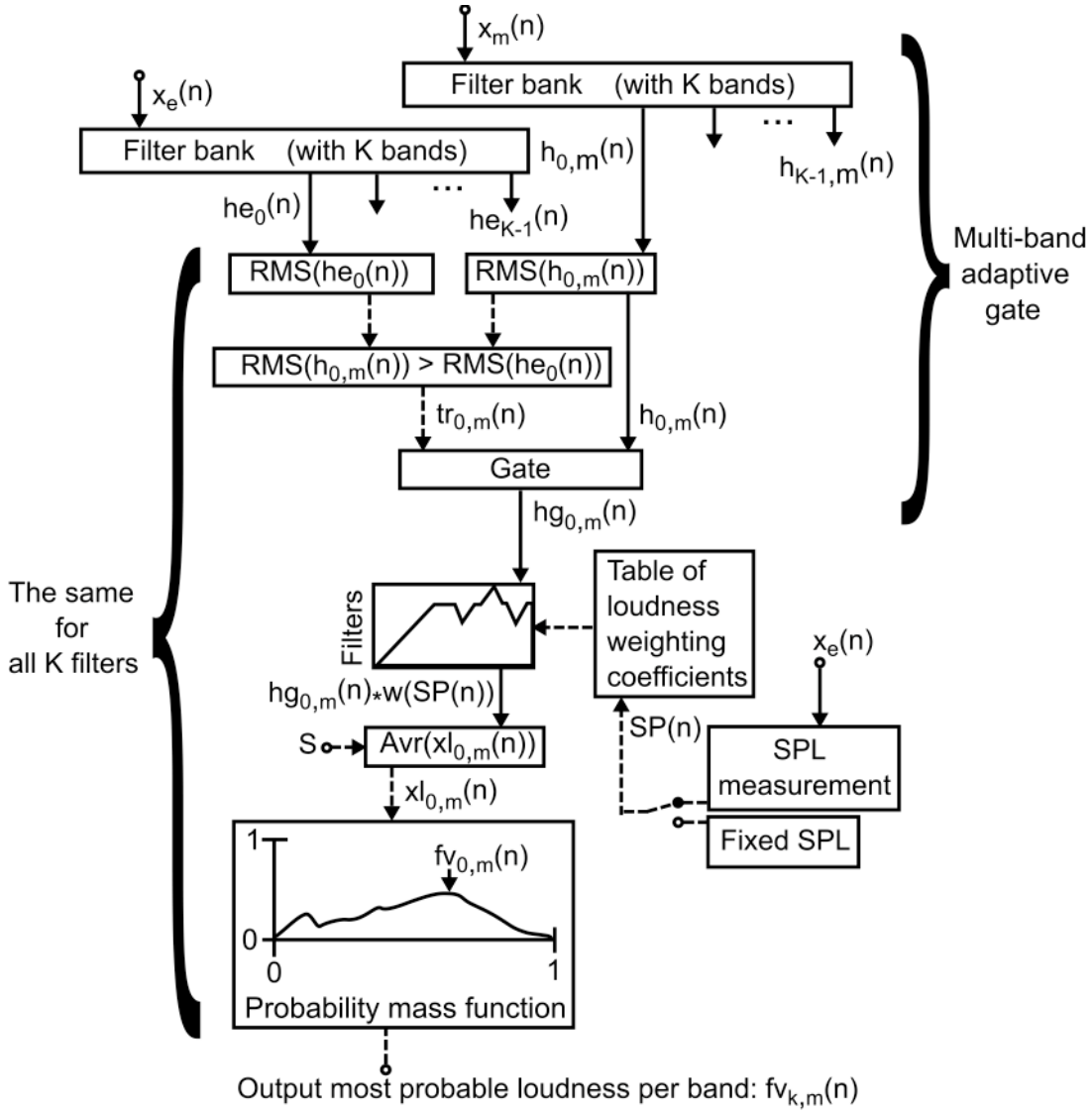

Figure 69 Loudness feature weighting diagram.

10.3.4 Peak loudness accumulation

Once we are sure that we have a noise-free representation of the perceived loudness, $xl_{k,m}(n)$, we can proceed to determine its accumulated peak loudness. The proposed method for obtaining a value representative of the spectral band is to accumulate its normalized histogram in order to determine the probability mass function of the analyzed loudness band. From this probability mass function we can then determine the most probable loudness value for a given spectral band. Given the on line use of the algorithm, it is necessary to ensure that the histogram variance is kept within range. This is due to the fact that knowing the maximum peak value of $x_m(n)$ does not ensure that the limits of the histogram values will be the same, since the peak magnitude of $x_m(n)$ is not the same as the weighted $xl_{k,m}(n)$ peak value after loudness weighting. For this reason a cross-rescaling mechanism was implemented. The system works by rescaling $xl_{k,m}(n)$ by a factor $r(n)$. The overall system gain reference, $r(n)$, is given by finding the maximum gain value that can satisfy all $xl_{km}(n)r(n)$ such that its maximum peak value is equal to one.

This multichannel cross-scaling function is accomplished by the following pseudo-code 2

```
if [|Bmax_{k,m}(n)|>0]
      rs_{k,m}(n)=rs_{k,m}(n-1)-d;
else
      rs_{k,m}(n)=rs_{k,m}(n-1);
```

Pseudo-code 2 multichannel cross-scaling function for probability mass function accumulation.

where $Bmax_{k,m}(n)$corresponds to the value taken by the highest bin of the histogram and $rs_{k,m}(n)$ corresponds to the maximum channel gain such that $max(xl_{k,m}(n)\ rs_{k,m}(n))<=1$ so that the channel has a $Bmax_{k,m}(n)=0$ and d is a fixed decrement step. $r(n)$ is given by equation 50

$$r(n) = \arg\min(rs_{k,m}(n)), \qquad (50)$$

where *r(n)* is simply the minimum value over all $rs_{k,m}(n)$. The scaling is now within range. We can proceed to look for the probability mass function peak, $cv_{k,m}(n)$ which should correspond to the most probable loudness value of a given channel band. The flow diagram of such a histogram rescaling system is depicted in Figure 70.

