The technique of elevation filtering and phase shift processing to improve the audibility of sounds in mixing

Shanyu Xie

MA by Research

University of York

Music

April 2015

Abstract

In the area of audio recording and reproduction, mixing is the process by which multiple sound sources (of either a recorded or synthetic origin) are combined, over one or more channels. However, when sound tracks are combined, mix engineers have to take care to preserve quality, as multiple sounds may occur at once. The more masking that occurs during mixing, the worse the audible sound quality. For this research project, I have chosen two methods to help improve audibility of sounds in the final mix. The first is elevation filtering. Elevation filtering is based on the theories of IID, ITD and pinna effect. I decided to design an elevation difference filter, which includes the elevation data. If we utilize such an elevation filter for a sound track, the perceived source position will change and allow it to be distinguished from other sounds. Another method is creating a 180 degree phase shift (out of phase) situation to reduce the level of masking. This technique is based on the theory of the binaural release of masking. Given that an out of phase situation can decrease masking, I have designed a Max/MSP patch which can create an 'out of phase' situation for two ears in any azimuth position. Finally, these two methods will be tested to see if the difference in sound quality is audible or not. A musical composition will also be created, utilizing these two technologies, which will constitute the musical outcome of my research.

List of contents

List of Figures

Author's declaration

I, Shanyu Xie, declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other University. All sources are acknowledged as References. I acknowledge that any person who consults my thesis, in hardcopy or electronic form, may use it in accordance with the provisions for 'fair dealing' set out of the Copyright Act. This includes:

Making reasonable quotation from it, provided that proper acknowledgement is made, for the purposes of criticism, review or news reporting; Making a single copy of a reasonable portion of it for the purposes of research or private study.

I authorize University of York to lend this thesis or Major Research Paper to other institutions or individuals for the purpose of scholarly research. I further authorize University of York to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

To the best of my knowledge and belief the work does not infringe the copyright of any other person.

1. Introduction

My MA in music technology by research deals with mixing technology. In the area of audio recording and reproduction, mixing is the process by which multiple sound sources (of either a recorded or synthetic origin) are combined, over one or more channels. In short, mixing can be described as the combination of audio tracks. The original sound source is irrelevant; it can be recorded voice, instruments, digital instruments or entirely synthetic sound. What is more, as part of the mixing process, the source signals' levels, frequency content (EQ), dynamic range, and panoramic positions (panning) are edited, and effects such as reverb may be added. However, when sound tracks are combined, mix engineers have to take care to preserve quality, as multiple sounds may occur at once. The more sound tracks dealt with, the worse the quality of the sound to the listener. Imagine a piece of music with more than 16 sound tracks combined together, all playing back at the middle pan position. In this case, sound tracks with a high intensity level will obscure any tracks with a significantly lower level. In particular, tracks with similar frequency content are subject to such masking. The more masking that occurs during mixing, the worse the audible sound quality. There are many methods for increasing the audibility of sounds in such situations. For example: panning. We often pan specific sounds to unique positions, allowing listeners to more clearly distinguish individual sounds. Another method is EQ adjustment. For drums and bass, low frequencies are often boosted, and higher frequencies tend to be boosted for stringed instruments. Enhancing the spectral character of sounds in this way helps them to be more distinct.

For this research project, I have chosen two methods to help improve audibility of sounds in the final mix. The first is elevation filtering. Elevation filtering is based on the theories of IID, ITD and pinna effect. IID and ITD are two main cues for identifying the source location of a sound. 'Differences in arrival time between the two ears are called interaural time differences (or ITDs), Interaural time differences are the most important cues to sound location… The other cue to sound location is the differences in the level of a sound at two ears' (Plack, 2005). The pinna can help to discern the elevation of a sound source (Plack, 2005). This is possible because of the ridges and cavities in the pinna. 'The ridges and cavities modify the incoming sound by processes including resonance within the cavities' (Plack, 2005). Given this information, I decided to design an elevation difference filter, which includes the elevation data. This consists of the IID, ITD and the impulse response of the pinna. The function for measuring these data is known as the Head-Related Transfer Function. If we utilize such an elevation filter for a sound track, the perceived source position will change and allow it to be distinguished from other sounds. Take the example of a piano and a guitar playing the same melody, panned centrally. In this situation, they will mask each other, because they are playing the same melody and their frequency profiles are similar. If we add 45 degrees of difference, using the elevation filter, to the guitar track, moving it up above the piano (which remains at 0 degrees), the two sounds will become clearly audible. Another method is creating a 180 degree phase shift (out of phase) situation to reduce the level of masking. This technique is based on the theory of the binaural release of masking. When the sound is 'out of phase' (the phase in one ear is the inverse of the phase in the other), at low frequency, the MLD will be a maximum of 15dB (Moore, 1977). This alleviates the masking problem by 15dB at low frequencies. 'For broadband noise maskers, the MLD falls to 2-3 dB for signal frequencies above about 1500 Hz' (Moore, 1977). Given that an out of phase situation can decrease masking, I have designed a Max/MSP patch which can create an 'out of phase' situation for two ears in any azimuth position. For sounds which are panned centrally, the IID and ITD are 0, so the phase of one channel can be inverted to decrease the level of masking. However, if a sound is panned to a non-central position, the IID and ITD are not 0. If we directly invert the phase of one channel, the signal from the speakers is no longer perceived at 180 degrees (because the ITD is not 0 when the sound is not in the middle). Thus, I have designed three functions in Matlab, which can calculate the amplitude combination and phase combination for two ears, and convert these data to a format readable by a Max/MSP patch, which attempts to create the 'out of phase' situation for two ears in any azimuth position.

Finally, these two methods will be tested to see if the difference in sound quality is audible or not. A musical composition will also be created, utilizing these two technologies, which will constitute the musical outcome of my research. In addition, the research report will include a lot of Matlab code. Because it is unusual for mixing operations to be carried out in Matlab, extra details are given to make this report accessible to a wider range of musicians and mixing engineers.

Although this paper has mentioned and described the knowledge and principles of binaural hearing and makes use of binaural tools, the final work and music outcome is ultimately interested in utilising these techniques and ideas for loudspeaker listening.

2. Literature review

2.1 The history of mixing

In the field of audio recording and reproduction, mixing is the process by which more than one sound sources of any origin are combined over one or more channels. For example, recording an orchestral symphony performance, and we have a range of instruments such as strings (violin, viola, cello etc.), woodwind (clarinet, oboe, bassoon etc.), brass (trumpet, trombone, horn etc.) and percussion (timpani, marimba etc.), with the intent to mix over 16 input channels to two output channels to be played back on loudspeakers or headphones. This is the process of mixing. The various channel levels, frequency profiles (EQ) and pan positions may also be edited, and some effects like reverb may also be used at this point.

As Izhaki (2008) put it, 'where or when the art of mixing was born is not an easy question to answer.' In his book 'Mixing Audio' (2008), he describes the history of mixing music:

'It was the appearance of the multitrack tape machine during the 1960s that signified the dawn of mixing. Multitracks allowed us to repeatedly play record material before committing sonic treatment to the mix. Equalizers, compressors and reverb emulators soon became familiar residents in studios. The audio consoles grew in size to accommodate more tracks and facilities. 8

tracks became 16 and 16 tracks became 24 tracks. Now we had more sonic control over individual tracks and over the final mater' (Izhaki, 2008).

Izhaki continues to describe the history of mixing:

'The 1990s significantly reshaped mush of the way music is made, produced, recorded mixed and even distributed - computers triumphed. Realtime audio plugins were first introduced with the release of Pro Tools III as far back as 1994. Steinberg and its 1996 release of Cubase VST that pioneered the audio plugins that we now take for granted. Professional studios will always, it seems, have an advantage over project studios, if only for their acoustic qualities. However, DAWs offer outstanding value for money, constantly improving quality and widening possibilities' (Izhaki, 2008).

When mixing, we must often consider source signals' amplitude, and ranges of meter (bar meters, peak meters, average meters), and adjust their levels appropriately. We must also carefully consider their frequency spectra, adjusting them with EQs (equalisers). We can alter the dynamic range for a particular situation by utilising compressors and expanders. What is more, we can create the illusion of space, by adding reverberation. While these processes can be approached creatively and utilised for a variety of results, at mixdown we may encounter the issue of masking.

'We can look at the instrumentation of orchestral pieces as a very primitive form of mixing - different instruments that played simultaneously could mask one another. So a producer would place musicians in a room so that the final recording would make sense in terms of level and depth' (Izhaki, 2008). In my research project, the primary focus is on how such masking issues can be alleviated, to make the music to be more audible.

2.2 IID, ITD and the IID, ITD recreation

2.2.1 ILD, ITD and the pinna effect

The two most vital cues for sound location are Interaural time differences and Interaural level differences (Plack, 2005). If a sound is coming from the listener's right side with a finite speed (340 meters per second in the air), the sound will arrive at the listener's right ear before it arrives at the listener's left ear (see Fig 2.1).

Time1 Time2

FIG 2.1 'THE VIEW OF THE HEAD WITH A SOUND SOURCE TO THE RIGHT. THE SOUND WAVES ARRIVE AT THE RIGHT EAR BEFORE THE LEFT EAR' (AFTER PLACK, 2005).

This is because the sound can travel to the listener's right ear directly, but will have to diffract around the head to reach the left. The arrival time depends on the path length, including the distance the sound travels as it bends around the listener's head (see Fig 2.2).

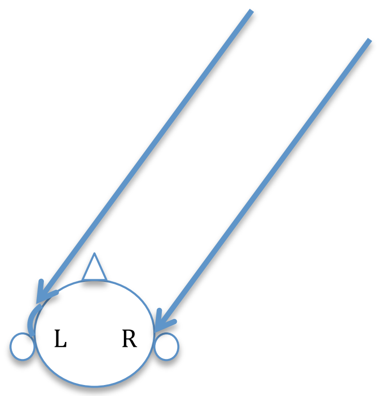


FIG 2.2 'THE LINES INDICATE THE PATHS OF SOUND WAVES ARRIVING FROM THE RIGHT. THE THICK LINE SHOWS THE TIME DIFFERENCES BETWEEN TWO EARS' (PLACK, 2005).

'Differences in arrival time between the two ears are called interaural time differences (or ITDs), Ineteraural time differences are the most important cues to sound location' (Plack, 2005). The other major cue to sound location is the differences in the level of a sound between the two ears (Plack, 2005). Sound intensity decreases as the distance from the source increases. If the left ear is far away from the sound source in comparison to the right ear, the level of the sound reaching the left each will be lower. If a high frequency sound source comes from the listener's right side, then the sound will be much more intense at the right ear than the left ear. This difference is called interaural level difference, or (ILD).

With the ILD and ITD cues, we can accurately identify whether the sound came from the left, right or middle. But how can it be possible to identify whether the sound comes from the front or rear? Or from above or below? The answer lies with the ear's pinna. The pinna alters a sound in a way that differs if the sound is coming from above or below (Plack, 2005), and azimuth also changes by the pinna. This is because the structure of pinna contains ridges and cavities. 'The ridges and cavities modify the incoming sound by processes including resonance within the cavities' (Plack, 2005). Because the cavities are small, the resonances can only affect high frequency sound components, with short wavelengths. The resonance introduces a set of peaks and notches in the high frequency region of the sound's spectrum, and the peaks and notches will be different depending on the angle from which the sound arrives at the pinna (see Fig 2.3).

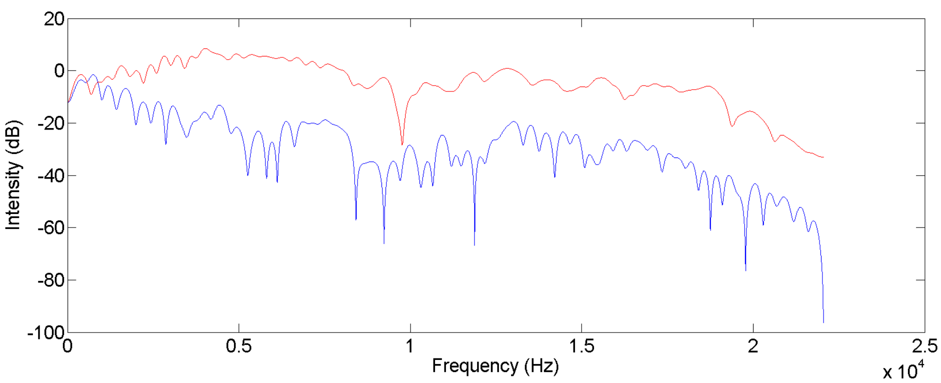


FIG 2.3 MAGNITUDE RESPONSE FROM ELEVATION OF EITHER -45 (BLUE LINE) OR +45 (RED LINE) DEGREES RELATIVE TO THE HEAD. THE SPECTRA DEMONSTRATE THE DIRECTION-SPECIFIC FILTERING PROPERTIES OF THE PINNA.

Thanks to the pinna effect, a listener is able to distinguish the location of a sound in three dimensions. "The pinna imposes the directional "signature" on the spectrum of the sound that can be recognized by the auditory system and used as a cue to location" (Plack, 2005). What is more, the pinna also introduces a shadowing effect for sounds originating behind the head. This shadowing attenuates high frequency sounds coming from the rear, and helps to resolve the ambiguities of distances from the front and back. Pinna effects are thought to be particularly significant for determining the elevation of sound sources.

Because the human ear works with ILD, ITD and the pinna effect, we are able to identify the location of sounds with remarkable accuracy. To take advantage of this, we can cycle through every azimuth angle and every elevation angle to capture an impulse response of a particular ear for sound arriving from any direction, allowing us to create a HRIR (Head-Related Impulse Response). The transfer function of the HRIR is the HRTF. HRIR data incorporates information about the ILD, ITD, and pinna effect for the particular subject. Using HRIR data, we can make convincing 3D music.

2.2.2 IID and ITD recreation

Binaural distance measurement

An inequality in the distance of each ear from a source is known as the binaural distance difference (Guilck, 1971). It is also important to consider the intensity, because sound pressure decreases with propagation. The binaural distance difference is always zero when the source is in the middle plane. On the other hand, a maximum binaural distance difference occurs for sources at an azimuth of 90^{o}. In fig 2.10, with the source at 30^{o} azimuth, and the source far enough away to consider the sound as approaching the head on a broad front, the binaural distance difference represents only a small proportion of the total distance. As Gulick (1971) stated: 'In the case of a source within one meter (1 m) of the head, the binaural distance difference (D) can be approximated in centimeters by the formula:

$$D = k\,(2\beta)$$

Where k is the radius of the head (about 8.75 cm) and the angle β, expressed in radians, is the azimuth. For a more distant source the formula becomes:

$$D = k\,(\beta + \sin\beta)$$

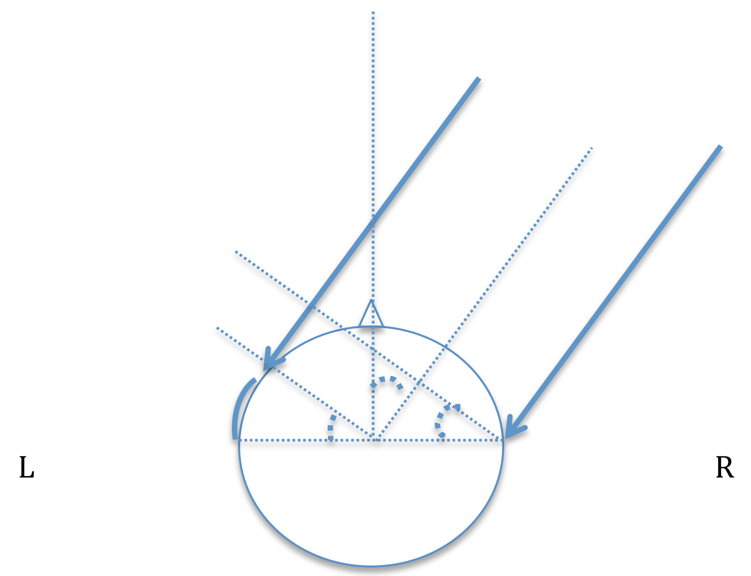


FIG 2.4 'GEOMETRICAL CONSIDERATIONS OF THE INFLUENCE OF THE DISTANCE (NEAR, FAR) OF A SOURCE AT 30^{o} AZIMUTH ON THE BINAURAL DISTANCE DIFFERENCE' (AFTER GULICK, 1971).

With these two formulas, we can now calculate the distance the sound travels to each ear. We can then calculate the time it takes to travel these distances. As previously stated, the speed of sound in air is 340m/s. $\mathrm{D_L}$ (D for left ear)/340 and $\mathrm{D_R}$ (D for right ear)/340 are the times sound takes to travel to the left and right ear, respectively. Subtracting $\mathrm{D_L}$ from $\mathrm{D_R}$, we can calculate the ITD of 30^{o} azimuth in millisecond ($\mathrm{D_L} - \mathrm{D_R} = D$).

Measurement of the IID for 0-180^{o} azimuth and 250-10,000 Hz frequency.

For the second part of my research, I created a Max/MSP patch, which can calculate the Interaural Phase Difference (IPD) for any azimuth measurement and frequency from 0 to 20000 Hz. And before this, we need to know the IID and ITD for every azimuth and frequency. The method for calculating the ITD was already covered in section 2.2.2. The next step is to work out how to calculate the IID. Fig 2.5, from Gulick (1971), shows differences in intensity in decibels at the ear for pure tones of 250, 1000, 5000 and 10,000 Hz, at 0, 45, 90, 135 and 180 degree azimuths. 'Measures were taken 20 cm from the opening on the binaural axis under open-field conditions with the source 2 m from the center of the head'

(Gulick, 1971). Using the figure, we look up the actual intensity of these five angles and four frequencies, such as 20 dB at 90^o and 10,000 Hz. Fig 2.6 shows wavelength against intensity at an azimuth of 90^o. Gulick (1971) provides these wavelength measurements for each of the four frequencies: 10,000 Hz = 3.44 cm, 5000 Hz = 6.88 cm, 1000 Hz = 34.4 cm, and 250 Hz = 137.6 cm.

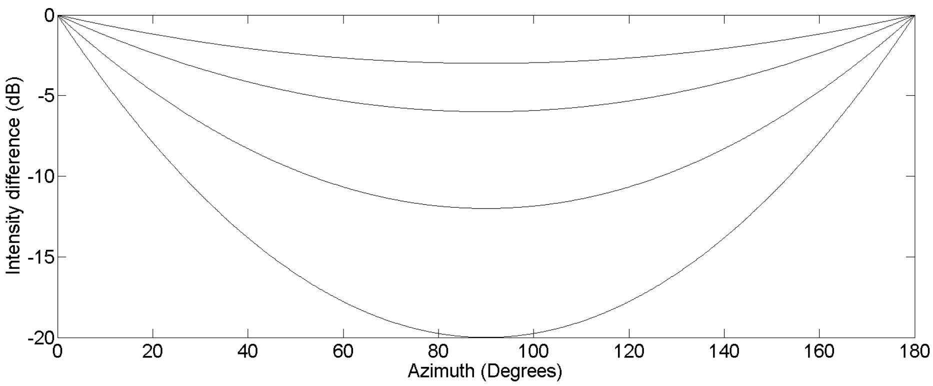


FIG 2.5 'DIFFERENCE IN INTENSITY IN DECIBELS AT THE EARS FOR PURE TONES OF 250, 1000, 5000, AND 10,000 HZ AT EACH OF FIVE AZIMUTHS' (AFTER GULICK, 1971).

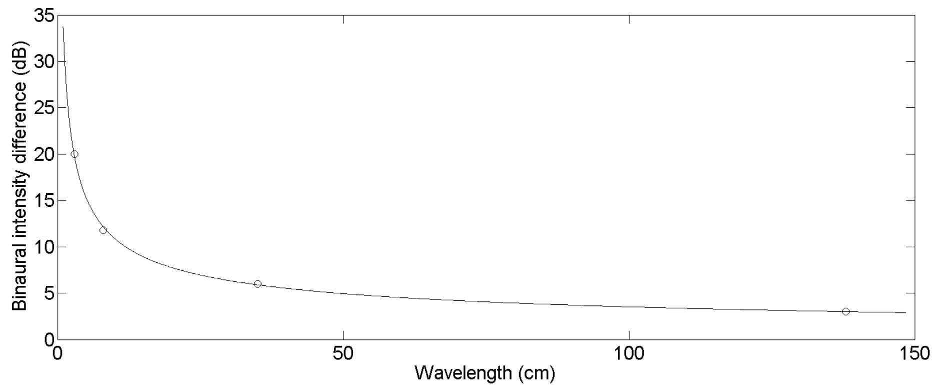


FIG 2.6 'BINAURAL INTENSITY DIFFERENCE IN DECIBELS FOR PURE TONES AT 90^o AZIMUTH AS A FUNCTION OF WAVELENGTH IN CENTIMETERS' (AFTER GULICK, 1971).

But the problem is that Gulick didn't provide any data except these four frequencies, and five azimuth angles. So how can we calculate all of the IID data for all frequencies and azimuths? The answer is to recreate these two figures in Matlab, and estimate the curves (fit a polynomial), to allow us to interpolate between the data for any azimuth and any frequency.

A Structural Model for Binaural Sound Synthesis

In 1998, C. Phillip Brown and Richard O. Duda presented a structural model for synthesising binaural sound sources from a mono sound source via wave propagation and diffraction techniques (Brown and Duda 1998). Brown and Duda introduced a basic description of Three-dimensional Sound and Head-Related Transfer Function HRTF/HRIR. Their research indicated that human torso, shoulders, head and outer ears (pinnae) could modify and affect the spectrum of the sound that reached the ear drums. The authors designed a structural model which included information from these three HRTFs (head shadow & Inter aural Time Difference [ITD], shoulder echo and pinna features), then synthesised this with a mono input, and final created the stereo output with binaural information.

Before Brown and Duda (1998) discussed the structural model, their paper (1998) described the physical basis for the HRIR. It first considered the low-frequency effects of the head alone, and then investigated the effects of the shoulders and pinna. For the low-frequency effects of the head alone, these researcher referred to the Rayleigh's Spherical Model. This model obtained a simple and very useful low-frequency approximation by deriving the exact solution for the diffraction of a plane wave by a rigid sphere. For the investigation of the effects of the shoulders and pinna, Brown and Duda (1998) referred to the Knowles Electronics Manikin for Acoustic Research (KEMAR) and also quoted Kuhn (1987) by stating that:

'As Kuhn has demonstrated, by removing the pinnae one can see effect of the torso and the nonspherical head, and by remounting the pinnae one can see the modifications that the pinna introduce.'

The structural model, firstly, is constituted in three parts: head shadow model – Infinite Impulse Response (IIR); shoulder-echo model – Finite Impulse Response (FIR); and pinnae-echo model (FIR). The principle of the model is the mono signal which first goes to the head shadow model and the shoulder model at the same time (HRTF of the left ear and because the right ear is different, the signal sends to two groups of this model); then synthesises the head shadow model's signal and the shoulder model's signal together and goes the pinnae model. With reflection coefficient and time delay processing, the signal will finally come together and will give an output to our ears (the left signal output to the left ear and the right signal output to the right ear). Furthermore, Brown and Duda (1998) argued that:

The rationale for this structure is that sounds can reach the neighborhood of the pinnae via two major paths--diffraction around the head and reflection from the shoulders. In either case, the arriving waves are altered by the pinna before entering the ear canal.

It therefore contains separate components for azimuth (head shadow & ITD) and elevation (pinnae and shoulder echoes).

In conclusion, Brown and Duda, (1998), indicated that they only adopted one parameter in their experiments and work with three subjects. A larger scale study would be required to determine which parameters are the most important for customisation (Brown and Duda, 1998). The limitations of the evaluation was also referred to, in that it was considered that it did not answer the question of how well the model recreates the illusion that a sound source is located at a particular place. Brown and Duda (1998) suggested that future work should include absolute localisation tests and they observed that the project was limited to the frontal half space and that whilst head motions are highly effective in resolving front/back confusion, the shoulder echo may play an important role for static discrimination (Brown and Duda, 1998). Overall, it was considered that structural models would play a key role in future spatial auditory interfaces.

Although Brown and Duda's (1998) research is recognised, in this research project, the work of the Matlab Code and phase shift processing are based on the principle from Gulick.

The Inverse Square Law

'As the sound spreads out from a source it gets weaker. This is not due it being absorbed but due to its energy being spread more thinly' (Howard, 1996). He continues to state: 'Consider a half blown-up spherical balloon which is coated with honey to a certain thickness. If the balloon is blown up to double its radius, the surface area of the balloon would have increased fourfold' (fig 2.7).

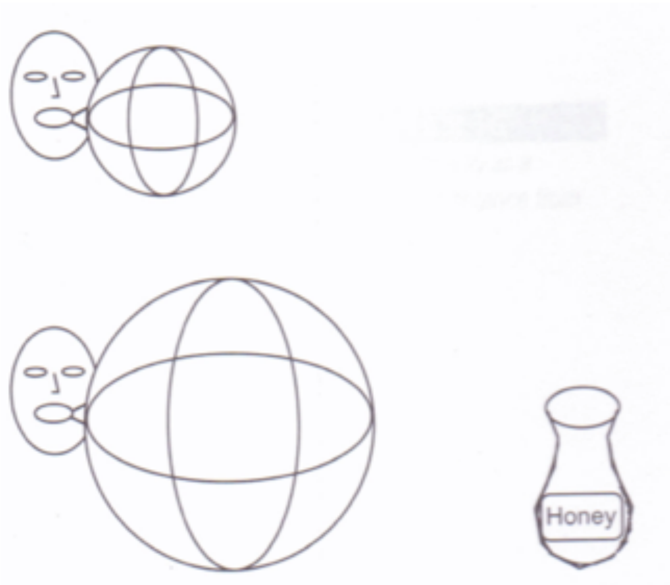


FIG 2.7 THE HONEY AND BALLOON MODEL OF THE INVERSE SQUARE LAW FOR SOUND (HOWARD, 1996).

If we want to calculate the IID we must first work out the intensity of the sound arriving at each ear. There is an inverse square relationship between sound intensity and the distance from its origin: this is the inverse square law.

The area of a sphere is given by the equation:

$$A_{sphere} = 4\pi r^2$$

The sound intensity is defined as the power per unit area. Therefore the sound intensity as a function of distance from a sound source is given by:

$$I = \frac{W_{source}}{A_{sphere}} = \frac{W_{source}}{4\pi r^2}$$

Where I = the sound intensity

W_{source} = the power of the source

and r = the distance from the source

This equation shows that the intensity for a sound wave that propagates in all directions from a source reduces relative to the square of the distance. This reduction in intensity is purely a function of geometry, and is not due to any process of physical absorption.

AT middle and high frequency area, the result (ILD) would be deviation due to the absorption of the head (see fig 2.6).

2.3 Masking issues and binaural release from masking

Binaural cues and decreasing the effect of masking

If a high-frequency signal is coming from the right, then it is much more intense at the right ear than the left because of the IID and ITD. The case is the converse when the signal originates at the left side. But in a situation where the signal is coming directly from the middle, the IID and ITD tends to be low. 'Suppose that the same noise and the same pure tone are presented to both a listener's ear over headphones so that the noise acts to mask the tone' (Plack, 2005). This effect is known as 'masking'. We can utilize binaural to release from masking, to some extent. 'More interesting perhaps is the use of interaural time differences to reduce the masking effect of one sound on another' (plack, 2005).

'In the more than two decades since 1948 now classical paper, much has been learned about the ability of the human auditory system to detect partially masked binaural signals. It is now well established that certain binaural stimulus configurations lead to detection performance which is indistinguishable from that for monotic listening, whereas other configurations lead to performance which is as much as 25 dB better. These increases in detectability are known as masking-level differences (or MLDs), and their existence implies that certain binaural conditions provide the auditory nervous system with information in addition to that available in the monotic condition or in those binaural conditions that do not lead to MLDs.' (McFadden, 1971).

The masking threshold of a signal can sometimes be markedly lower when listening with two ears than listening with one ear (Moore, 1977). For example, the situation in fig 2.8 (a) shows two noises from the same noise generator being fed to both ears via stereo headphones. At the same time, a pure tone also from the same signal generator, is fed separately to each ear and mixed with the noise (Moore, 1977). The combined signals at the two ears are identical because of the masking problem. 'Assume that the level of the tone is adjusted until it is just masked by the noise, and let its level at this point be L_0 dB' (Moore, 1977). Now assume that keeping everything else the same, but with one ear's tone inverted (the waveform is flipped upside down). This action is equivalent to shifting the phase of the signal by 180^o or π radians (fig 2.8 (b)). The tone becomes audible again because the masking is released. Adjusting this level by L_π, the difference between the two levels, $L_0 - L_\pi$ (dB), is known as a masking level differences (MLD) (Moore, 1977). Moore (1977) continues to describe this: 'its value may be as large as 15 dB at low frequencies (around 500Hz), decreasing to 2-3 dB for frequencies above 1500 Hz. Thus simply by inverting the signal waveform at one ear the signal can be made considerably more detectable'.

Fig 2.8 (c) shows that the noise and signal are fed to one ear only, the signal is adjusted to e at its masking threshold. If the noise alone is added to the other ear (fig 2.8 (d)), the tone becomes audible again. Thus, adding noise at the ear not receiving the tone makes the tone somewhat more perceptible (Moore, 1977).

FIG 2.8 ILLUSTRATION OF TWO SITUATIONS IN WHICH BINAURAL MASKING LEVEL DIFFERENCE OCCUR. IN CONDITIONS (A) AND (C), DETECTABILITY IS POOR, WHILE IN CONDITIONS (B) AND (D) (WHERE THE INTERAURAL RELATIONSHIPS OF THE SIGNAL AND MASKER ARE DIFFERENT) DETECTABILITY IS HIGH (AFTER MOORE, 1977).

'A particular situation can be described by using the symbols N (for noise) and S (for signal), each being followed by a suffix denoting relative phase at the two ears' (Moore, 1977). Given this information, we can use N_0S_0 to denote the condition where both the noise and the signal are in phase at both ears (Fig 2.8 (a)). Similarly, N_0S_π refers to the situation in which the noise is in phase at both ears, but the signal is inverted in phase (Fig 2.8 (b)). N_u means that the noise is uncorrelated at the two ears. The suffix m indicates monaural presentation. Fig 2.9 gives the magnitude of the MLD for a variety of combinations of signal and noise.

Interaural condition	MLD (dB)
$N_u S_\pi$	3
$N_u S_0$	4
$N_\pi S_m$	6
$N_0 S_m$	9
$N_\pi S_0$	13
$N_0 S_\pi$	15

FIG 2.9 VALUES OF THE MLD FOR VARIOUS INTERAURAL RELATIONSHIPS BETWEEN THE SIGNAL AND MASKER. THESE RESULTS ARE TYPICAL FOR BROADBAND MASKERS AND LOW-FREQUENCY SIGNALS (AFTER MOORE, 1977).

Now we can see that if the sound is presented with the $N_0 S_\pi$ condition at low frequencies, the MLD will reach a maximum of 15 dB. This means that masking can be alleviated by 15 dB at low frequencies. 'For broadband noise maskers, the MLD falls to 2-3 dB for signal frequencies above about 1500 Hz' (Moore, 1977). Original research by McFadden (McFadden, 1972) presents more detail on the MLD, at 1000 Hz and 2000 Hz:

'The center frequency of the noise band was either 1000 or 2000 Hz, in different experiments both diotic (N0-S0) and dichotic (N0-Sπ) data were taken at both frequencies. By varying the signal-to-masker ratio, and the angle α at which the signal is added to the masker in the N0-Sπ condition, it is possible to control the magnitudes of the two binaural cues - interaural time difference and interaural level difference. At 1000 Hz, sizeable masking-level differences (MLDs) were observed for all subjects at all values of α. At 2000Hz, the MLDs were large at $\alpha = 0^o$; however, detectability was essentially the same in the conditions N0-Sπ $\alpha = 90^o$, and N0-S0, $\alpha = 90^o$'.