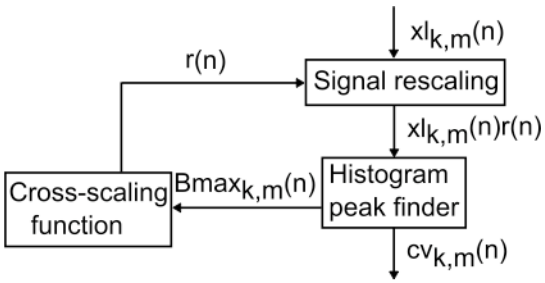

Figure 70 Peak loudness accumulation diagram.

10.3.5 Cross-adaptive function

The final signal processing of the equalized channel signals, $xeq_m(n)$ for a set of channel inputs, $x_m(n)$, has the following function prototype

$$xeq_m(n) = EQ[x_m(n), cv_{k,m}(n)], \qquad (51)$$

where $cv_{k,m}(n)$ corresponds to the equalizer band gain coefficients. Then, in order to achieve a continuous variation of $cv_{k,m}(n)$, so that a common average loudness between all channels and their corresponding equalization bands is maintained, the system is modeled with the system diagram in Figure 71.

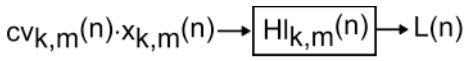

Figure 71 Loudness feature diagram.

Figure 71 is a multiband extension of the cross-adaptive model presented in chapter 9. From its input output ratio function, the corresponding equation for determining the equalizer band gain coefficients $cv_{k,m}(n)$, can be derived using equation 52.

$$cv_{k,m}(n) = \frac{L(n)}{Hl_{k,m}(n)x_{k,m}(n)} \qquad (52)$$

In equation 52 the numerator $L(n)$ is the average loudness of all channel equalization bands. $L(n)$ is given by the following equation:

$$L(n) = \sum_{m=0}^{M-1}\left(\sum_{k=0}^{K-1} fv_{k,m}(n) / K\right) / M \qquad (53)$$

Where $l(n)$ is the average of all $fv_{k,m}(n)$ for all k bands and m channels. Given $fv_{k,m}(n)$ corresponds to the most probable loudness value for each spectral band, M corresponds to the maximum number of channels involved in the mix and K corresponds to the number of bands in the spectral decomposition.

Then we can say that $Hl_{k,m}(n)$ is the input output ratio function of the combined system, given by $Hl_{k,m}(n)= fv_{k,m}(n)/x_{k,m}(n)$, such that the control loudness value per channel spectral band is $fv_{k,m}(n)=Hl_{k,m}(n)x_{k,m}(n)$, where $fv_{k,m}(n)$ corresponds to the most probable loudness state per spectral band, and is therefore the denominator in equation 52, therefore $cv_{k,m}(n) = L(n)/f v_{k,m}(n)$.

Finally, the auto-equalized mix is given by

$$y_{mix}(n) = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} cv_{k,m}(n)[hg_{k,m}(n) * x_m(n)] \qquad (54)$$

.

In order to maintain system stability, a method that will maintain maximum gain before feedback has been implemented. This system prevents transfer function gains above unity in order to maintain system stability (Perez_Gonzalez and Reiss 2008). The final user interface for the automatic equalization tool is presented in Figure 72.

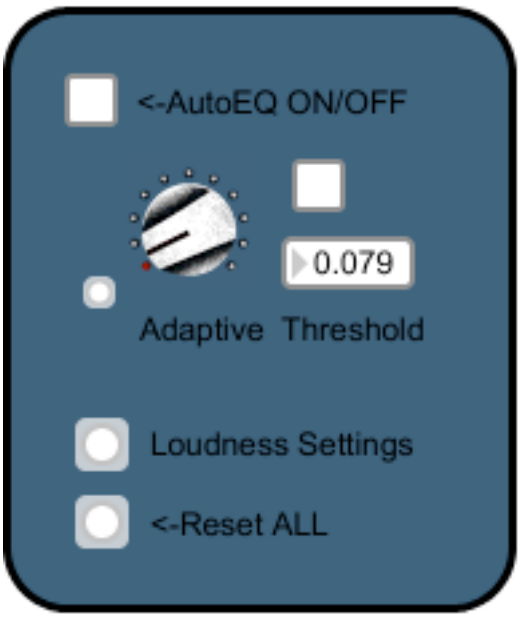

Figure 72 Automatic equalization tool user interface.

10.3.6 Decomposition filter bank and matching equalizer

A first order, 5 element filter bank with flat frequency response was implemented, in order to test the proposed system. The spectral decomposition filter bank consists of the following Butterworth designs: a low-pass filter with a cut of frequency of 63Hz, three band-pass filters with mid band frequencies at

127Hz, 750Hz and 4000Hz and a high-pass filter with a cut off frequency of 8000Hz. Such implementation has been depicted in the top section of Figure 73. Its corresponding equalizer design makes use of the same filter topology, as shown in the bottom plot of Figure 73 with K=5. Individual filter response has been plotted as a solid line, while combined frequency response at unity gain position has been plotted as a dashed line.

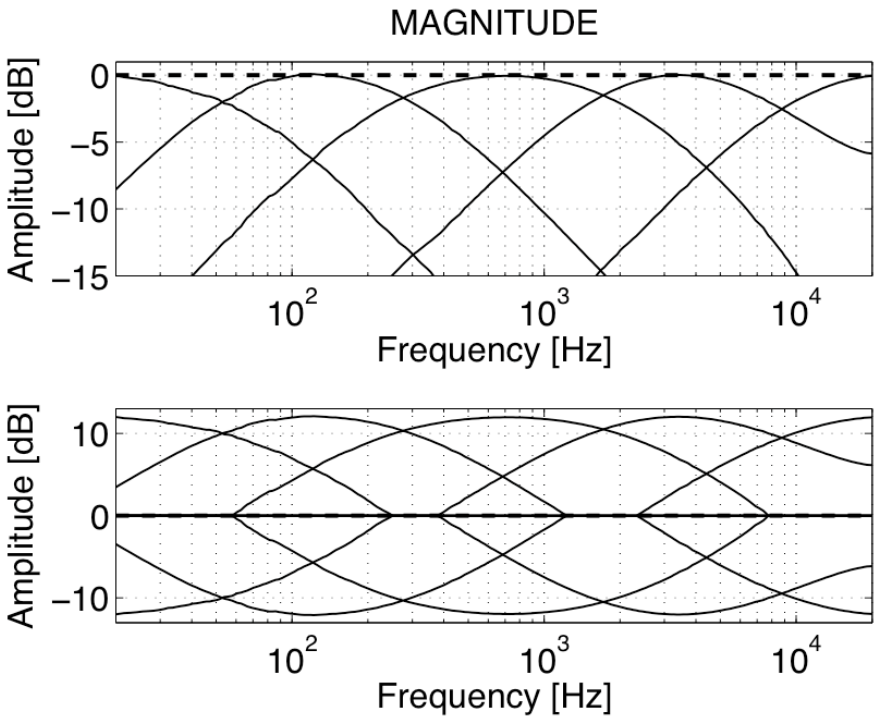

Figure 73 Filter bank transfer function (top) and matching equalizer transfer function (bottom).

10.4 Test and results

A set of eight channel live multi-track recordings with different styles of music was used for testing the system. A single omni-directional flat frequency response microphone was used to capture the ambient noise and calculate the SPL. Spectral changes occurred after applying the algorithm. The first impressions are positive. However we did not carry out thorough subjective evaluation. The system performs better in the high frequency range than in the

low end. Spectrum comparisons indicated a tendency to increment the high frequency range. An increase of dynamic range on the auto-equalized signal was encountered in all signals tested. For all the signals tested an increase of 3dB crest factor was observed. Crest factor is calculated from the peak amplitude of the waveform divided by the RMS value of the waveform. This can clearly be seen in Figure 74, where the un-equalized time domain signal seems to present more cluttering, while the auto-equalized signal has more defined transients. The recording depicted in Figure 74 had an increase of crest factor of approximately 3.2 dB.

The resulting equalized transfer functions for an auto-equalized multi-track recording consisting of Ch1= vocals, Ch2= guitar, Ch3= synthesizer-left, Ch4= synthesizer-right, Ch5= Snare, Ch6= kick-drum Ch7= high-hat and Ch8= over-head, were plotted in Figure 75. In all audio samples tested, it was clear that the algorithm tends to improve the high frequency section of the spectrum but has a tendency to under boost low frequencies.

10.4.1 Test signals

To examine system characteristics, a test signal measurement consisting of white noise was input to the system. If the system performs as expected, it should match the inverted loudness weighing curve applied to the signal, $1/w(SP(n))$. We found that for high SPL levels the system performs as expected, see top plot of Figure 76 where the equalizer transfer function gives a good match to $1/w(120dB)$. On the other hand, it fails to match the low frequency spectrum section when below $1/w(90dB)$, as shown in the bottom plot of Figure 76. We found that below 90dB the selection of having a LPF with a cut-off frequency of 63Hz caused the analysis to have no low frequency signal available in that lower sub band. This means that to be able to approximate the white noise signal to a loudness curve of $1/ w(SP(n))$ for $SP(n)<=90dB$ it is necessary to have an analysis filter bank with more than five filters or a higher cut-off up point for the LPF.

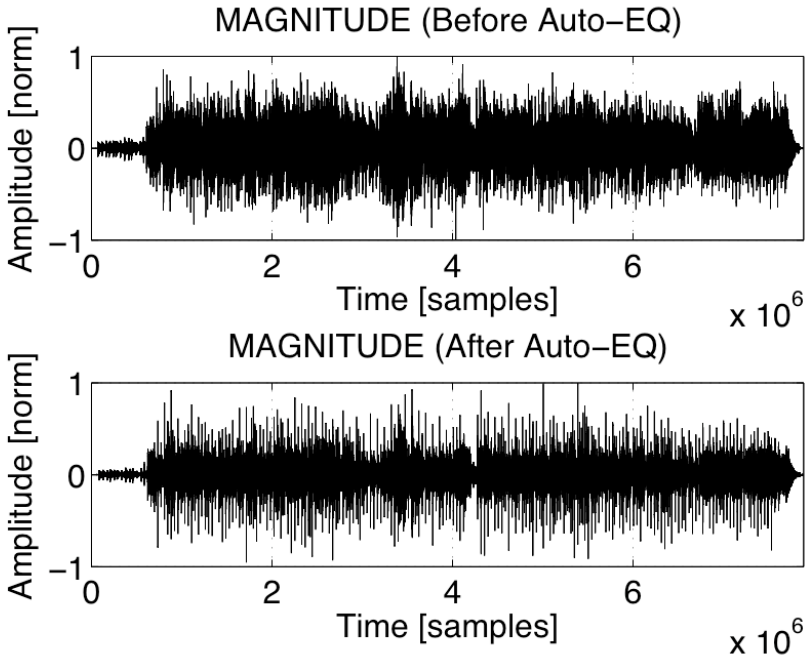

Figure 74 Time domain self equalization of a music signal.

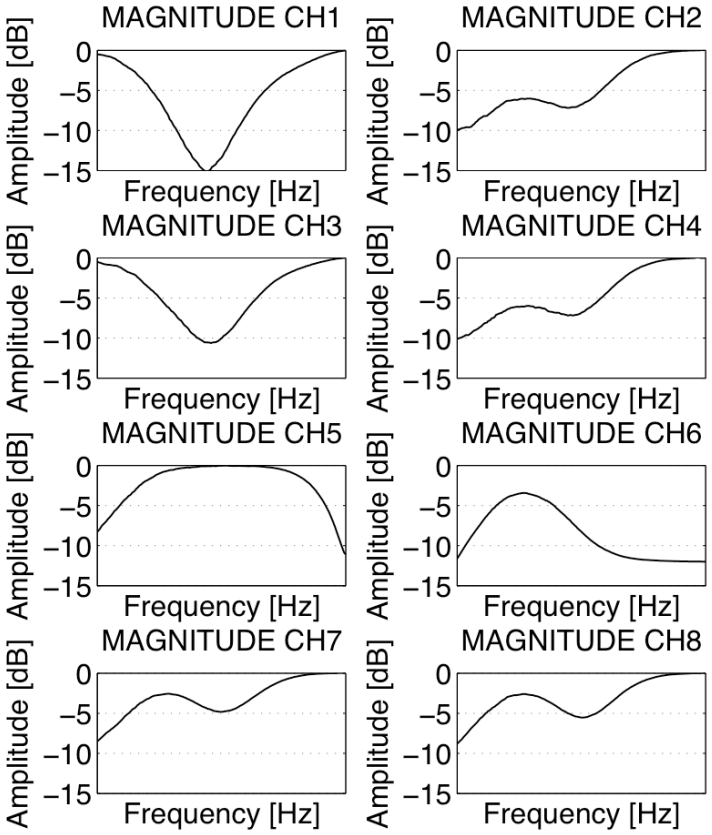

Figure 75 self-equalization of a music signal.