When the angle of α is 0^o, it means that there would have been an increment in level in one ear and a decrement in the other, with no interaural time difference (McFadden, 1972). That means the signal will only arrive at one ear. When the angle of α is 90^o, it means that the two ears will obtain the same time delay. We are concerned with the binaural hearing situation, so we should focus on $\alpha = 90^o$. As McFadden (1972) indicates, sizeable MLDs are observed at 1000 Hz, that means situation N0-Sπ (out of phase) can reduce the masking level compare with situation N0-S0 (in phase). However, when the frequency is higher (2000 Hz), the two situations, N0-S0 and N0-Sπ, are essentially the same, $\alpha = 90^o$. It means that in a higher frequency range (2000 Hz), the MLD will not be significantly changed when we set up the phase shift, even at 180 degrees. When the frequency is more than 2000 Hz, the 'out of phase' situation does not release the MLD any more.

2.4 HRTFs filter design to release the masking

2.4.1 The Head-Related Transfer Functions

'Spectral filtering of a sound source before it reaches the ear drum that is caused primarily by the outer ear is termed the head-related transfer function (HRTF)' (Field, 2000). As Field continues to state: 'The binaural HRTF (terminology for referring to both left- and right-ear HRTFs) can be thought of as a frequency-dependent amplitude and time-delay differences that result primarily from the complex shaping of the pinnae.'

Utilisation of HRTFs is a key consideration for 3-D sound system and HRTF mixing.

2.4.2 the CIPIC HRTFs database

The CIPIC (Center for Image Processing and Integrated Computing) was foundation by Prof. V. Ralph Algazi, at the University of California, in 2001. As their official website described (2001): 'The CIPIC HRTF Database is a public-domain database of high-spatial-resolution HRTF measurements for 45 different subjects, including the KEMAR mannequin with both small and large pinnae'.

'The database includes 2,500 measurements of head-related impulse responses for each subject. These "standard" measurements were recorded at 25 different interaural-polar azimuths and 50 different interaural-polar elevations. Additional "special" measurements of the KEMAR manikin were made for the frontal and horizontal planes. In addition, the database includes anthropometric measurements for use in HRTF scaling studies, technical documentation, and a utility program for displaying and inspecting the data' (Algazi, 2001).

Capturing the HRIR from the CIPIC database

Our motivation for capturing the HRIR is to utilize this data to design our FIR filter. If we know the HRIR for a particular azimuth and elevation pair (such as +45 azimuth and +45 elevation), then we can use these data in the design of the FIR filter, finally creating a new HRIR filter which includes the azimuth or elevation data. Any audio passing through this filter will be perceived at +45 azimuth or +45 elevation position relate to other sounds.

Let us outline an example, capturing the magnitude response of a HRIR at 0 azimuth and 45 elevation.

First, we randomly choose subject 015 for the test. In the documentation for the UCD HRIR Files (Algazi, 2001), it provides the HRIR values: 'HRIR values are given for 25 different azimuths, 50 different elevations and 200 instants in time.'

The azimuth angle vector in Matlab is:

$$\text{azimuths} = [-80 - 65 - 55 - 45{:}\,5{:}\,45\ 55\ 65\ 80]$$

So, the vector can be expressed as the azimuth from -80 to 80 degrees, in steps of 5 degrees. Elevations range from -45^o to $+230.625^o$ in steps of 5.625^o (Algazi, 2001). In Matlab, the elevation angle is the element of the vector:

$$\text{elevations} = -45 + 6.625 * (0{:}\,49)$$

The temporal sampling frequency is $f_s = 44.1\text{ kHz}$ (Algazi, 2001).

code:

```
elevations = -45 + 5.625*[0:49]

azimuths=[-80 -65 -55 -45:5:45 55 65 80]

50 elevations and 25 azimuths included.
```

In this example, the azimuths will all be set to 0 degrees, so the azimuth vector required is number 13 (#13 = 0 degrees). For the elevation vector we will select the value of 45 degrees (#17 = 45 degree).

elevation: 0 azimuth 0

code:

```
h1 = reshape(hrir_r(13,9,:),1,200);

[h1,f]=freqz(h1,1,2^17,44100);

plot(f,20*log10(abs(h1)))
```

The plot image is shown in Fig 2.10. It shows the magnitude response of 0 azimuth and 0 elevation.

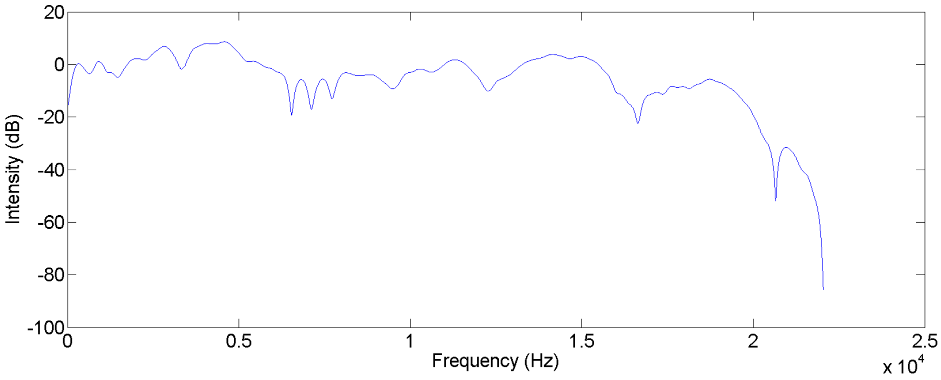


FIG 2.10 THE MAGNITUDE RESPONSE OF 0 AZIMUTH AND 0 ELEVATION.

elevation: 45 azimuth 0

code:

```
h2 = reshape(hrir_r(13,17,:),1,200);

[h2,f]=freqz(h2,1,2^17,44100);

plot(f,20*log10(abs(h2)))
```

The plot image is shown in fig 2.11. It shows the magnitude response of 0 azimuth and 45 elevation.

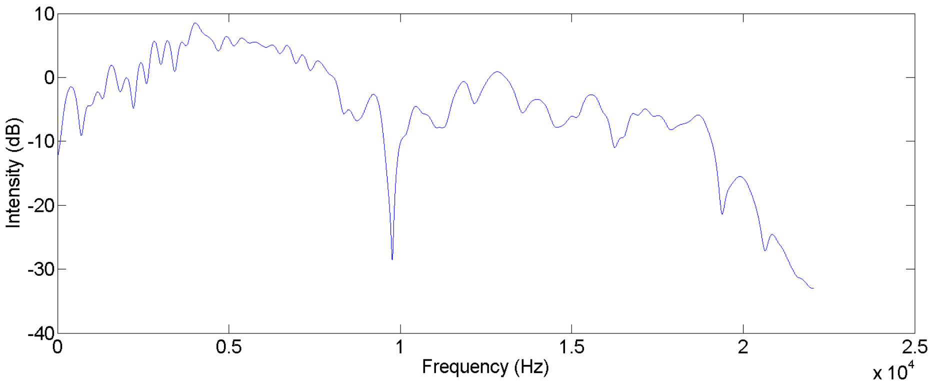


FIG 2.11 THE MAGNITUDE RESPONSE OF 0 AZIMUTH AND +45 ELEVATION.

2.4.3 Z-transform and HRTF filter design

The Z-transform is a transform function, which transfers the filter from the time domain to the frequency domain, or vice versa. Editing the zero and pole of the Z-plane can determine the time domain filter coefficients (Wells, 2000). Additionally, we can use the transform to derive the magnitude responses from time domain filter. The coefficients 'b' and 'a' represent the zeros and poles on the Z-plane, respectively. Because we are designing a FIR filter, there are no poles (the coefficient 'a' is 1) and we only need to concern ourselves with the zeroes.

An example of a filter in the time domain and the Z-transform domain is shown below (Wells, 2015):

$$y[n] = 1.0x[n] - 0.5x[n-1] + 1.0x[n-2] + 0.5y[n-1]$$

$$y = x(1 - 0.5z^{-1} + 1.0z^{-2}) + y(0.5z^{-1})$$

$$y(1 - 0.5z^{-1}) = x(1 - 0.5z^{-1} + 1.0z^{-2})$$

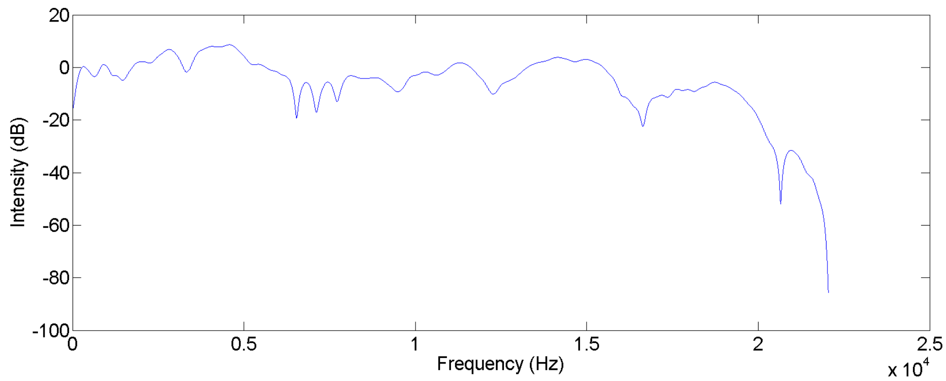
$$\frac{y}{x} = \frac{1 - 0.5z^{-1} + 1.0z^{-2}}{1 - 0.5z^{-1}}$$

The final line of this equation is the transfer function of the filter. In order to turn the top part into a quadratic function, we must multiply the equation by z^{-2}. Thus, the equation becomes:

$$\frac{y}{x} = \frac{z^2 - 0.5z + 1.0}{z^2 - 0.5z}$$

Z-transform calculation in Matlab

For example, we want to design a 45-degree difference elevation filter. In order to achieve this, we must know the HRIR at 0 degrees and 45 degrees. We can use the Z-transform within Matlab to analyse the magnitude of these two filters (Fig 2.12 and 2.13).



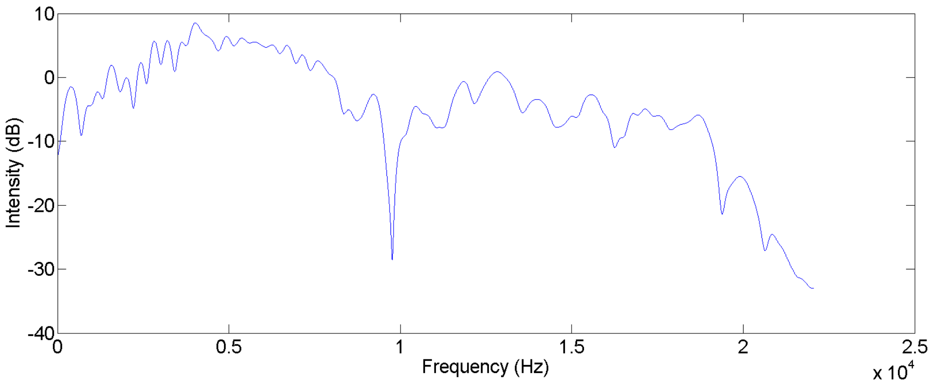


FIG 2.12 AND FIG 2.13 (THE MAGNITUDE RESPONSE OF 0-DEGREE HRTFS FILTER AND 45-DEGREE FILTER)

To obtain the magnitude response on Matlab, we can use the 'freqz' function, and run the following code:

[h,f]=freqz(b0,1,2^17,44100);

(the coefficient 'b' is the HRTF data for the 0-degree filter coefficient 'a' equals 1, filter size is 2^17, sample rate is 44100 Hz)

plot(f,abs(h)) (plot the image of the 0-degree filter's magnitude response)

2.5 The Studer Virtual Surround Panning (VSP) system

Studer introduced the D950S of the D950 mixing console series, which incorporates a new panning technology known as VSP, at the 103rd AES convention in New York (Swiss Sound, 1998). 'These processes are capable of realistically projecting virtual acoustical scenes by means of conventional surround formats.' (Swiss Sound, 1998). The VSP utilises HRTFs to pan a sound's position. The operation of the HRTFs panning mode is as follows. Firstly the signal is sent to a variable filter that depends upon the selected panning angle. A delay line is applied and ITD of two ears for the given panning angle is determined. IID information is determined by reference to HRTFs. After that, the delay information and HRTF information are both sent to allpass filters which are capable of producing a uniform phase rotation. From here the data is sent to the loudspeaker outputs.

'The principal function of the variable FIR filter is the implementation of non-integer multiples of a sampling interval, as well as the prevention of noise when the controls are actuated. With the filters referred to as HRTF, interaural intensity differences are to be simulated.' (Swiss Sound, 1998).

VSP uses HRTFs to achieve sound position panning, potentially yielding a much more accurate result than other mixing console panning systems. That is because the HRTF directly captures the IID, ITD and pinna data from a human's ear. Using the VSP can better reproduce the sound position from the headphone or the loud speaker. The principle of the VSP stereo panpot is shown in Fig 2.14.

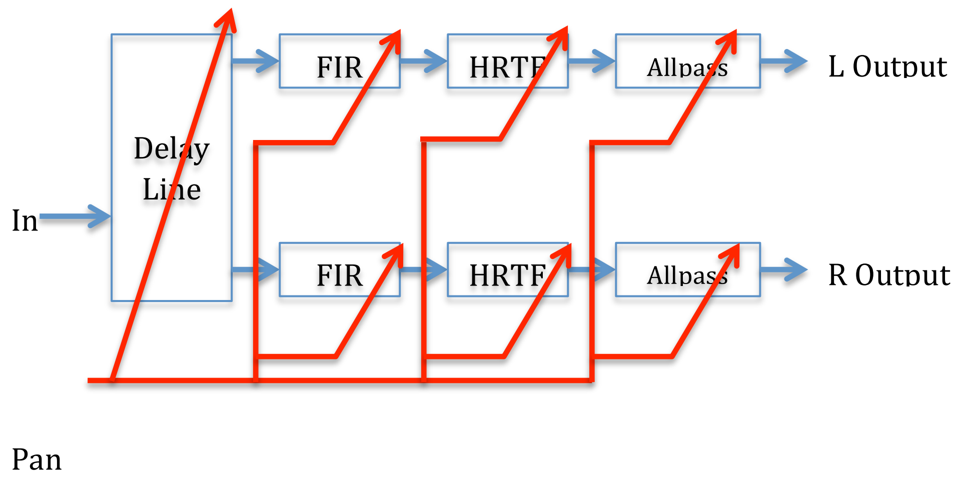


FIG 2.14 THE VSP STEREO PANPOT (AFTER SWISS SOUND, 1998)

VSP includes HRTF panning mode on the console, and this mode can be even use for 5.1, 7.1 and IMAX mixing. It provides a front/back option and front/back control to pan the sound in the front or rear position in 5.1, 7.1 or IMAX playback systems. It also contains the control parameters for the echo mode.

The advantage of VSP's HRTF mode, is its ability to use HRFTs data to pan the sound around a multi-speaker system (such as 5.1). It provides mixing engineers with a new method to pan sound across the front or rear with high quality. Listeners can clearly identify the sound locations because of the HRTFs. However, every listener has a particular group of HRTFs, and an individual listener may not identify the sound location when the HRTF doesn't match theirs. The HRTF panning mode can be used for surround sound panning, but VSP is not using HRFT for elevation panning; it just utilises HRTFs for panning with azimuth. That means the HRTF mode is not supported for elevation panning, - VSP can not provide height information to the listener.