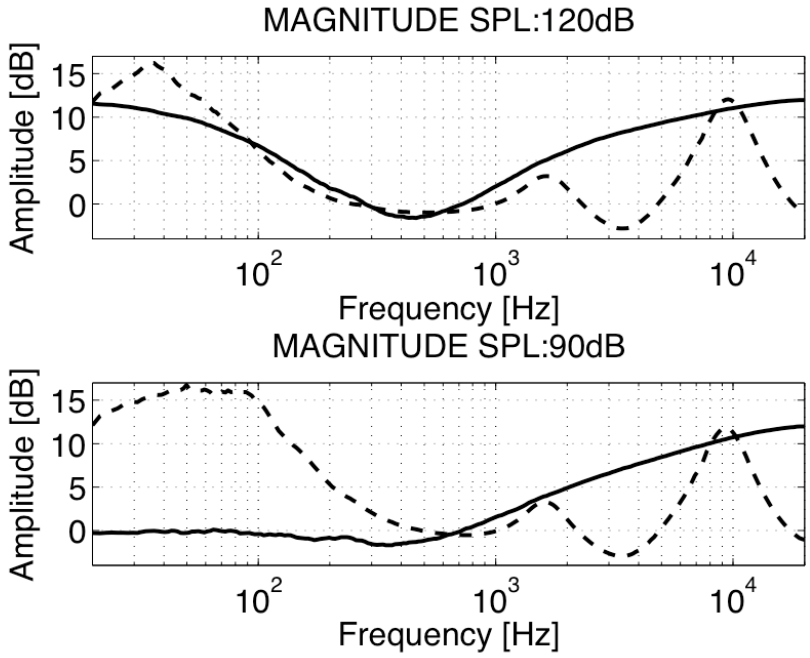

Figure 76 Self-equalization of a white noise test signal, solid line. Top, auto-equalized response for *j(n)=120dB*. Bottom, auto-equalized response for *j(n)=90dB*. Bottom. 1/ *w(j(n))* is represented by the dashed line.

Results indicate that a better implementation either with more filters in the filter bank or a Fourier approach will greatly improve performance the low frequency implementation.

10.5 Summary

The theory and implementation of a cross-adaptive system capable of using perceptual weighing in order to achieve equal probabilistic psychoacoustic weighing of the equalizer bands has been presented. Current implementations indicate that the system has potential uses in live equalization for music. The system is limited to fixed band gain adjustments. Results produced using a fixed five-band spectral decomposition indicate that, in some cases, there is not enough filter flexibility to achieve an ideal equalization curve.

Part III Conclusions and future work

Chapter 11
Conclusions

We will now summarise the outcomes of this thesis and will suggest directions of future research. This includes possible improvements to the automatic mixing tools presented herein and the possibility of researching unexplored areas in the field of automatic mixing. Finally, a brief closing statement reflecting the author's thoughts has been included.

11.1 Conclusions

Overall it was shown that by using feature validation, cross-adaptive architecture, adaptively gated accumulative feature extraction and cross-adaptive mapping, it is possible to generate automatic mixing tools that provide static mixes that satisfy a set of technical constraints. In cases close to live mixing conditions the system, and in particular the automatic panning tool, behaved in similar ways to a human mixer.

In chapter 2 we introduced the current state of automatic mixing and found that automatic mixing for live music, as opposed to speech only applications, is under-developed. In chapter 3 we introduced a framework and building blocks of the automatic mixing tools presented in this thesis. The framework consists of a cross-adaptive structure and is able to reproduce complex actions performed by a human mixing engineer, who will take decisions not only based on the content of the channel to be processed but also on the signal content relationships between other channels. A side chain

processing structure was presented which is capable of taking multiple audio signals as inputs, and outputting a series of control data values which relate to the signal processing parameters of the mixer. The side chain processing consists of feature extraction and cross-adaptive processing sections. The need to validate the inputs to the side chain processing section was exposed and methods to ensure noise robustness were introduced with special emphasis on adaptive noise gating. Accumulative methods based on probability mass function analysis were introduced as a way to achieve a statistically robust feature for every channel. Once a set of features was obtained for each channel, their inter-channel relationships were evaluated using a cross-adaptive function. The resulting outputs are the control variables that control the signal processing parameters of the audio mixer. Finally, the need for an automatic system stability building block was introduced.

From chapter 4 onwards a series of automatic mixing tools were introduced. In chapter 4 a normalization technique based on transfer function analysis was introduced as a means of maintaining stability of the system regardless of the changes to the control parameters by either human interaction or automatic mixing actions. In chapter 5 a method for accurately setting the head amplifier input of the mixer while maximizing dynamic range and reducing distortion was introduced. In chapter 6 a polarity and offset corrector capable of automatically identifying polarity errors between channels was introduced. The system was shown to be robust to noise and reverberation. In chapter 7 a spectral enhancer was introduced. This enhancer is capable of performing gain changes on all of the involved channels in a mix depending on their spectral interrelations. The idea of having a common cross-adaptive function among different channels is the basis of all automatic mixing tools presented in this thesis. In chapter 8 an automatic panner was introduced. The automatic panner makes extensive use of all the building blocks introduced in chapter 3. An exhaustive subjective test showed that when using band pass spectral decomposition the system performs in a similar manner to that of an expert human mixer. In chapter 9 an accumulative automatic fader tool was

introduced. This method differs from the previous method as it used psychoacoustic features and accumulation of the feature data to determine a static mix. Finally, in chapter 10 a multiband extension to the automatic fader approach was presented as a way to create an automatic equalization tool. Although successful this automatic equalizer was limited as it had a fixed filter in which gain was the only variable. All tools presented were implemented and tested in real time applications and to a different degree they all show capabilities for automatic live mixing of music.

In this thesis we have introduced a class of digital audio effects that can do automatic mixing. As well as demonstrating a number of examples, a framework has been presented into which future automatic mixing tools, can be placed. The system architecture utilises cross-adaptive processing of features extracted from the input signals. Depending on the speed of the feature extraction mechanisms, the automatic mixing has been classed as either dynamic or accumulative. Optimizing the accuracy of the feature extraction mechanism can significantly improve the performance of the automatic mixing.

11.2 Future directions

A compendium of relevant future directions on automatic mixing will be presented. Automatic mixing is a new field of research therefore there are numerous directions the research could take. First, a set of directions for the tools presented in this thesis will be outlined. Second, a list of unexplored research opportunities into automatic mixing will be outlined.

11.2.1 Automatic mixing tools improvements

Automatic gain normalization Implementing a fixed point per octave transform will correct the errors of the algorithm at low frequencies. The use of all-pass filters for compensation of inter–channel phase problems can be an interesting direction of study. The use of acoustic external inputs to determine loop gain could result in more efficient implementation.

Automatic head amplifier gain The system currently decreases gain until it reaches an optimal level. Unfortunately the system cannot differentiate between noise and a musical input, therefore the system could benefit from signal characterization and some statistical analysis in order to avoid erroneous settings due to momentary noise.

Automatic polarity and time offset correction The implementation of the current algorithm with recursive decimation could greatly improve the error rate of the algorithm due to its ability to calculate low frequencies correctly. This would also extend the usable length of delay detection and correction without sacrificing accuracy. Implementing a coherence measure could be used for validating data in order to improve reliability. Improving reliability for percussive time offset correction is also needed. Currently the method works for a single source with multiple microphones therefore extending it to multiple sources and multiple microphones would be a very interesting direction of research, for this, the development of new mathematical techniques together with expanding current delay estimation methods will be needed.

Automatic Spectral enhancer The use of a weighting psychoacoustic model before the signal arrives to the spectral decomposition classifier could improve the final result of the spectral enhancer. The improvement of the quantitative spectral masking metrics can benefit from the further development of this tool. Its phase implementation for improving directional masking could be a very interesting direction of research.

Automatic panning The priority rule is currently the only part of the automatic panner which requires aid from the user. It is thought by the author that priority plays an important role in the aesthetic result of the mixture. For this reason the inclusion of a better priority scheme or even a restricted rule scheme based on some type of source recognition / instrument identification could be used to improve the performance of the tool. No current source or instrument recognition known by the author is currently accurate enough for this purpose.

Improving the quantization of the panning space towards a more smooth, continuous panning space could make the system perform closer to that of a human mixer. Extending the system to be able to perform beyond stereo would be an interesting direction of improvement.

Automatic accumulative fader method The system relies on infinite statistical accumulation therefore this makes the system too stiff for reacting to unexpected temporal variations. Implementing the system with a time-forgetting weighting algorithm could improve this. This tool would also benefit from better loudness models. As the method stands it provides a good starting mix but it would need further research for it to be capable of delivering a fine tuned final mix. This would probably involve taking into account some aesthetic and subjective considerations.

Automatic equalizer This is one of the most challenging parts of automatic mixing. Improving this tool should start by widening our understanding of the nature of equalization procedures performed by an audio engineer. Some preliminary work on understanding how audio engineers equalize has been done in (Bitzer and et al. 2008; Bitzer and LeBeuf 2009). The current system is limited to a gain only system; a system that is capable of adjusting filter centre frequencies and Q is yet to be developed. All-pass filter automatic equalisation is also a fascinating direction of research.

11.2.2 Automatic mixing tools unexplored directions

Cross-adaptive feedback effects Currently all the tools presented here make no use of cross-adaptive feedback topologies. The side chain process could perform more accurately if it has access to extracting features from the channel inputs as well as from the processed outputs. This could be used for more complex cross-adaptive processing mapping and for error optimization. Therefore the use of such topologies could be beneficial to the performance of the automatic mixing tools. Such implementations remain a source of future research.

Dynamic and spatial effects In this thesis no automatic dynamic or automatic spatial tools, such as reverberators and compressors, were researched. The study of automatically determining the parameters of a dynamic and spatial effect remains one of the ultimate challenges of automatic mixing. This is especially because their use and objectives are varied and in many cases based on subjective decisions. Some current research on automatic noise gating has been performed by (Terrell and Reiss 2009).

Target mixing The idea of making one mix sound like another is an interesting area of research. Research by (Kolasinski 2008; Barchiesi and Reiss 2009) allows the extraction of several linear parameters of a mixer in order to understand how it was mixed but more research into nonlinear parameter extraction needs to be done. The idea of taking two completely different mixes and trying to impose the sound characteristics of one onto the other remains a very difficult task.

Automatic mixing for small rooms In the current approach we suppose rooms are big enough so that the sound of the sources is decoupled from the

audience. For this reason a system capturing loud sounds such as drums or guitar amplifiers can assume they need to be amplified without the need to take into account if they are already present in the room. This is true of big rooms such as big theatres, arenas and stadiums but such a phenomena is not true of small spaces such as pubs or small audience rooms. Many musicians who perform in these small spaces could benefit from automatic mixing tools, therefore a way of including room constrains could be a challenging way to expand this research. An approach currently under research by (Terrell and Reiss 2009) contemplates introducing to the system the transducer locations and characteristics of a room in order for an optimisation process to be used. Current research in this field is still limited and could be a useful direction of research that would involve a more acoustic point of view of audio mixing.

Cross-adaptive host Currently there is no audio processing host dedicated to the use of multichannel cross-adaptive effects. The development of such a tool would widely accelerate the development of automatic mixing tools. The optimisation of dedicated interface and hardware for this type of tool is still to be fully developed. Current digital mixer designs have trouble coping with the data update rate required for updating the user interface since many control interfaces were not designed to be updated all at once in real time. Systems designed to cope with real time feature extraction and information sharing across tracks are of ultimate importance for automatic mixing. This requires a rethink of the current design of audio mixers.

Environmental and room compensated auto mixing An atmospheric probe capable of inputting temperature and pressure to the system could be included. This would be used to infer a rescaling value of the master fader in order to maintain stability of the system even during environment changing conditions, such as the change in gain before feedback due to a change in temperature.

Use of visual and location tracking A shortcoming of some automatic mixing tools is the lack of information they process of the outside world. For example, in the case of the automatic panner tool, its ability to infer information solely from the audio signals limits its potential. It is known that audio engineers tend to pan sources according to visual feedback such as the location of the musicians on stage. The inclusion of position tracking devices linked to the panning positions of the sources could solve this shortcoming. This improvement would also permit the automatic panner to perform spatialisation beyond stereo sources into other formats like ambisonics, 5.1, or some high order spatial format.