2.6 The benefit of sound's clarity

The purpose of phase shift processing is to make the individual sound more audible. It makes the signal appear to one ear to be in phase and to the other ear to be out of phase, so that the phase shift of both ears remains at 180 degrees. According to the principle of binaural release of masking, keeping the phase shift at 180 degrees, achieves a maximum clarity level of the individual sound (see Section 2.3). This processing might result in a lack of focus and make the sound appear to be free. This is because we pan the sound out of phase with 180 degree phase shift. If the sound arriving at our ears is in phase, the sound is the central focus. However, although out of phase causes the need to adjust the focus of the sound, it makes the sound clearer and more audible.

Some musicians or musical styles tend to demand a concentration of different instruments or sounds in the same location in the stereo field rather than separating the sounds and so requiring a panning method. Orchestration is also not just about masking sound and sound clarity, it is also concerned with the combination of sounds so as to achieve interesting and detailed textures. However, sound clarity still benefits and there will be an improvement in the sound quality. Deruty, Pachet, and Roy (2014) indicated that:

'At lower monitoring levels and/or on budget monitors, with the exception of the bass guitar, track spectrum and loudness are set conjointly so that each track is optimally understandable.

Given such monitoring conditions, audio engineers appear to be concerned by the comprehension of each individual track.'

At a lower monitoring level and at low frequency range, the sound quality should be more focused on the comprehension of each individual track. This conclusion is published in their paper 'Human–Made Rock Mixes Feature Tight Relations Between Spectrum and Loudness' in 2014. They interest was in the spectrum and loudness resulting from human mixing. Deruty *et al*., (2014) also mentioned that the sound quality was a fundamental concern of the mixing process:

'A fundamental concern regarding the mixing process is the extent to which the listener can hear each track making up a mix individually. Indeed, several automatic mixing frameworks are based on the sole hypothesis according to which each track in the mix should be as audible as possible.'

Deruty *et al*., (2014) observed that the utilisation of a low monitoring level and/or budget monitors was able to offer a significant correlation between individual track audibility and loudness. On the other hand, a significant correlation between individual track brightness and loudness could be observed at high monitoring levels and/or on high-end monitors. After the test of high level monitoring and low level monitoring and analysis of the data, Deruty *et al*., (2014) observed that:

'Under bad listening conditions, track audibility stands out as a priority. Under good listening conditions, priority shifts to fitting a specific spectral profile. The success of a mix may lie in the compliance to these two directives.'

Finally, I suggest that Deruty *et al.*,'s (2014) paper proved that the individual sound clarity plays a key role and is important for the audibility and loudness under poor listening conditions.

Although this paper has mentioned and described the knowledge and principles of binaural hearing and makes use of binaural tools, the final work and music outcome is ultimately interested in utilising these techniques and ideas for loudspeaker listening.

3 Elevation filtering

3.1 The purpose of elevation filtering

As previously mentioned, the masking issue occurs while during mixing. When it occurs, it causes sound tracks to cover each other, preventing listeners from identifying each of them clearly. One of the techniques which mixing engineers will use to prevent masking is panning the sound tracks to different sides. Take the example of a very simple song, which includes a voice, a guitar and a drum. If we do nothing and leave them all panned centrally, these three sounds may mask each other. Consequently, the listener may not be able to identify the sound clearly and they may say: ”I can not hear the voice singing clearly” or “I can not actually hear what the guitar is playing”. The solution is to utilise the panning tool, and pan the voice to the left speaker, pan the guitar to the right speaker, and leave the drum in the centre. Now, we can distinguish the individual sounds again because they are not masked, not covering each other.

Following from this, I will attempt to pan the sound using elevation rather than azimuth. In section 2.2.1, the ILD, ITD and the pinna effect were outlined. It was shown that humans could identify the origin of a sound, via these three cues of human’s hearing. Plack (2005) also mentions the magnitude response of sound, which is different from different degrees of elevation (Fig 2.3).

We can see that the magnitude responses of the blue line (-45 degree elevation) and the red line (+45 degree elevation) are totally different. As the first paragraph described, panning sound with azimuth can help prevent masking, to some extent. Now we know that panning with elevation can also achieve this. Moreover, we can benefit from elevation filtering. For example, if we are listening to a classical jazz drum kit performance, we should be able to visualise the kick drum at the bottom of the sound stage. The snare drum should be roughly in the middle, and the hi-hat is at a higher position compared to the snare drum. With elevation panning, we can attempt to reproduce the performance in with sounds perceived at the correct positions.

3.2 The processing of the elevation filter

If we want to filter sound to allow it to be perceived at a certain elevation, we must first determine how to create an elevation filter. The first step is to retrieve the HRTF data from the CIPIC database. The next step is to decide which elevation location we want to pan to. The final step is to calculate the difference between the elevations we want and the zero elevation, then create a new filter with this elevation difference. We will accomplish these three steps using Matlab.

3.2.1 Retrieving the HRTF data from the CIPIC database

In the documentation for the UCD HRIR Files (Algazi, 2001), it provides the HRIR values: 'HRIR values are given for 25 different azimuths, 50 different elevations and 200 instants in time.' The azimuth angle vector in Matlab is:

$$\text{azimuths} = [-80 - 65 - 55 - 45{:}\,5{:}\,45\ 55\ 65\ 80]$$

So, the vector can be described as the azimuth from -80 to 80 degrees, in steps of 5 degrees. Elevations range from -45^{o} to $+230.625^{o}$, in steps of 5.625^{o} (Algazi, 2001). In Matlab, the elevation angle is the element of the vector:

$$\text{elevations} = -45 + 5.625 * (0{:}\,49)$$

The temporal sampling frequency is $f_s = 44.1\ \text{kHz}$ (Algazi, 2001).

From these two equations, we can see that the CIPIC database provides from -80 degree to 80 degree azimuth HRTF data and from -45 degree to +230.625 degree elevation HRTF data. We can see that the sampling frequency is 44100 Hz.

For example, if we now want to capture the 45 elevation degree HRTF data from the CIPIC database from subject 015, we can firstly open the subject 015 HRTF m file from Matlab, then run:

```
az= [-80 -65 -55 -45:5:45 55 65 80];
```

```
el= -45+5.625*(0:49);
```

This acts to set up the range of azimuths and elevations from the HRTF data. The next step is to retrieve the 45 degree elevation HRTF data. Before that though, we must know what is the index of the 45 degree elevation is, and the index of 0 azimuth (because we do not need to by azimuth; we just want to pan the sound at 45 degree elevation, so the azimuth will be zero). We can use the 'find' function from Matlab to find out want number it is:

```
find(az==0)

ans=

    13

find(el==45)

ans=

    17
```

Now that we have discovered the index of 0 azimuth is 13, and the index of 45 elevation is 17, we can proceed to the next step: finding out the magnitude response of 45 degree elevation HRTFs. Because the data is in the time domain, we should use function 'freqz' to find out the magnitude response in the frequency domain:

```
h2 = reshape(hrir_r(13,17,:),1,200);

[h2,f]=freqz(h2,1,2^17,44100);
```

```
plot(f,20*log10(abs(h2)))
```

In this example, we capture the HRTF of the right ear (hrir_r). We can also capture the HRTF data of the left ear by changing 'hrir_r' to 'hrir_l'. Finally, we plot the image of the 45 degree elevation HRIR (Fig 3.1)

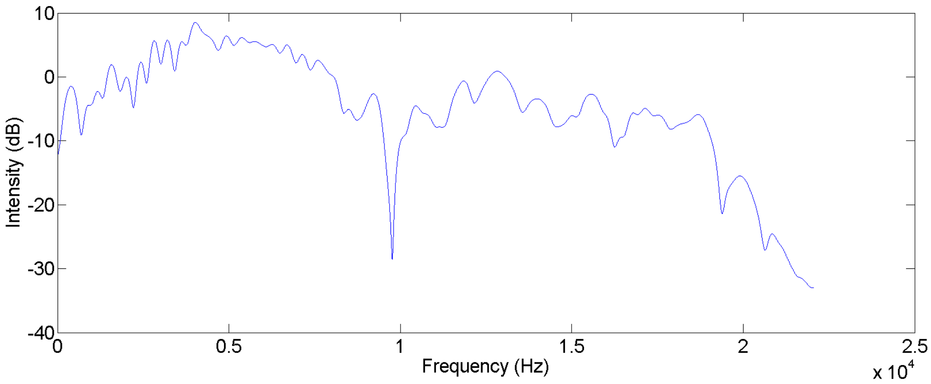


FIG 3.1 THE MAGNITUDE RESPONSE OF 0 AZIMUTH AND +45 ELEVATION.

3.2.2 Calculating the elevation difference and create a new elevation difference filter

In section 3.2.1, we already figured out how to capture the magnitude response of a particular elevation HRTF. In this section, we are going to calculate the elevation difference, and create a new elevation difference filter. In the previous section, we determined the magnitude response of 45 degree elevation. If we want to calculate the elevation difference between 45 elevation and 0 elevation, we also need to know the magnitude response of 0 elevation. We can use 'find' to select the index of 0 elevation, and retrieve its magnitude response:

```
find (el==0)

ans=

     9

h1 = reshape(hrir_r(13,9,:),1,200);

[h1,f]=freqz(h1,1,2^17,44100);

plot(f,20*log10(abs(h1)))
```

If we run this code in Matlab, we are presented with another plot image: the magnitude response of the 0 elevation HRTF (Fig 3.2)

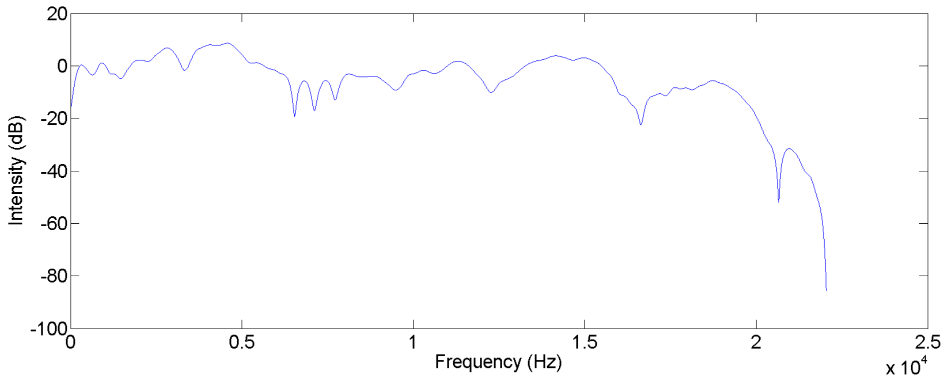


FIG 3.2 THE MAGNITUDE RESPONSE OF 0 AZIMUTH AND 0 ELEVATION.

Now that we have two magnitude responses: 0 elevation and 45 elevation, the next step is to calculate the elevation difference. We can subtract that of 0 elevation from that of 45 elevation to obtain it:

```
newdB = 20*log10(abs(h2))-20*log10(abs(h1));

new = 10.^(newdB/20);

f = f/f(end);                    (adjust the frequency scale from 0 to 1)
```

We use a new character ‘c’ to represent the new filter, and we assign the elevation difference data to this filter ‘c’. To create a filter in Matlab, we can use ‘fir2’ function:

```
c = fir2(199,f,new);

[hc,f] = freqz(c,1,2^17,44100);

plot(f,20*log10(abs(hc)),'r')
```

The filter ‘c’ has now been created. This FIR filter incorporates the 45 degree elevation difference data. If we run this code in Matlab, we should see a plot image with the magnitude response of this filter (Fig 3.3).

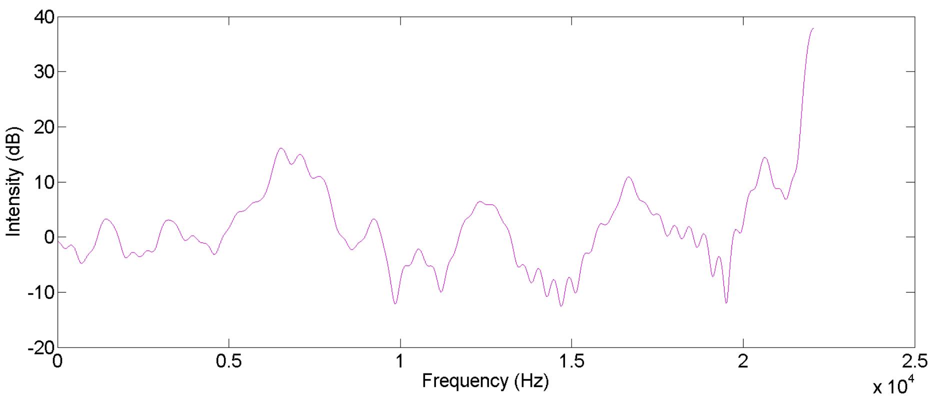


FIG 3.3 45-DEGREE ELEVATION DIFFERENCE FILTER MAGNITUDE RESPONSE

3.3 Load the elevation difference filter in Logic Pro 9

Before we export our elevation filter to music software such as Cubase, Pro tools or Logic Pro, we must first learn how to import a wave file to Matlab and export a wave file from Matlab, with these workstations in mind. The appropriate method is to use the functions 'wavread' and 'wavwrite'. To import a wave file to Matlab, one uses:

```
a= wavread ('wavefile.wav');
```

In this case, we want to save the elevation difference filter to a .wav format audio file. As shown in the previous section 3.2.2, we already know the filter is named 'c', and that the HRTF data is provided as 44.1 kHz f_S. So we keep that sample frequency 44.1 kHz, and set up the bit rate to 24 bit for higher quality (the maximum bit rate of Logic Pro 9 is 24 bit):

```
wavwrite(c,44100,24,el_filter);
```

Usually, most music software supports loading IR (Impulse Response) filters into a plugin of some description. In Logic Pro 9, there is a plugin called 'Space Designer', which can import the elevation difference filter (Fig 3.4):

FIG 3.4 IR FILTER '45IR_FILTER.WAV' LOADED UP USING PLUG-IN 'SPACE DESIGNER' IN LOGIC PRO 9

After importing the IR filter to Logic Pro, we can use this plugin on a sound track to test if we can discern the elevation of the sound. For example, if we are dealing with music, including perhaps one piano and one guitar, playing the same note or melody. We can use the IR filter with the guitar, to see if the masking issues are improved or not. If the audience can clearly distinguish the piano and the guitar sounds, than we can say the masking issues have been alleviated.

4 Phase shift recreation

4.1 Purpose of phase shift recreation

The purpose of phase shift recreation is to create a 180 degree phase shift, or as close to a 180 degree phase shift as is possible. The motivation behind creating a 180 degree phase shift is because when one ear is 'out of phase' (180 degree phase difference), relative to the other ear, the masking problem will be improved. This theory was already outlined in some depth in section 2.3. The 'out of phase' situation can release the masking threshold by a maximum of 15 dB at 500 Hz (Moore, 1977). So, if we can achieve a 180 degree phase shift, or as close as we can achieve at any azimuth, we can obtain the maximum release of masking at any given azimuth, and the music with many sound tracks (such as 16 tracks or 24 tracks), could be more audible with more distinct tracks. Now we know that the objective is to make the phase shift reach 180 degrees between the two ears. This means that the IPD (Interaural Phase Difference) should be 180 or as close as 180 as is possible. If we want to recreate the IPD, we must first know the ILD (Interaural Level Difference), or IID (Interaural Intensity Difference), and ITD from the sound source to our ear. For my research, I decided to create a function, which can calculate the IID, ITD, and IPD, and adjust the IPD in Matlab. I will then create a Max/MSP patch to which the IPD data will be transferred and in which the mix can be recreated.

In addition, although it is easy to create a 180 degree phase shift at the ears with headphones, it is more difficult for non-centrally panned sources, via loudspeakers. This is because when we are listening to sound source from the headphones, the left signal arrives directly at our left ear, and the right signal arrives directly at our right ear; the IID and ITD of the headphones is almost equal to the IID and ITD of the ears. When we are listening with speakers, there is usually a two metres gap between the speakers and our ears, and this distance will change the IID and ITD when the signal arrives at our ears. In addition, the left signal not only goes to our left ear, but also arrives at our right ear and vice versa with the right signal. In this situation, we must accurately calculate the amplitude combination and phase combination of our ears. When our left ear receives the left speaker's signal, we must also consider that the left ear also receives the right speaker's signal, and the combination of these two signals is the final result that we want to obtain.