The use of tracking devices for determining the musicians position could also be used for performing rescaling and thus as a form of maintaining system stability and avoiding unwanted acoustic feedback artifacts. Tracking could be performed using Bluetooth or wireless microphones. As both devices use radio frequency transmission, theoretically the location can be estimated using power triangulation.

Complicated mixing scenarios. We contemplated mixing in cases where a base, static mix is required such as live mixing where once the audio engineer reaches a base mix he concentrates on doing small changes during the live performance. On the other hand there is the case where the whole mix can change radically such as the case of carefully crafted studio mixes where it is common for there to be less time constraints for developing multiple parameter automation in mixes. Being able to perform these time varying mixes is an interesting research challenge.

More complicated mixing scenarios such as the ones required for complex scene changing games can pose a degree of complexity for which the system presented in this thesis would require further research. Optimising for

such application would be an interesting challenge. Another complex example is monitor mixing for live and recording where more than one type of mix is required to be delivered simultaneously.

Aesthetic and subjective mixing Being able to take subjective decisions based on aesthetic constraints similar to those that human audio engineers perform would lead to the ultimate automatic mixing tool. Unfortunately we have a large gap in understanding of how the audio engineer decides aesthetic considerations while performing an audio mix. This could ultimately lead to a redesign of current audio mixers where operation is based on low level signal features to mixers with controls that are related to higher level psychoacoustic features.

Improvement of features and feature decision making The development of better psychoacoustic models and understanding of how humans listen and why we perform certain mixing decisions based on perceptual information would greatly benefit the automatic mixing field. The application of real time machine learning techniques such as (Reed 2000), optimisation and real time information retrieval and on-the-fly semantic tagging could benefit the complexity of automatic mixing decisions that can be taken. More advanced methods of feature extraction such as the use of source separation or microphone interference reduction techniques could improve automatic mixing reliability. Therefore this is an interesting route of research especially for cross-field cooperation with the other research communities.

11.3 Final thoughts

When the available processing effects or setup time are limited, the system performed on average in a similar manner as a human mixer. On the other hand, when the available processing effects and setup time is unlimited the system does not perform as well as a human mixer. In an informal test an automatically generated mix was submitted to an Audio Engineering Society recording competition. The judges commented that the mixture was far from what was expected from a professional recording engineer, but they were unable to identify that the mix was performed by a machine until they were told. There seems to be a need for research and expansion into automatic dynamics and automatic special effects in order to achieve better results that fully approximate a human mixer.

The intention of the automatic mixing tools is to aid or replace certain tasks that are normally undertaken by the audio engineer. Only time will tell how autonomous a digital signal processing unit, as contained in a music mixer, will become and to what extent it will be accepted by the user. Although the automatic mixing tools have been implemented for live music mixing applications, several industry sectors have shown interest in such tools. This includes the gaming, recording, postproduction, and the mastering industry.

Automatic mixing is at present a growing and exciting field of research and several commercial devices based on such principles have started to emerge. The use of different configurations and topologies in the implementation of automatic mixing tools remains to be explored. The tools proposed here deal mainly with technical mixing constraints and are meant to be used as a tool that allows the sound engineer to concentrate on more creative aspects of the mix.

In the future, more extensive use of perceptual models may not only improve the performance of automatic mixing but may also allow more subjective mixing decisions to be explored. Research into these subjective areas of mixing is, however, likely to remain controversial. It is the vision of this researcher based on the results obtained on this thesis that audio mixers that are as simple to operate as an automatic digital camera will one day, not t0o far into the future, be a reality.

Part IV Appendices And Bibliography

Appendix A
List of relevant published work

11.4 Published work

A list of published scientific contributions has been given next. They have been divided into academic scientific publications, patents, invited seminars and articles describing this work, which featured in the popular press. Finally, the official URL for the automatic mixing work will be also mentioned.

11.4.1 Scientific publications

E. Perez_Gonzalez and J. Reiss. “An autonomous audio panning system for live music”. EURASIP Journal Advances in Signal Processing, Special Issue on Digital Audio Effects, manuscript accepted 23 April 2010.

E. Perez_Gonzalez and J. Reiss. “Automatic Gain and Fader Control For Live Mixing”. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, New York, 18-21 October, 2009.

E. Perez_Gonzalez and J. Reiss. “Automatic equalization of multi-channel audio using cross-adaptive methods”. 127th Convention Audio Engineering Society, New York, 9-12 October 2009.

E. Perez_Gonzalez and J. Reiss. “Determination and correction of individual channel time offsets for signals involved in an audio mix”. 25th AES Convention Audio Engineering Society, San Francisco, USA, 2-5 October 2008.

E. Perez_Gonzalez and J. Reiss. “Improved control for selective minimization of masking using interchannel dependency effects”. 8th International Conference on Digital Audio Effects (DAFx-08), Espoo, Finland, ISBN 978-951-22-9517-3, pp. 75-81, 1-4 September, 2008.

E. Perez_Gonzalez and J. Reiss. “An automatic gain normalization technique with applications to audio mixing”. 124th Convention Audio Engineering Society, Amsterdam, The Netherlands, 17-20 May 2008.

E. Perez_Gonzalez and J. Reiss. “Automatic mixing: live down mixing stereo panner”. 7th International Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, ISBN 978-88-901479-1-3, pp. 63-68, 10-15 September 2007.

11.4.2 Book chapter

E. Perez_Gonzalez and J. Reiss. “Automatic mixing”. (To appear) DAFX Digital Audio Effects. Second edition, Chapter 12, Edited by Zölzer, U., West Sussex, England, John Wiley & Sons, Ltd.

11.4.3 Patent

E. Perez_Gonzalez and J. Reiss. “Anti-feedback device”. UK patent GB0808646.4 (WO/2009/138754), filed 13 June 2008 and published 19 November 2009.