4.2 Executing the phase shift

4.2.1 Matlab function creation

Recreating the figure from Gulick's book to calculate the IID

Before calculating the IID and ITD, we must first detemine how to get the IID and ITD data. Gulick's book (1971) mentions differences in intensity in decibels at the ear for pure tones of 250, 1000, 5000, and 10,000 Hz, at 0, 45, 90, 135 and 180 degree azimuths. The distance between the sound source and the ear is 2 metres (Fig 2.5). However, he didn't mention how to calculate the IID, and didn't provide any equation to get the IID. In this situation, we can use the 'polyfit' and 'polyval' functions to create this figure and utilize a polynomial to calculate the closest IID value.

From this figure, we can basically ascertain three points on each frequency curve: at 0 degrees, 90 degrees and at 180 degrees. We can simply find that the IID is always 0 dB at 0 or 180 degrees. At 250 Hz, the IID of the 90 degree case is approximately 3 dB. We can also determine the IID at 90 degrees for another three frequencies: 6 dB at 1000 Hz, 12 dB at 5000 Hz and 20 dB at 10000 Hz. For the next step, we want to use 'polyfit' to estimate and recreate these four curves:

```
az=[0,90,180];            (azimuth data: 0, 90, 180)

in=[0,-20,0];             (intensity difference of 10000 Hz: 0, -20, 0)

p=polyfit(az,in,2)

x=linspace(0,180,1000);

y=polyval(p,x)

plot(x,y)                 (plot the 10000 Hz curve with default blue color)

hold on
```

```
ina=[0,-12,0]                (intensity difference of 5000 Hz: 0, -12, 0)

pa=polyfit(az,ina,2);

ya=polyval(pa,x);

plot(x,ya,'r')              (plot the 5000 Hz curve with red color)

hold on


inb=[0,-6,0]                 (intensity difference of 1000 Hz: 0, -6, 0)

pb=polyfit(az,inb,2)

yb=polyval(pb,x);

plot(x,yb,'g')              (plot the 1000 Hz curve with green color)

hold on


inc=[0,-3,0];               (intensity difference of 250 Hz: 0, -3, 0)

pc=polyfit(az,inc,2);

yc=polyval(pc,x);

plot(x,yc,'b')              (plot the 250 Hz curve with black color)
```

When we finish running this code in Matlab, we can obtain an estimate of the figure in Gulick's book (Fig 4.1)

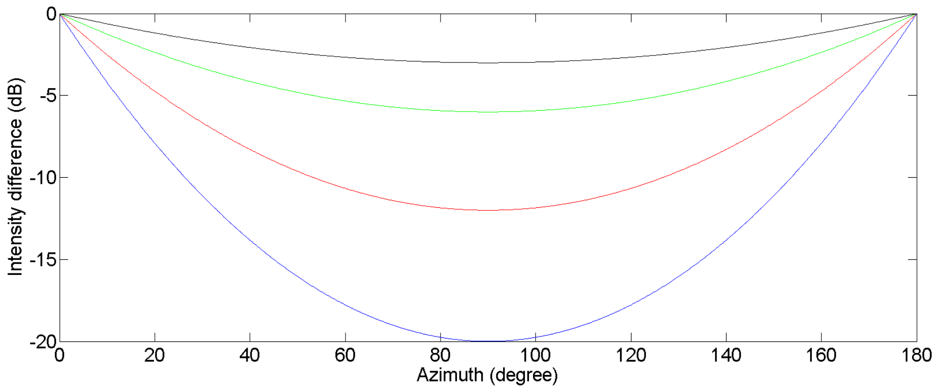


FIG 4.1 THE DIFFERENCES IN INTENSITY (IID) IN DECIBELS AT THE EARS FOR PURE TONES OF 250, 1000, 5000, AND 1000 HZ AT EACH AZIMUTHS (AFTER GULICK, 1971).

Next, we are going to recreate another of Gulick's figures. Fig 2.6 provides four wavelength data for all four frequencies at 90 azimuth: 3.44 cm = 10,000 Hz, 6.88 cm = 5000 Hz, 34.4 cm = 1000 Hz, and 137.6 cm = 250 Hz.

From this figure, we can get the wavelength data at 250 Hz, 1000 Hz, 5000 Hz and 10000 Hz. Next, we want to find out the approximate IID for these four wavelengths: we get 20 dB at 3.44 cm, 11.8 dB at 6.88 cm, 6 dB at 34.4 cm and 3 dB at 137.6 cm. Next, we can use 'polyfit' and 'polyval' again to recreate the figure:

```
x = [3.44,6.88,34.4,137.6];    (Data of wavelength)

y = [20,11.8,6,3];             (Data of intensity difference)

plot(x,y)

figure,plot(log(x),log(y))



p = polyfit(log(x),log(y),1);

xx = linspace(0,5,1000);
```

```
yy = polyval(p,xx);

plot(x,y)

hold on

plot(exp(xx),exp(yy),'r')        (plot the image with red color)
```

When we finish running this code in Matlab, it should obtain an approximation of Gulick's figure (Fig 4.2).

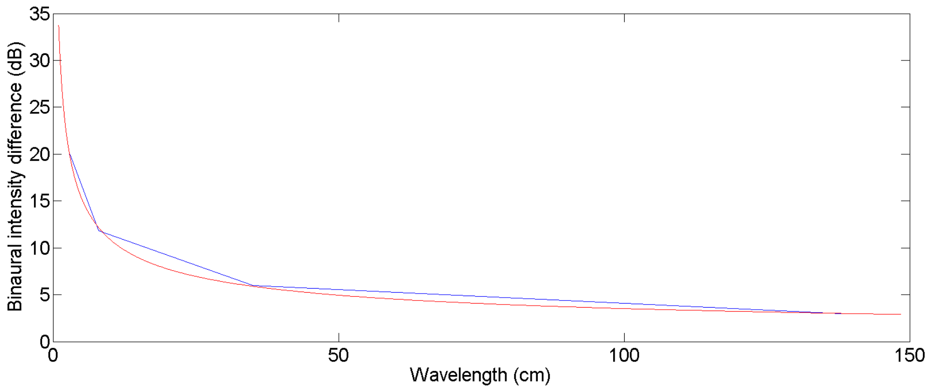


FIG 4.2 BINAURAL INTENSITY DIFFERENCE IN DECIBELS FOR PURE TONE AT 90 AZIMUTH AS A FUNCTION OF WAVELENGTH IN CENTIMETERS (AFTER GULICK, 1971).

Example: produce an estimate of the interaural intensity difference (IID) at 2 kHz, at 60 degrees azimuth.

Now we've recreated these two figures successfully, the task now is it determine how to calculate and obtain the IID data we want from these figures. We can perform a simple example, obtaining the IID for a specific frequency: our task is to produce an estimate of the interaural intensity difference at 2 kHz, at

60 degrees azimuth.

First we should work out the wavelength at 2 kHz. The equation of wavelength is

$$wavelength(w) = \frac{the\ speed\ of\ sound\ (m/s)}{frequency\ (Hz)}$$

We can define the speed of sound as 340 m/s, so the wavelength will be 340/frequency= 340/2000= 0.17= 17 cm. Then, to find the value 17 from the value of 'xx' (from Fig 4.4 recreation), xx= 568, and exp(d) is the intensity difference at 2000 Hz at 90 azimuth. exp(d)= 8.3852.

```
d=polyval(p,xx(568)) (find the IID of 2000 Hz at 90 azimuth from Fig 4.4)

exp(d)
```

Next step, we create a new curve at 2000 Hz

```
az=[0,90,180];              (azimuth data: 0, 90, 180)

in=[0,-8.3852,0]            (intensity difference data:0, -8.3852, 0)

p=polyfit(az,in,2)

x=linspace(0,180,1000);

y=polyval(p,x)

plot(x,y)                   (plot the image)
```

Then we can the 2 kHz curve (Fig 4.3).

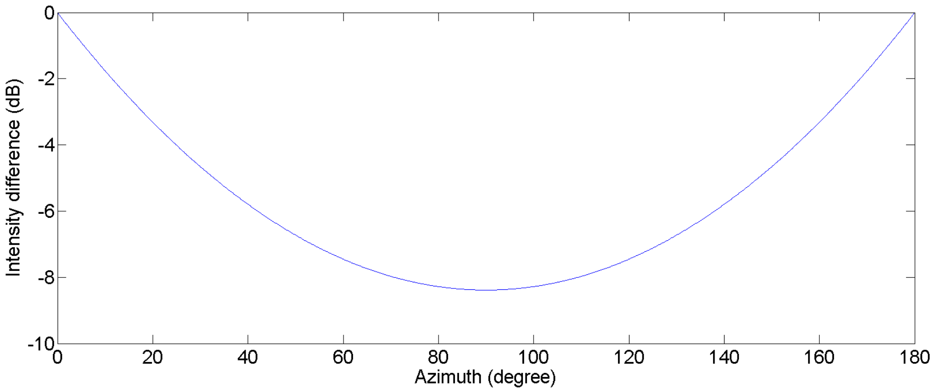


FIG 4.3 THE DIFFERENCES IN INTENSITY (IID) IN DECIBELS AT THE EARS FOR PURE TONES OF 2000 HZ AT 90 AZIMUTHS

The final step is to work out the intensity difference at 60 degrees from the 2 kHz curve.

```
find(x==60)

>> e=polyval(p,x(334))
```

So, at 2 kHz and 60 degrees, the interaural intensity difference (IID) is 7.4535 dB.

4.2.2 Create a Matlab function: 'interaural'

As the previous section, 4.2.1.1, described, we now have the ability to calculate the IID for any frequencies, and any azimuth from 0 to 180. The method to calculate this is detailed in section 2.2.2. We assume that the sound source is coming from right side of the listener's head, and the distance from the sound source to the right ear is 2 metres. So the distance D_r is 2, and the extra distance the sound source must travel to the left ear is $D = k\ (\beta + sin\beta)$ (Gulick, 1971). Where k is the radius of the head (8.75 cm) and the angle β represents the angle from the sound source to the ear, in radians. So the total distance from the sound source to the left ear D_l= 2 (original distance)+ $k\ (\beta + sin\beta)$ (extra distance). The equation to calculate ITD is:

$$Interaural\ Time\ Difference\ (s) = \frac{distance\ (D)}{speed\ of\ sound\ (m/s)}$$

So the final equation to calculate the ITD is:

$$ITD\ (ms) = \frac{0.0875 * (\beta + sin\ (\beta))}{340} * 1000$$

The purpose of creating the 'interaural' function is to provide a function, which can be conveniently invoked to calculate the IID and ITD data of any frequencies at 0-180 azimuth, for both ears.

Functions description:

Function: [x,y] = interaural (a,b)

This function is to calculate the IID (Interaural Intensity Different), and ITD (Interaural Time Difference) of any frequency and any azimuth for two ears.

[x,y] which represent the output IID(x)(dB) and ITD(y)(ms).

(a,b) which represent the input frequency(a) and angle from which the sound source travels to the ear(b).

To calculate the ITD, we use the equation $(r*(\theta+\sin(\theta)))/340*1000$

Example: calculate the IID and ITD at 2 kHz and 60 degree.

[x,y]=interaural(2000,60)

x(IID)= 7.4704 dB

y(ITD)= 0.2709 ms

Functions code:

```
xx= [3,8,35,138];          (Recreate Gulick's figure by utilising polynomial)

yy = [20,11.8,6,3];

p = polyfit(log(xx),log(yy),1);

xxx = linspace(0,5,1000);

yyy = polyval(p,xxx);

c = 340/a*100;

d = polyval(p,log(c));

d = exp(d);

az = [0,90,180];               (calculate the IID)

in = [0,d,0];
```

```
p2 = polyfit(az,in,2);

x = polyval(p2,b)

e = 0.0875*(degtorad(b)+sin(degtorad(b))) (calculate the ITD)

y = e/340*1000
```

4.2.3 Create a Matlab function: 'ampipd'

The previously defined function, 'interaural', provides a method to calculate the IID and ITD of a pair of ears. The IID and ITD represent the intensity difference and time difference between two ears:

$$IID = I_l(intensity\ of\ sound\ traveling\ to\ the\ left\ ear) - I_r\ (intensity\ of\ sound\ traveling\ to\ the\ right\ ear)$$

$$ITD = T_l\ (time\ of\ sound\ traveling\ to\ the\ left\ ear) - T_r\ (time\ of\ sound\ traveling\ to\ the\ right\ ear)$$

So, is there any method to calculate the individual amplitude and time for a given ear? The answer is, of course, yes. In section 2.2.2, the inverse square law is mentioned, and this is the equation we want to calculate the sound intensity at one ear. Sound intensity as a function of distance from a sound source is given by:

$$I = \frac{W_{source}}{A_{sphere}} = \frac{W_{source}}{4\pi r^2}$$

Where I = the sound intensity

W_{source} = the power of the source

and r = the distance from the source (Howard, 1998)

We assume that the amplitude from the sound source is W_{source}=1 dB, the distance from to sound source to our ear is our previously defined default r=2 metres (the motivation for a value of two metres is that this is usually the distance that speakers are set up from the listener), the equation of amplitude change to intensity is $I = a^2$. So the equation could be:

$$I = \frac{1^2}{4\pi(2^2)} = \frac{1}{16\pi}$$

This equation can help us to calculate the intensity of sound traveling to the nearest ear. The relationship of the sound source going to the nearest ear, and the farthest ear is shown in Fig 4.4. When we have the intensity of the nearest ear (I_n), we already have the method needed to calculate the IID between two ears. The intensity of the farthest ear (I_f) can be worked out as $I_n - IID$.

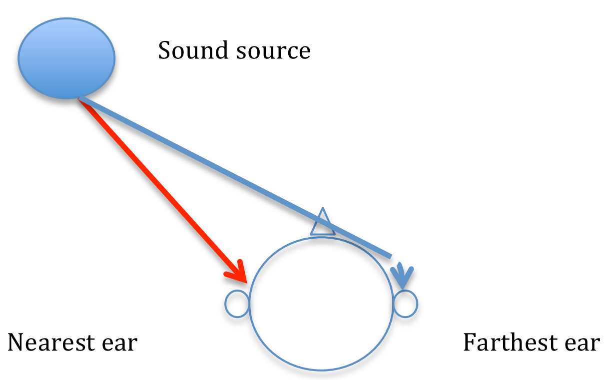


FIG 4.4 THE RELATIONSHIP OF THE SOUND SOURCE GOING TO THE NEAREST EAR AND FARTHEST EAR. THE INTENSITY AND THE PHASE OF THE NEAREST EAR IS $I_n,\ p_n$. THE INTENSITY AND THE PHASE OF THE FARTHEST EAR IS $I_f,\ p_f$.

In section 4.2.2, we already defined the method to calculate the individual time for the nearest and farthest by using the equation:

$$Interaural\ Time\ Difference\ (s) = \frac{distance\ (m)}{speed\ of\ sound\ (m/s)}$$

The distance from the source to the nearest ear is, in this case, our default of 2 m. The distance from the source to the farthest ear is 2 + $k\ (\beta + sin\beta)$ m. After obtaining these distances, we can calculate the individual time at each ear. Because in the end we want to create a 180 degree phase shift or a close approximation of that, we are more concerned with the phase difference than the time difference. The equation of ITD change to IPD is:

$IPD = \frac{frequency*360*ITD}{1000}$ (because the ITD is major with ms, the change to IPD must be secondary)

The phase at the nearest ear will be:

$$P_n = \frac{distance\ (m)}{speed\ of\ sound\ (m/s)} * frequency\ (Hz) * 360 = \frac{170}{f} * 360$$

The phase at the farthest ear will be:

$$P_f = \frac{170}{f} * 360 + \frac{frequency * 360 * ITD}{1000}$$

Function description:

[an,af,ipd,pn,pf] = ampipd(a,b)

This function's purpose is to calculate the two amplitudes of each ear from one speaker.