11.4.4 Invited seminars

Guest speaker for “New concepts In audio: Automatic Mixing” Audio Engineering Society Latin American conference 2008 Mexico City, Mexico, 27-29 April 2008

Guest speaker for “Programa Educativo Audio Engineering Society 2010”. Jornada educativa Audio Engineering Society Latin American 2010, Mexico City, Mexico, 25-27 April 2010

11.4.5 Popular press scientific publications

E. Perez_Gonzalez and J. Reiss, **“**Enter the robot sound desk**”** by Nic Fleming New Scientist Magazine No.2745 United Kingdom, pp.17, 30 January 2010

E. Perez_Gonzalez and J. Reiss, **“**When a squeal and a wail won’t do**”** by Laura Margottini New Scientist Magazine No.2650 United Kingdom, pp.26, 3 April 2008

E. Perez_Gonzalez and J. Reiss, **“**Is this the end for feedback? New software aims to take the buzz and screech out of live music**”** by James Randerson The Guardian Newspaper Front page science section United Kingdom, 3 April 2010

Other relevant press interviews have been part of the content of the BBC Radio 4, BBC World Service, Radio Deutschlandfunk, ITN and LBC's Morning Report, and more recently, ProSound News Supplement Magazine and the AES Europe Convention 2010 Live Podcast on the 25 May 2010. Most of this can be found on the official automatic mixing tool URL detail next.

11.4.6 Official automatic mixing tools website

The following URL contains general automatic mixing tools information, links to scientific papers, popular press reports and demos:

http://www.elec.qmul.ac.uk/digitalmusic/automaticmixing/

The list of demos in automatic mixing tools available in the URL are presented next:

Automatic normalization technique
Polarity offset correction
Spectral enhancer
Automatic panner
Audio examples from the automatic panner evaluation
Accumulative automatic gain and fader adjustments
Automatic equalization
Song 1 no auto -Equalization
Song 1 auto –Equalization
Song 2 no auto -Equalization
Song 2 auto –Equalization
Combined Mixing tools
With dynamic faders
With auto-Eq

Bibliography

Advendano, C. and Jot, J. M. (2004). "A Frequency-Domain Approach to Multichannel Upmix." Journal of the Audio Engineering Society **52**(7/8): 740-749.

Amatrian, X. and et al. (2003). "Content-based transformations." Journal of New Music Research **32**(1): 95-114.

Anderson, J. L. (2008). "Classic Stereo Imaging Transforms—A Review." Retrieved 10 February, from http://www.dxarts.washington.edu/courses/567/08WIN/JL_Anderson_Stereo.pdf.

Antman, H. S. (1965). "Extension to the theory of howlback in reverberant rooms." Journal of the Acoustical Society of America **39**(2): 399.

Ballow, M. G., et al. (2002). Handbook for Sound Engineers. Oxford, UK, Focal Press / Elsavier.

Barchiesi, D. and Reiss, J. (2009). "Automatic target mixing using least-squares optimization of gains and equalization settings." 12th Int. Conference on Digital Audio Effects (DAFx-09), Como, Italy.

Bartlett, B. (2009). Recorder-mixers and Mixing Consoles. Practical Recording Techniques. Oxford, U.K., Focal Press / Elsevier**:** 259-275.

Beament, J. (2001). The Direction-Finding System. How We Hear Music: The Relationship Between Music and the Hearing Mechanism. Suffolk, UK, The Boydel Press**:** 127-130.

Benjamin, E. (2006). "An Experimental Verification of Localization in Two-Channel Stereo." 121st Convention of the Audio Engineering Society, San Francisco, California, USA.

Bitzer, J. and et al. (2008). "Evaluating perception of salient frequencies: Do mixing engineers hear the same thing?". 124th Convention of the Audio Engineering Society, Amsterdam, The Netherlands.

Bitzer, J. and LeBeuf, J. (2009). "Automatic detection of salient frequencies.". 126th Convention of the Audio Engineering Society.

Brandstein, M. S. and Silverman, H. F. (1997). "A Robust Method for Speech Signal Time-Delay Estimation in Reverberant Rooms." IEEE International Conference on Acoustics, Speech and Signal Processing, Munich.

Brixen, E. B. (2007). "Audio Levels and Readings." Addition to User's Manual of The Master Stereo Displays from DK-Audio. from http://www.dk-technologies.com/downloads/Audio%20Levels.pdf.

Brown, J. C. (1992). "Calculation of a constant Q spectral transform." J. Acoustic Society of America **89**(1): 425-434.

Cable, R. C. and Hilliard, J. K. (1980). "The Practical Applications of Time-Delay Spectrometry in the Field." Journal of the Audio Engineering Society **28**(5): 302-209.

Campbell, E. and Whittemore, R. T. (1982). "Automatic microphone mixing apparatus ". European Patent Office, US 4357492 (A)

Dacht, P. (2008). "Understanding Acoustic Feedback." from www.artcontemporain.lu/larsen/larsen.htm.

Dannenberg, R. B. (2007). "An Intelligent Multi-Track Audio Editor." Proceedings of the 2007 International Computer Music Conference, San Francisco, The International Computer Music Association.

Davis, D. and Patronis Jr., E. (2006). Sound System Engineering. Oxford, UK, Focal Press.

Dugan, D. (1975). "Automatic Microphone Mixing." 51st Convention of the Audio Engineering Society, Los Angeles.

Dugan, D. (1989). "Application of Automatic Mixing Techniques to Audio Consoles." 87th Convention of the Audio Engineering Society, New York.

Gerzon, M. A. (1992). "Signal Processing for Simulating Realistic Stereo Images." 93rd Convention of the Audio Engineering Society, San Francisco, USA, Audio Engineering Society.

Griesinger, D. (2002). "Stereo and Surround Panning in Practice." 112th Convention of the Audio Engineering Society, Munich, Germany.

ISO (2003). Acoustics - Normal equal-loudness-level contours, ISO 226:2003(E) Geneva, Switzerland, Internaiton Organization for Standardization. **226**.

Izhaki, R. (2007). Mixing domains and objectives - Panning. Mixing audio : concepts, practices and tools. Burlington, USA, Focal Press/Elsevier**:** 58-71 and 184-203.

Johnson, N. L., et al. (1993). Univariate Discrete Distributions, page 36.

Julstrom, S. and Tichy, T. (1984). "Direction-Sensitive Gating: A New Approach to Automatic Mixing." Journal of the Audio Engineering Society **32**(7/8): 490-506.

Julstrom, S. and Tichy, T. (1987). Microphone actuation control systems suitable for teleconference systems. USA, Shure Brothers, Inc. **4658425**.

Kamerling, S. and et al. (1998). "A New Way of Acoustic Feedback Suppression." 104th Convention of the Audio Engineering Society, Amsterdam, The Netherlands.