The IPD(Interaural Phase Difference) between two speakers.

The individual phase for each ears.

[an,af,ipd,pn,pf] which represent the output.

'an' is the amplitude of the nearest ear.

'af' is the amplitude of the farthest ear.

'ipd' is the IPD(degree)

'pn' is the phase of the nearest ear.

'pf' is the phase of the farthest ear.

(a,b) which represent the input amplitude(a)and frequency(b).

IPD = 360*frequency*ITD

Example: calculate the amplitude at 1 for two ears and IPD for 1 kHz.

[an,af,ipd,pn,pf]=ampipd(1,1000)

an = 0.1410

af = 0.0962

IPD = 97.5429

pn = 61.2000

pf = 158.7429

Function code:

```
an=(a.^2)/(16*pi);          (calculate the individual amplitude at ear)

[c,itd]=interaural(b,30);

f=10^(c/10);

af=(an/f);

an=sqrt(an);

af=sqrt(af);

ipd=360*b*itd/1000;         (calculate the IPD)

pn=(170/b)*360;             (calculate the individual phase at ear)

pf=(170/b)*360+(360*b*itd/1000);
```

4.2.4 Create a Matlab function: 'ampphs'

Using the function 'ampipd', we can obtain the amplitude and phase for both the nearest and farthest ear. However, the function 'ampipd' is defined only to calculate the amplitude and phase of one sound source. In a situation with two sound sources (a very normal situation is listening with two speakers), we are not only concerned about the signal from one speaker arriving at the ears, but the other speaker's signal too. In a stereo listening environment, the sound sources play back via two speakers simultaneously. The left speaker signal arrives first at our left ear (nearest ear); the amplitude and phase at the left ear can be calculated by using 'ampipd'. The listening situation with two loudspeakers is shown in Fig 4.5. We define the amplitude from the left speaker coming to the left ear as 'a_1', and the phase from the left speaker coming to the left ear is defined as 'pn_1'. We then calculate the amplitude from the left speaker coming to the right ear (farthest ear) 'a_2', and the phase from the left speaker coming to the right ear is defined as 'pf_1' ('n' indicates near and 'f' indicates far). Considering stereo playback, we also need to calculate the amplitude from the right speaker coming to the right ear (nearest ear) 'b_1', the phase from the right speaker coming to the right ear 'pn_2', the amplitude from the right speaker coming to the left ear (farthest ear) 'b_2', and lastly, the phase from the right speaker coming to the left ear 'pf_2'.

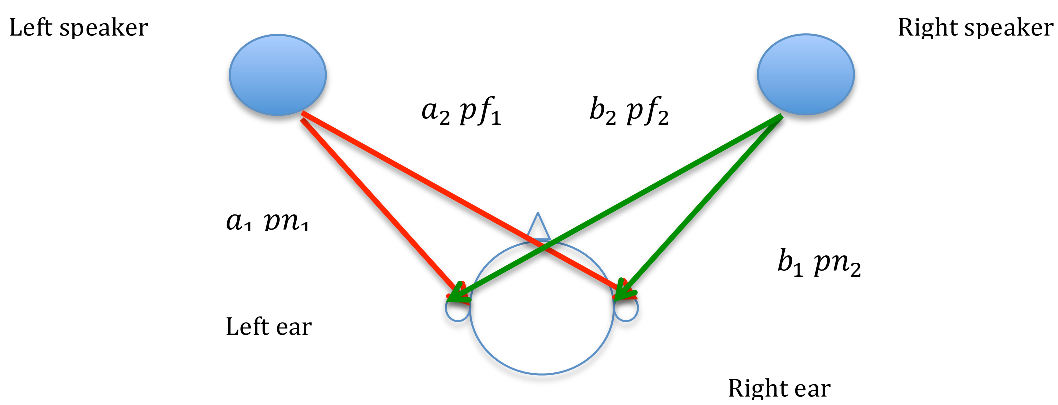


FIG 4.5 THE LISTENING SITUATION WITH TWO LOUDSPEAKERS. THE AMPLITUDE FROM THE LEFT SPEAKER

COMING TO THE LEFT EAR IS 'a_1', AND THE PHASE FROM THE LEFT SPEAKER COMING TO THE LEFT EAR IS DEFINED AS 'pn_1'. THE AMPLITUDE FROM THE LEFT SPEAKER COMING TO THE RIGHT EAR (FARTHEST EAR) IS 'a_2', THE PHASE FROM THE LEFT SPEAKER COMING TO THE RIGHT EAR IS DEFINED AS 'pf_1'. THE AMPLITUDE FROM THE RIGHT SPEAKER COMING TO THE RIGHT EAR (NEAREST EAR) IS 'b_1', THE PHASE FROM THE RIGHT SPEAKER COMING TO THE RIGHT EAR IS 'pn_2'. THE AMPLITUDE FROM THE RIGHT SPEAKER COMING TO THE LEFT EAR (FARTHEST EAR) IS 'b_2', AND THE PHASE FROM THE RIGHT SPEAKER COMING TO THE LEFT EAR IS 'pf_2'.

The purpose of the function 'ampphs' is to calculate the amplitude combination and phase combination for one ear, when listening with two speakers. The equation for calculating the amplitude combination can be expressed as:

$$AC\ (amplitude\ combination)$$
$$= \sqrt{(cos(pn) * an + cos(pf) * af)^2 + (sin(pn) * an + sin(pf) * af)^2}$$

Where $pn = the\ phase\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ nearest\ ear.$

$$pf = the\ phase\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ farthest\ ear.$$

$$an = the\ amplitu\text{?}e\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ nearest\ ear.$$

$$af = the\ amplitude\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ farthest\ ear.$$

* The phase at the ear is originally in degrees. So before using this equation, 'pn' and 'pf' must first be converted from degrees to radians.

Based on this equation, the amplitude combination at the left ear can be calculated by:

$$AC_l = \sqrt{(cos(pn_1) * a_1 + cos(pf_2) * b_2)^2 + (sin(pn_1) * a_1 + sin(pf_2) * b_2)^2}$$

The amplitude combination at the right ear can be calculated by:

$$AC_r = \sqrt{(cos(pn_2) * b_1 + cos(pf_1) * a_2)^2 + (sin(pn_2) * b_1 + sin(pf_1) * a_2)^2}$$

We are not only concerned with the amplitude combination of the two speakers, but must also consider their phase combination. Calculating this phase combination is incredibly important, since we can only adjust the phase shift to 180 degrees after determining the original phase shift of our ear. The equation for calculating the phase combination can be written as:

$$PC\ (phase\ combination) = atan\frac{sin(pn) * an + sin(pf) * af}{cos(pn) * an + cos(pf) * af}$$

Where $pn = the\ phase\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ nearest\ ear.$

$pf = th\ phase\ of\ the\ ear\ when\ this\ ear\ becomes\ the\ farthest\ ear.$

$an = the\ amplitude\ of\ the\ ear\ when\ this\ ear$ becomes the nearest ear.

af = the amplitude of the ear when this ear becomes the farthest ear.

* The phase at the ear is originally in degrees. So before using this equation, 'pn' and 'pf' must first be converted from degrees to radians.

Based on this equation, the phase combination at the left ear can be calculated by:

$$\mathrm{PC_l} = \mathrm{atan}\frac{\sin(\mathrm{pn_1}) * \mathrm{a_1} + \sin(\mathrm{pf_2}) * \mathrm{b_2}}{\cos(\mathrm{pn_1}) * \mathrm{a_1} + \cos(\mathrm{pf_2}) * \mathrm{b_2}}$$

The amplitude combination at the right ear can be calculated by:

$$PC_r = \text{atan}\frac{\sin(pn_2) * b_1 + \sin(pf_1) * a_2}{\cos(pn_2) * b_1 + \cos(pf_1) * a_2}$$

Function description:

[x,y]=ampphs(a,b,pn,pf)

This function calculates the amplitude combination and the phase

combination from two different amplitudes and the IPD.

The inputs a and b represent the amplitude of the nearest ear and farthest

ear, respectively.

'pn' is the phase of the nearest ear.'pf' is the phase of the farthest ear

The input 'ipd' represents IPD.

These five values can be obtained by using the function 'ampipd'.

[x,y] which represent the output.

'x' is the amplitude combination.

'y' is the phase combination.

Example: calculate the amplitude combination and phase combination of an: 0.1410, af: 0.0962,

pn: 61.2000 and pf:158.7429

[x,y]=ampphs(0.141,0.0962,61.2,   158.7492)

x= 0.1083

y= 71.7319

Function's code:

```
d=degtorad(pn);        (convert from degrees to radians)

e=degtorad(pf);

x=sqrt((cos(d)*a+cos(e)*b).^2+(sin(d)*a+sin(e)*b).^2);

(calculate the amplitude combination)

yy=atan((sin(d)*a+sin(e)*b)./((cos(d)*a+cos(e)*b)));

(calculate the phase combination)

y=radtodeg(yy);
```

4.3 Creating the 'out of phase' situation

Usually, when the sound tracks are panned to the centre, and output via two loudspeakers, the phase of the left ear and the phase of the right ear should be the same. That means that the phase combination difference of two ears is 0 degrees. We can use the function I created to calculate in this situation.

For example, the signal now is panned to the centre, and the angle from the speaker to our head is 30 degrees. Next, we want to calculate the phase combination difference at the ear, at 1000 Hz:

```
degtorad(45);    (panning the sound to the centre)

x=ans;

y=cos(x);

y2=sin(x);

[iid,itd]=interaural(1000,30);            (calculate the IID and ITD)

[a1,a2,ipd,pn1,pf1]=ampipd(y,1000);    (calculate the an, af, pn, pf for one speaker)

[b1,b2,ipd2,pn2,pf2]=ampipd(y2,1000); (calculate the an, af, pn, pf for the other speaker)

[amp1,phs1]=ampphs(a1,b2,pn1,pf2);       (the amplitude combination and phase combination
for the left ear)

[amp2,phs2]=ampphs(b1,a2,pn2,pf1);       (the amplitude combination and phase combination
for the right ear)

phs1                                      (phase combination of left ear)

phs2                                      (phase combination of right ear)
```

The result claims that the phase combinations at the left and right ear are both 5.7536 degrees. Calculating the phase combination difference, we come to a value of 5.7536-5.7536=0 degrees. If we want to create the ‘out of phase’ situation, we can rotate by 180 degrees at the right speaker (usually in an audio workstation; we invert the signal at the right speaker), to create an ‘out of phase’ situation.

This example, with all sounds panned centrally, is simple and easy to make 'out of phase'. However, when the sounds are not panned exactly to the center, the IID and ITD will not be the same. This means that the phase combination values for each ear might not be the same. Fig 4.6 analyses two unique sound source position situations. Situation (a): the sound source is coming from the middle, and the intensity and phase of two ears is the same. Situation (b): the sound source is panned to the left by 30 degrees, and the intensity and phase of two ears is not the same due to the IID and ITD. In this situation, what phase shift should we undertake to achieve a 180 degree phase shift? In addition, different frequency profiles could alter the phase difference. We must keep the phase shift at 180 degree at all frequencies.

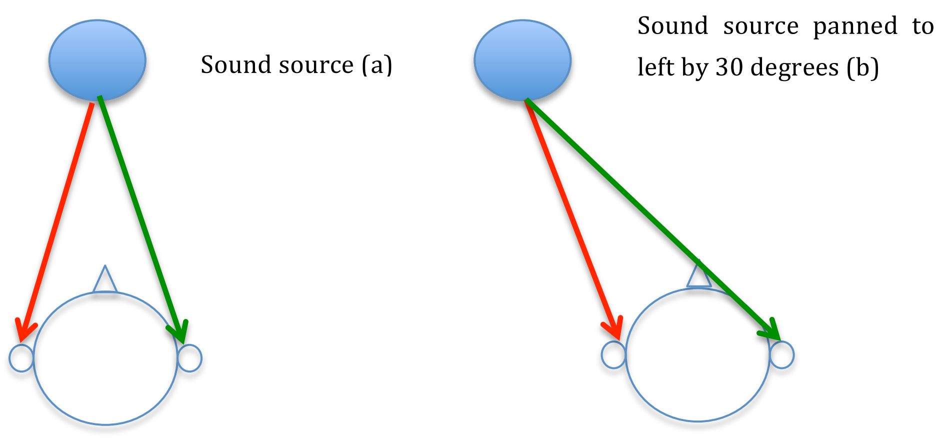


FIG 4.6 TWO DIFFERENT SOUND SOURCE POSITION SITUATIONS. SITUATION (A): THE SOUND SOURCE IS COMING FROM THE MIDDLE, AND THE INTENSITY AND PHASE OF TWO EARS IS THE SAME. SITUATION (B): THE SOUND SOURCE IS PANNED TO LEFT BY 30 DEGREES, AND THE INTENSITY AND PHASE OF TWO EARS IS NOT THE SAME.

The method required to determine the answer, is to calculate the minimum value of the 180 phase shift, minus the phase combination difference. Then we use this value to determine the exact phase shift we should add, in the range of 0-360 degrees. To accommodate for different frequency profiles, I will use a function 'thirdOctaveFilterBankGenerate' (Wells, 2015). This function will create 29 third octave filter banks from 0 Hz to 20159 Hz.

For example, at panning position 30 degrees to the left, we want to find out the phase shift value we should add for these 29 filter banks.

```
[filterBank,p] = thirdOctaveFilterBankGenerate(44100,10);

hold off                (firstly set up the filter banks)

degtorad(30);       (sound position panned 30 degrees to the left)

x=ans;

y=cos(x);

y2=sin(x);

for n=(1:length(p));



[iid,itd]=interaural(p(n),30);         (the frequency range is the range of filter banks)

[a1,a2,ipd,pn1,pf1]=ampipd(y,p(n));

[b1,b2,ipd2,pn2,pf2]=ampipd(y2,p(n));

z=linspace(0,360,1000);         (add a 0 to 360 degree phase shift to the right signal)

pf2=pf2+z;

pn2=pn2+z;

[amp1,phs1]=ampphs(a1,b2,pn1,pf2);
```

```
[amp2,phs2]=ampphs(b1,a2,pn2,pf1);

phs0=phs1-phs2;          (phs0 is the phase combination difference for two ears at the 30
degrees left position)

phs0=phs0-phs0(1);

phs0(find(phs0<-180))= phs0(find(phs0<-180))+360; (set up the phase shift range from -180 to
180)

phs0(find(phs0>180))= phs0(find(phs0>180))-360;

[m,i]=min(abs(180)-abs(phs0));    (find out the minimum value of 180 degree phase shift minus
phase difference)

pd=z(i);          (find out the value of phase shift we should at to the right output
signal)

td(n)=pd/360/p(n)*1000;

end

>> for n=(1:length(p));

pd(n)=td(n)/1000*360*p(n);

end

for n=(1:length(p));

pd(n)=td(n)/1000*360*p(n);   ('pd' is the phase shift we should add to the right signal for 29
filter banks' frequencies)

end
```

After the calculation, we are presented with a result: how many degrees we should shift the right signal to obtain the 180 degree phase shift. If we type ‘pd’ into Matlab, we can view the 29 filter banks’ phase.

This is the phase we should add to the right signal for these 29 frequencies. If we want to know a phase shift for a particularly frequency band, for example, the phase shift we should implement at 1000 Hz, we must first find out which filter bank represents 1000 Hz. So we can run ’find(p==1000)’, and the returned index is 16. We can then retrieve the 16th index of ‘pd’: pd(16)= 228.4685. So at 1000 Hz, 30 degrees to the left, we should add an 228.4685 degree phase shift, to create a 180 degree ‘out of phase’ situation.