Knapp, C. and Carter, G. (1976). "The generalized correlation method for estimation of time delay." IEEE Transactions on Acoustic, Speech and Signal Processing **24**(4): 320-327.

Kolasinski, B. (2008). "A Framework for Automatic Mixing Using Timbral Similarity Measures and Genetic Optimization." 124th Convention of the Audio Engineering Society, Amsterdam, The Netherlands.

Li, Y. and Driessen, P. F. (2005). "An Unsupervised Adaptive Filtering Approach of 2-To-5 Channel Upmix." 119th Convention of the Audio Engineering Society, New York.

Matthes, M. (2003) Freeverb~ Schroeder / Moorer reverb model. **1.1**,

McCarthy, B. (2007). Sound Systems: Design and Optimization: Modern Techniques and Tools for Sound System Design and Alignment. Oxford, UK, Focal Press.

Meyer, J. (1984). "Equalization Using Voice and Music as the Sources." 76th Convention of the Audio Engineering Society, New York.

Meyer, J. (1992). "Precision Transfer Function Measurments Using Program Material as the Excitation Signal." 11th International Audio Engineering Society Conference: Test & Measurement, Portland, Oregan.

Mongomery, P. (2007). Pseudostereo Techniques. G. P. i. A. a. Acoustics. Sydney, Australia, Faculty of Architecture, Design and Planning, Uneversity of Sydney,.

Neiman, R. (2002). Panning for Gold: Tutorials. Electronic Musician Magazine. USA, Electronic Musician.

Owsinski, B. (2006). Element Two: Panorama - Placing the Sound in the Soundfield. The Mixing Engineer's Handbook. Vallejo,Ca.,USA, Mix Books**:** 20-24.

Pachet, F. and Delerue, O. (2000). "On-the-Fly Multi-Track Mixing." 109th Convention of the Audio Engineering Society, Los Angeles, California, USA.

Painter, T. and Spanias, A. (2000). "Perceptual Coding of Digital Audio." Proceedings of the IEEE **88**(4): 449 - 450.

Perez_Gonzalez, E. and Reiss, J. (2007). "Automatic mixing: live downmixing stereo panner." 7th International Conference on Digital Audio Effects (DAFx-07), Bordeaux-France.

Perez_Gonzalez, E. and Reiss, J. (2008). "An automatic gain normalisation technique with applications to audio mixing." 124th Convention of the Audio Engineering Society, Amsterdam, The Netherlands.

Perez_Gonzalez, E. and Reiss, J. (2009). "Automatic Gain and Fader Control For Live Mixing." IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York.

Perez_Gonzalez, E. and Reiss, J. (2010). "Automatic mixing tools for audio and music production." from http://www.elec.qmul.ac.uk/digitalmusic/automaticmixing/.

Peters, R. W. (1978). Priority mixer control USA, Industrial Research Products, Inc. . **4149032**.

Reed, D. (2000). "A perceptual assistant to do sound equalization." 5th International Conference on Intelligent user interfaces, New Orleans, Louisiana, USA.

Rombouts, G., et al. (2006). "Acoustic feedback suppression for long acoustic paths using a nonstationary source model." IEEE Transactions on Signal Processing **54**(9): 3426 - 3434.

Rumsey, F. and McCormick, T. (2006). Mixers. Sound and Recording: An Introduction. Oxford, UK, Focal Press / Elsevier**:** 96-153.

Schick, B., et al. (2005). "First investigations on the use of manually and automatically generated stereo downmixes for spatial audio coding." 118th Convention of the Audio Engineering Society, Barcelona, Spain.

Schroeder, M. R. (1996). "The “Schroeder freqency” revisited." J. of the Acoustical Society of America **99**(5): 3240–3241.

Self, D. and et al. (2009). Audio Engineering: Know It All. Audio Engineering: Know It All. D. Self. Oxford, U.K., Newnes/Elsevier. **1:** 761-807.

Sethares, W. A. and et al. (2009). "Spectral Tools for Dynamic Tonality and Audio Morphing." Computer Music Journal **33**(2): 71-84.

Shure Brothers Inc. (1978). Data Sheet Models M625 and M625AM: Voice Gate. Shure Brothers Inc.

Shure Brothers Inc. (2000). Data Sheet Models FP410: Portable Automatic Mixer. Shure Brothers Inc.

Shure Brothers Inc. (2007) Microphone Techniques, Live Sound Reinforcement. **AL1266H**,

Skovenborg, E., and Lund T. (2008). "Loudness Descriptors to Characterize Programs and Music Tracks." AES.

Snyder, R. H. (1953). "History and development of stereophonic sound secording." Journal of the Audio Engineering Society **1**(2): 176-179.

SSL (2008). Duende Users Guide, 82S6MC060A.**:** Pages 9.

Terrell, M. and Reiss, J. (2009). "Automatic Monitor Mixing for Live Musical Performance." Journal of the Audio Engineering Society **57**(11): 927-936.

Terrell, M. and Reiss, J. (2009). "Automatic Noise Gate Settings for Multitrack Drum Recordings." 12th Int. Conference on Digital Audio Effects (DAFx-09), Como,
Italy.

Troxel, D. (2005). Understanding Acoustic Feedback & Suppressors. RaneNote 158. Mukilteo, WA, Rane Corporation,.

Tsingos, N. (2005). "Scalable Perceptual Mixing and Filtering of Audio Signals Using an Augmented Spectral Representation." 8th International Conference on Digital Audio Effects (DAFx' 05), Madrid, Spain.

Verfaille, V., and et al. (2006). "Adaptive Digital Audio Effects (A-DAFx): A New Class of Sound Transformations." IEEE Transactions On Audio, Speech, and Language Processing **14**(5): 1817-1831.

Vincent, E., et al. (2006). "BASS-dB: The Blind Separation Audio Source Separation Evaluation Database." from http://www.irisa.fr/metiss/BASS-dB/.

White, P. (2000). The Creative Process: Pan Position. The Sound on Sound Book of Desktop Digital Sound. UK, MPG Books**:** 169-170.

Zölzer, U. (2002). DAFX Digital Audio Effects. West Sussex, England, John Wiley & Sons, Ltd.

Zölzer, U. (1997). Digital Audio Signal Processing. Chichester, UK, John Wiley & Sons, Ltd.