When we export these data and create a Max/MSP patch, which can create a 29 filter banks and load up the delay with these data, we should convert the phase shift and the time shift, so that we can work out exactly how many seconds of delay each filter bank requires. As previous mentioned, the equation of time conversion to phase is:

$$\mathrm{P} = \frac{\text{distance (m)}}{\text{speed of sound (m/s)}} * \text{frequency (Hz)} * 360$$

So the time delay (td) can be converted like:

$$\mathrm{td} = \frac{\mathrm{pd}}{360} \Big/ \text{frequency} * 1000$$ (we want to view the time delay in milliseconds, so we multiply by 1000)

Finally, the ‘td’ shows the delay time values we should use at the 30 degrees left position, in order to create the ‘out of phase’ situation.

So at the 30 degrees left position, the 16th filter bank (1000 Hz) should add 0.6346 ms delay time to achieve the 180 degree 'out of phase' situation.

4.4 The phase shift error

In section 4.3, we were able to create the 180 degree 'out of phase' situation for every azimuth. That means if we pan the sound to a certain angle, the phase difference between the left ear and right ear could potentially always be 180 degrees, and this 180 degree phase shift can reduce the effects of masking by the maximum possible amount. Theoretically, I've calculated the correct delay time for each filter bank, and imported the delay data to Max/MSP; the result of the output could be a 180 degree phase shift in every filter bank. However, some discordance is witnessed between the 180 degree case and the actual phase shift Max/MSP reads.

We apply an impulse to the input of the patch, the impulse goes to 29 filter banks separately and are summed together at the output stage. After that, the signal is exported from the MaxMSP patch to be analysed in Matlab. For each of the 29 filter banks' frequencies, the phase shifts both have some degree of error, compared to 180 degrees. We can use Matlab to find out the difference. Simply using 'find' function, finding the minimum value of 180 minus the actual phase shift, we can arrive at Fig 4.7. The table shows the phase error of each filter bank.

Filter bank's frequency (Hz)	Phase error (x°)	Filter bank's frequency (Hz)	Phase error (x°)
31	-175.8463	1000	24.1765
39	-54.2675	1260	104.8109
50	29.318	1587	41.6878
63	3.867	2000	48.5696
79	-105.3406	2520	-148.4453
99	25.3258	3175	85.3777
125	2.944	4000	99.191
157	147.9769	5040	62.8552
198	50.2045	6350	171.8283
250	6.0403	8000	-146.3237
315	-63.7826	10079	-58.7891
397	100.2824	12699	154.0352
500	12.0968	16000	-148.3584
630	-126.9956	20159	61.8775
794	-158.6650		

FIG 4.7 THE PHASE ERROR OF EVERY FILTER BANKS.

The reason for the phase error does not seem to have a pattern. It is unclear what factor exactly might cause the phase error. Theoretically, the signal, which is imported to Matlab, should reflect the 180 degree phase shift or a phase shift very close to 180 degrees. It could be a problem with the delay object or delay operation in Max/MSP. The size of the filter banks is 1 by default. I've increased the filter banks' size to 10 (an increase in size can improve the quality of every filter bank. When the size of the filter banks becomes smaller, the peak filter bank will not arrive at 0 dB). The result I obtained still produces an error for every filter bank. A comparison between the phase shift error for size 1 and size 10 filter banks is shown in Fig 4.8. It doesn't seems the 10 size filter banks provides a significant error reduction, in comparison with the 1 size filter banks.

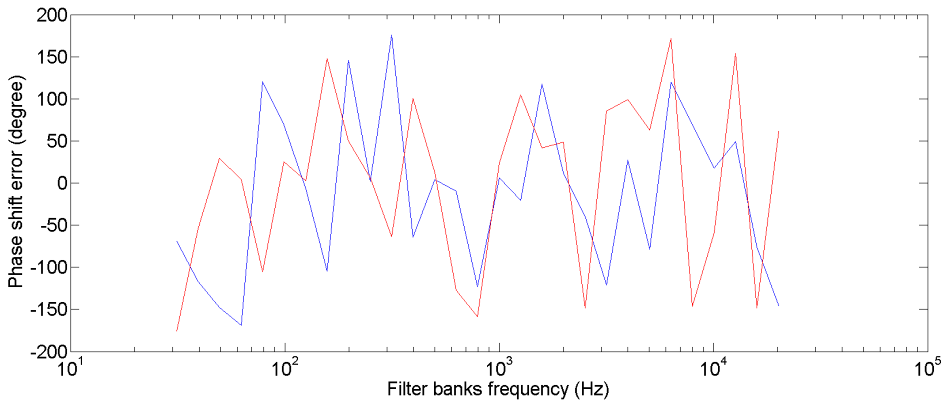


FIG 4.8 THE COMPARISON OF 1 SIZE FILTER BANKS AND 10 SIZE FILTER BANKS' PHASE SHIFT ERROR IN LOGARITHM.

Blue line: 1x filter bank. Red line: 10x filter bank

So the results obtained from the Max/MSP patch are not perfect at achieving 180 degrees of phase shift for every frequency band. I will make finding a solution to this error a primary target in the 'future work' section. The error may be reduced when the phase shift changes from time domain to Fourier domain.

5 Music outcomes

5.1 Some examples from musicians who are interested in different approaches to mixing (homogenous and individualistic)

Phil Spector's Wall of Sound (homogenous)

Phillip Spector (1939) is an American musician, songwriter and record producer and is the creator of the 'Wall of Sound' recording technique. Spector's studio group often consists of various selections of musicians and instruments, such as three drummers, bassists, keyboard players, and numerous guitarists with the occasional use of a string orchestra and brass section. His recordings use large bombastic, reverberation instruments which constantly threatened to drown out vocals. His recording groups are large and have ranges of instrument. Spector's work belongs to a homogenous mixing style. Famous songs and compositions such as the song by The Crystals, 'Da doo ron ron' (1963) and the song by the Ronettes, 'Be My Baby' (1963) are two good examples of Spector's 'wall of sound'.

Bob Clearmountain and Trevor Charles Horn (individualistic)

Bob Clearmountain, an American music engineer, mixer and producer and Trevor Charles Horn (1949) who is an English pop music record producer, songwriter, musician and singer, focus both of their mixing techniques on individual elements. For example, 'Try' – a Bob Clearmountain mix, (2013) by Noah Francis John is a version of a remix by Bob. The whole song is clearly recorded and every instruments including guitar, strings, drum and vocals can be heard. Bob utilised few instruments, and made every instrument clearly audible, so that the listener could clearly identify each of the sounds and instruments. 'Sh Boom' (2003) mixed by Trevor Horn is a song of OST in film 'Mona Lisa Smile' (2003). The mixing style is very similar to that of Bob's in that they both recorded a limited range of instruments. In this song, Bob mixed jazz with brass, drum and vocals, and was concerned about the clarity of the individual elements.

The main difference to that of Phillip Spector is, Bob and Trevor are primarily interested with emphasising individual elements, so they tend to mix using a limited range of instruments and improving the clarity of each sound rather than working with a choice of large bombastic, reverberation instruments which would constantly threatened to drown out vocals. Both homogenous and individualistic styles have advantages and disadvantages; orchestration is about creating new combinations from many components as well as creating individual components which can be heard separately. Although the clarity of individual components is not always the prime concern, this research is concerned with the sound clarity audibility of each instrument.

5.2 elevation filter test

Firstly, I've created a simple demo, which consists of a piano and a guitar, both playing the same melody and both panned centrally. The purpose of this is to test the elevation filter, not only to test if the masking level is reduced is comparison with the original version, but also to discover if the audience can identify the height of a sound relative to the original output. The difference between the elevation filter and VSP from Studer, is that VSP's HRTF panning mode exists only to allow mixing engineers to pan the sound in the azimuth area. Although VSP incorporates HRTF data for panning, the HRTF data cannot be used to pan with elevation. The elevation filter is calculated from elevation magnitude difference of two filters, and then combined for a new elevation difference filter. When the sound track is run through the elevation difference filter, it is affected by a HRTF with height data, and as such, the listener can identify that the sound is at a specific height position. That is what the elevation filter can achieve, and VSP can not.

As the last paragraph stated, the original output is a mono piano and a mono guitar playing the same melody, both panned to the centre. Because the melody of the piano and the guitar are exactly the same and the spectral content of the sounds is largely similar, they may cover each other, and as such the masking problem may occur.

The first test is to elevate the guitar track by +45, keeping all other elements at their original positions. When this is done, the mix should be subjected to careful listening to see if the piano and guitar are significantly more discernable than they previously were. Theoretically, these two melody tracks should be more likely to be audible because the magnitude responses of 0 and +45 elevation, respectively, are different (Fig 5.1).

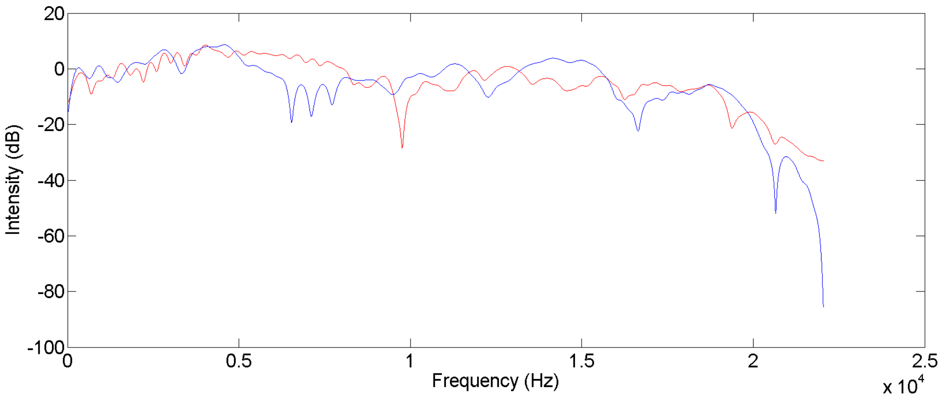


FIG 5.1 THE MAGNITUDE RESPONSE OF 0 ELEVATION HRTFS (BLUE LINE) AND 45 ELEVATION HRTFS (RED LINE).

The other important point of testing is: we use 0 elevation HRTFs subtracted from +45 elevation HRTFs, and create a 45 degree elevation difference filter (Fig 5.2). If we run the guitar track through this filter, it should be perceived 45 degrees above the piano. In accordance with the pinna effect, our ear can identify the sound's origin, in terms of up and down, front and rear, and the HRTF data helps us to capture the height information. So our ear can obtain this height information via the elevation filter, and the audience can feel that the guitar sound is in a position 45 degrees above the piano.

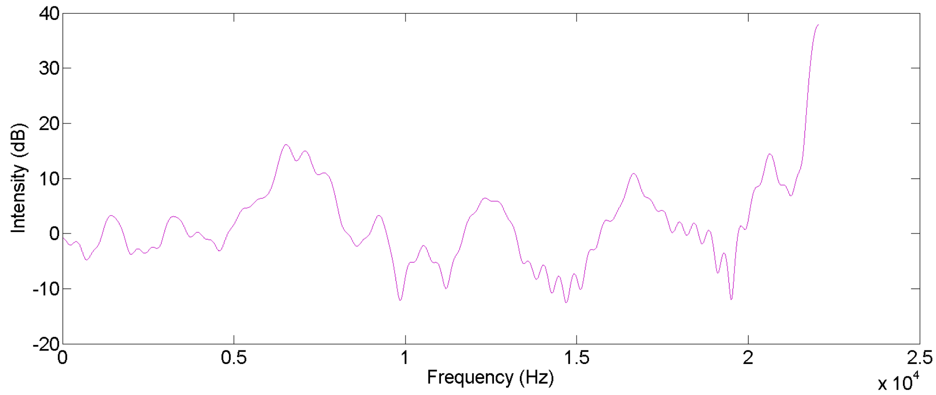


FIG 5.2 THE MAGNITUDE RESPONSE OF THE 45 DEGREE ELEVATION DIFFERENCE FILTER.

The second test is to add a -45 elevation filter to the guitar track to see if the guitar is perceived 45 degrees below the piano. Fig 5.3 shows the magnitude response of the 0 elevation HRTF and the -45 elevation HRTF. Fig 5.4 shows the magnitude response of the -45 degree elevation difference filter. This time, our ear can feel that the guitar sound is coming from a source 45 degrees lower than the original position.

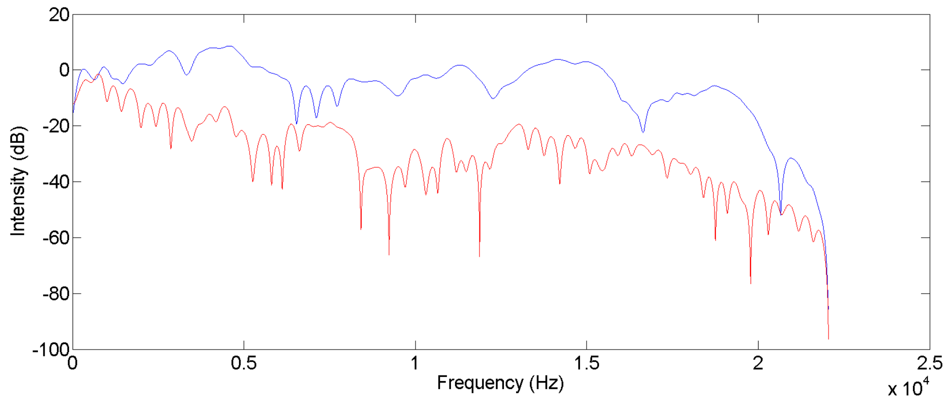


FIG 5.3 THE MAGNITUDE RESPONSE OF 0 ELEVATION HRTFS (BLUE LINE) AND -45 ELEVATION HRTFS (RED LINE).

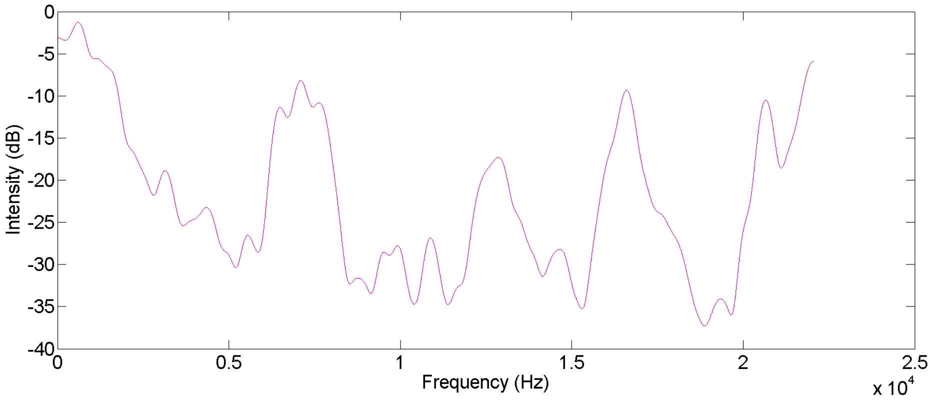


FIG 5.4 THE MAGNITUDE RESPONSE OF THE -45 DEGREE ELEVATION DIFFERENCE FILTER.

The test audio will be provided in the appendix section.

5.3 Max/MSP phase shift test

In the Max/MSP patch, we must firstly store the 29 filter banks in the buffer. We must then perform the convolution, and load up the 29 filter banks. We can use the object ‘multiconvolve’ to utilise these 29 filter banks. The principle of the Max/Msp patch is explained in Fig 5.5.

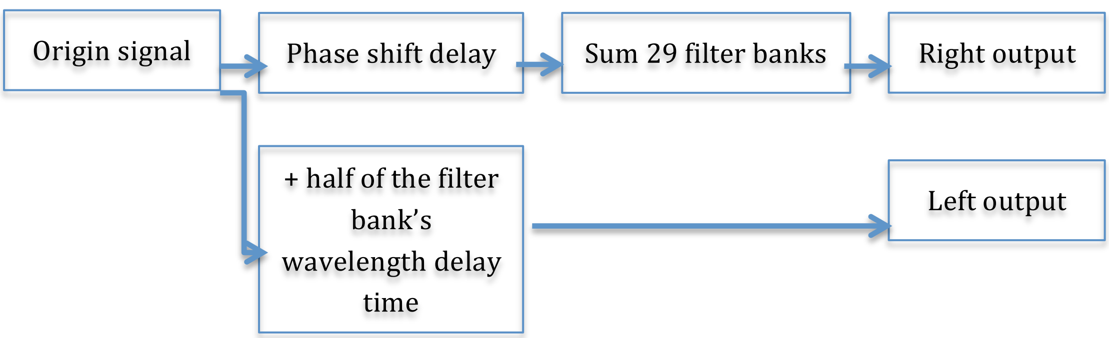


FIG 5.5 THE PROCESSING FLOW CHART OF THE MAX/MSP PATCH

The flow of the patch is: the signal we want to create the phase shift on is firstly exported from Logic Pro. After this step, the signal is split to two inlets, one going to 29 filter banks, with every filter bank adjusted by delay time, which calculated via Matlab. The other goes to a single delay. This delay time is half of the filter bank's wavelength. After that, the signal with half of the filter bank's wavelength delay is sent directly to the left output. The other signal, which goes through 29 filter banks, will be summed to one sign again, given the delay information, and then sent to the right output. Finally, the right ear is shifted 180 degrees out of phase, relative to the left ear (Fig 5.6).

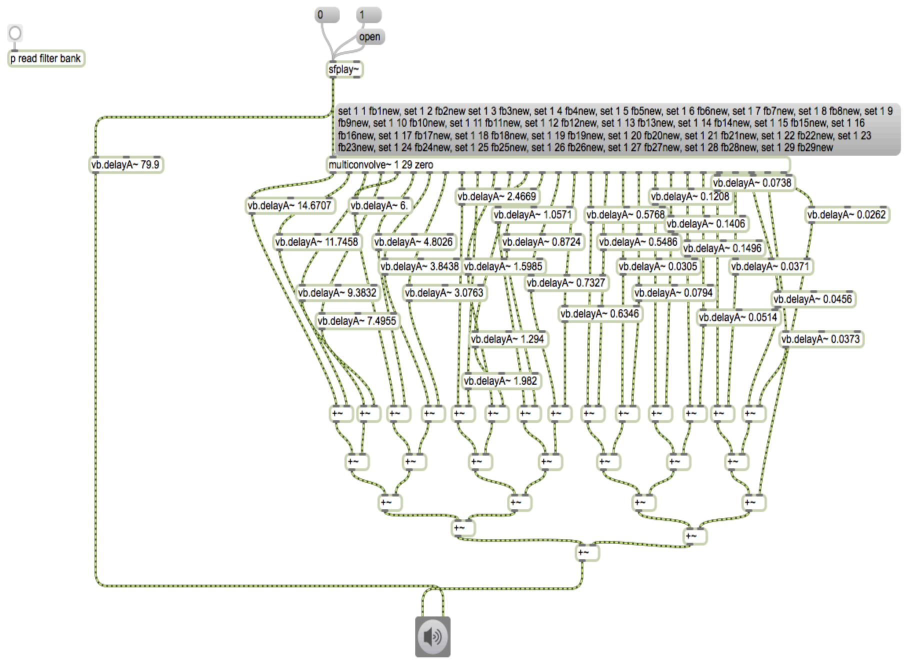


FIG 5.6 THE MAX/MSP PATCH

The test the Max/MSP patch, we want to test these five comparisons:

Using the demo where the piano and the guitar are doubling each other; turn off reverberation, then make the following comparisons:

1. both sounds panned centrally, no phase shift.

2. both sounds panned centrally, but one of them presented with the phase inverted in one channel.

3. both sounds panned to the left by 30 degrees, no phase shift

4. both sounds panned to the left by 30 degrees, one with phase inverted.

5. both sounds panned to the left by 30 degrees, one processed by the Max/MSP patch.

The demo we used is the same demo seen in the elevation filter test. In the first situation, the piano and guitar are both panned centrally with no phase shift. That is the original situation, and because the piano and guitar both play the same melody, plus the instruments' frequency profiles are very similar,, they will mask each other and not be particularly distinguishable for the audience. In the second situation, the piano and guitar are still panned centrally. However, this time we inverted the phase of the guitar. So the left output of the guitar is playing in phase, and the right output of the guitar is out of phase. Because both sounds are panned to the centre, the phase difference of both ears is 0. The second case is in a 'perfect' out of phase situation. The sound of what our right ear is hearing is 180 degrees inverted from what our left ear hears. Thus, the masking will be released by the maximum possible amount. Case 3: we panned both sounds 30 degrees to the left. This means that the sound position is not in the middle. What is more, the IID and ITD of our ears are changed. Because the sound source is now coming from 30 degrees on the left, the sound will arrive first in our left ear, and our left ear will witness a higher intensity compared to the right ear. The phase difference is not going to be 0 any more. The third situation still doesn't invert so the piano and guitar are still masked by each other.

The most important and interesting point is to compare situation 4 and situation 5. Situation 4 sees both sounds panned 30 degrees to the left, and the phase of the guitar inverted. Situation 5 has both sounds panned 30 degrees to the left, and the guitar is processed by the MaxMSP patch. Which situation will be more audible, and in which situation will the masking problem be best dealt with? Digital audio workstations including Logic Pro, Cubase and Pro Tools etc. and lots of mixing consoles such as Euphonix and Studer systems, offer an 'invert phase' function. It can directly inverse the phase of one channel without any delay. This method can provide the maximum release from masking when the sounds are panned centrally. But when mixing, panning sound is a very normal step taken to create the illusion of space, and when we pan the sound, the IID, ITD and phase difference of our ear are also changed. At this point, the phase that our ear receives is the phase difference between two ears, plus the phase of the sound coming from the speaker (the inverted sound track doesn't have delay, so the phase should be 0). Because of the phase difference at our ears, the phase shift at our ears is not 180 degrees. The masking release will not be optimal. Looking at the fifth situation, the actual phase shift from which the sound source arrives at our ear, is the phase difference of two ears, plus the delay time we calculate from Matlab. It is not a 180 degree phase shift (in section 4.4 the phase shift error has already been covered), but the phase shift is actually changed and is close to 180 degrees. If the sound source of the phase shift processing could be improved in the future, the guitar could play backed with the perfect 'out of phase' situation, and the piano and guitar would both be entirely, perfectly audible because the masking would be released by the maximum magnitude. Finally, comparing situations four and five, theoretically, situation 5 will result in a greater release of masking, and the sounds will be more easily discernable.

These five comparisons are provided in the appendix section.

5.4 Composition

My final composition includes 29 sound tracks. These consist of background parts: chinese guzheng, violin, viola etc. There is also a melodic/harmonic component: piano, guitar etc. Additionally, rhythm and bass: Asian kit, timpani etc. Some sound effects: chinese xiao, house remix parts (synthesizer plugins) etc. The main melody: piano and chinese dizi.

This piece is a Chinese style musical composition. As such, I've use a lot of eastern digital instruments such as the chinese guzheng, chinese erhu, chinese xiao and Asian kit etc. In addition, I've added some stringed instruments such as violin, viola, and cello to fill out the background. Actually, the spectral content of erhu and violin, flute and chinese xiao, guzheng and marimba are very close to each other respectively. Stylistically speaking, putting them in the same piece will not cause them to conflict with each other, in terms of the coherence and the musical style. The verse is divided into two parts, and the second is a repetition of the first part. I decided to pan the main melody to the left by 30 degrees in the first part, and pan the main melody to the right by 30 degrees in the second part. The reason why I doing this, first of all, panning the main melody and leaving the harmony and the background central, so the melody will become audible in different positions. Secondly, I decided to inverse the dizi's phase, using the inversion function that Logic Pro provides, also using my Max/MSP patch to create the 180 degree phase shift. The intent here is to alleviate the masking on the main melodic components (piano and dizi). What is more, I can also compare these two methods with regards to release of masking in the context of a sizeable musical work. In the verse part, the audience can not only identify the melody and the harmony clearly, but also distinguish the main melodies of the piano and dizi. In the chorus part, this time I decided to pan all the sound tracks centrally, and added the +45 degree elevation filter to the dizi track. At the same time, I added a -45 degree elevation filter to the drums and bass. The motivation is to release the masking when 29 tracks are playing together in the center. The elevation filter can change the magnitude response of the sound track so that the masking is less of a problem. The listener can hear the piano melody clearly, but at the same time they can identify the dizi melody and recognize that it originates from a higher position. The two melodies do not mask each other at all. What is more, the drum and bass section is thicker than the other sound tracks because of the low frequency content. 'Moving down' the rhythm part can prevent them from clashing with the melody and the background harmony. Additionally, it also helps to prevent masking. The listener can clearly identify the melody, harmony and rhythm clearly and simultaneously.

6 Conclusion and future works

6.1 Conclusion

The purpose of my MA research project was to find ways to make the individual components of mixed music pieces more audible. First of all, we can only feel that the masking situation is less severe when considering low frequency sound. The highest frequency we can feel the masking releasing at is in 2000 Hz. So when the sound frequency is higher than 2000 Hz, we can not perceive any difference if we did some processing to release the masking. The maximum release of masking level occurs at 500 Hz, at which point the masking is released by up to 15 dB (Moore, 1977).

With the elevation filtering element, I attempt to adjust the sound's elevation level via an elevation difference filter in order to increase distinguishability of sounds. In fact, we could feel that the sound is more audible when the elevation filter is applied, because the filter adjusted the EQ of the sound tracks, and the difference in frequency content across tracks acts so that the listener can more clearly identify the filtered sounds. Another effect of the elevation difference filter is that it can move the perceived sound origin to a certain position in height. If the sound is processed by a +45 degree elevation filter, it should be perceived 45 degrees higher by the listener. In fact, when I listening to the testing demo or my final composition, the elevation effect does not appear to be obvious. I can feel that the sound tracks separate better than in comparison to the previous mix, when sound tracks were not using the filter, but I can not really reliably feel that the sound has been elevated. This is a normal result, because I am dealing with a HRTF filter. Actually, the HRTFs of every listener could not possibly be the same. Maybe I can identify the sound position more accurately when I load my own HRTF data into the filter. The angle of the speaker to the listener and the distance between speakers and the listener can also effect the final result. What is more, listening with loudspeakers presents the cross talk problem (unlike listening via head phones, the left speaker signal does not just arrive at the left ear, but also arrives at our right ear, and vice versa, when listening with loudspeakers). This also affects the ability to discern individual sound sources.

For the phase shifting component, I attempted to create a 180 degree 'out of phase' situation for any azimuth value, so that the listener can enjoy the maximum release from masking with sounds coming from any azimuth. Because the 180 degree phase shift can release masking by the maximum possible amount, I've made some functions and done a lot of calculation to work out the phase difference of two ears for every azimuth and every frequency. Theoretically, when I converted the result to time delay, and loaded the delay into the MaxMSP patch, the signal should have been subjected to a 180 degree phase shift in every filter bank. In fact, as section 4.4 mentioned, the phase shift error occurs when I import the phase shifted signal to Matlab. The reason for this

I've also mentioned in section 4.4: it may be associated with the delay part of the MaxMSP patch. The phase shift error may be reduced by applying a STFT (Short Time Fourier Transform) to change the phase shift from time domain to the frequency domain. So, after I listened to the testing demo and my final composition, I can feel that masking is released for certain levels and sounds subjected to my MaxMSP processing are more audible than previously. Even if the phase shift is not a perfect 180 degree phase shift, it can still release the masking. So, I can say that the MaxMSP patch can release the masking by a certain amount but not to the maximum potential.

6.2 Future works

1. If we can measure the listener's HRTFs data, we can better provide the ability to locate sound sources via elevation.

2. The speaker angle and distance may also affect the final result. In this research, I always set the speakers to a 30 degree angle from our ears, and the distance from the speakers to our ear is set to the default of 2 metres. All of the calculations use these data. If we change the speaker positions and the distance of the speakers from the listener, the results should be different.

3. Concerning the phase shift error: it should be reduced if we improve the delay part of the MaxMSP patch. If we apply a STFT (Short Time Fourier Transform) in the MaxMSP patch to change the phase shift from time domain to frequency domain, the result of the phase shift error should be reduced, and the result of the data will be more accurate.

4. We should also consider the cross talk problem when we are playing back with loudspeakers. When the left speaker signal arrives to our right ear as well as the left, and vice versa from the right side, the results of any research relying on them could be inaccurate. I calculated the cross talk signal when I was dealing with phase shift calculation (see section 3.2).

Appendices

CD1 Demo test

Track01 piano and guitar panned centrally with same phase (original)

Track02 piano and guitar panned centrally, guitar added +45 elevation difference filter with same phase

Track03 piano and guitar panned centrally, guitar added -45 elevation difference filter with same phase

Track04 piano and guitar panned centrally, guitar inverted phase

Track05 piano and guitar panned left 30 degrees with same phase

Track06 piano and guitar panned left 30 degrees, guitar inverted phase

Track07 piano and guitar panned left 30 degrees, guitar add Max/MSP patch phase shift processing

CD2 The composition

Track01 The composition

Verse

0:43 original piano and dizi panning to L30 degrees in phase

1:06 original piano and dizi panning to R30 degrees in phase

1:34 piano in phase L30 dizi Max/MSP out of phase

1:59 piano in phase R30 dizi Max/MSP out of phase

Chorus

2:24 chorus both pan to center

3:15 chorus both pan to center, dizi add +45 elevation filter drum add -45 filter

Track02 excerpt from composition: original piano and dizi in phase (without phase shift processing)

Track03 excerpt from composition: piano in phase and dizi Max/MSP out of phase

Track04 excerpt from composition: chorus both pan to center (without elevation filtering)

Track05 excerpt from composition: chorus both pan to center, dizi added +45 elevation filter drum added -45 elevation filter

References

V. Palph Algazi, 1998, Documentation for the UCD HRIR Files, CA 95616

Durand R. Begault, 2000, 3-D Sound for Virtual Reality and Multimedia, NASA/TM-2000-209606

C. Phillip Brown and Richard O. Duda, 1998, A Structural Model for Binaural Sound Synthesis

Crystals, 1963, Music 'Da doo ron ron', Mix by Phillip Spector

Emmanuel Deruty, Francois Pachet, and Pierre Roy, 2014, Human-Made Rock Mixes Feature Tight Relations Between Spectrum and Loudness

Kuhn, G.F. (1987). Acoustics and Measurements Pertaining to Directional Hearing. In *Directional Hearing.* W.A. Yost and G. Gourevitch (Eds)

W. L. Gulick, 1971, HEARING: PHYSIOLOGY AND PSYCHOPHYSICS

David M. Howard, 2009, Acoustics and psychoacoustics, ISBN: 978-0-24052-175-6

Noah Francis John, 2013, Music ‘try’, remix by Bob Clearmountain

Dennis McFadden, Lloyd A. Jeffress, and Jhon R. Lakey, 1972, Differenes of Interaural Phase and Level in Detection and lateralization: 1000 Hz and 2000 Hz

Dennis McFadden and Lloyd A. Jeffress, 1971, Differenes of Interaural Phase and Level in Detection and lateralization

Brian. C. J. Moore, 1977, An Introduction to the Psychology of Hearing, sixth edition, ISBN: 978-1-78052-038-4

Music, 2003, ‘Sh Boom’, OST from film ‘Mona Lisa Smile’, mix by Trevor Horn

Christopher J. Plack, 2005, The Sense of Hearing, ISBN: 978-1-84872-987-2

Izhaki Roey, 2008, Mixing Audio, ISBN: 978-0-240-52068-1

Ronettes, 1963, Music ‘Be My Baby’, Mix by Phillip Spector

Swiss Sound, 1998, News and Views from Studer, May 1998 No. 42

The CIPIC HRTF Database, 2001,

http://interface.cipic.ucdavis.edu/sound/hrtf.html, first access in 2013

Jez Wells, 2015, Compter music tools,

http://www.jezwells.org/Computer_music_tools.html, first access in 2014