A Tetrahedral Microphone Processor for Ambisonic Recording

Fons ADRIAENSEN
Laboratorio di Acustica ed Elettroacustica, Parma, Italy
fons@kokkinizita.net

Abstract

This paper introduces a Linux audio application that provides an integrated solution for making full 3-D Ambisonics recordings by using a tetrahedral microphone. Apart from the basic A to B format conversion it performs a number of auxiliary functions such as LF filtering, metering and monitoring, turning it into a complete Ambisonics recording processor. It also allows for calibration of an individual microphone unit based on measured impulse responses. A new JACK backend required to make use of a particular four-channel audio interface optimised for Ambisonic recording is also introduced.

Keywords

Ambisonics, tetrahedral microphone, recording.

1 Introduction

The standard first-order Ambisonics B-format consist of four signals named W, X, Y and Z. In acoustic field theory terms, W represents the *pressure* signal at a given point in space, and X, Y, Z the three components of the *velocity vector* at the same point, projected onto orthogonal axes. Conventionally X points forward, Y left, and Z up.

These four signals also correspond to the outputs of four real microphones - an omnidirectional one for W, and three figure-of-eight ones for X, Y and Z - provided one can find a way to put these four microphones at exactly the same point in space.

For horizontal-only surround recordings Z is not used, and it is possible to mount the three required mics close together in a vertical line so they are effectively coincident for sounds arriving from horizontal directions. But for full 3-D such a *direct B-format* setup is no longer practical.

One solution, already developed by Michael Gerzon e.a. in the early years of Ambisonics [Gerzon, 1975], is to use a *tetrahedral microphone*. This contains four cardioid or near-cardioid capsules mounted very close together at the vertices of a regular tetrahedron and pointing outwards, as shown schematically in fig. 1. Figure 2 shows an example of how this may be realized in practice.

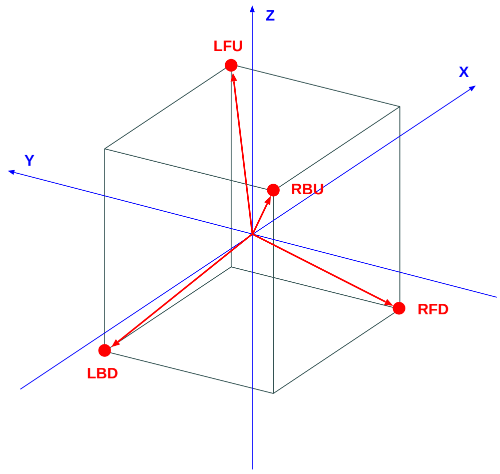

Figure 1: Tetrahedral mic geometry.

Given the four A-format microphone signals LFU, RFD, RBU and LBD[1], we can find the B-format signals from

$$\begin{aligned} W' &= LFU + RFD + RBU + LBD \\ X' &= LFU + RFD - RBU - LBD \\ Y' &= LFU - RFD - RBU + LBD \\ Z' &= LFU - RFD + RBU - LBD \end{aligned}$$

To find the correct W, X, Y, Z, we also need to filter the outputs of this *A-B matrix* and apply some gain factors. Two types of filter are required, one for the zero order component W, and a second one for each first order one, X, Y, Z. These filters will be discussed in more detail in section 3.

The first tetrahedral mics were manufactured by Calrec Ltd. in the UK, and their technology

[1] LFU means *left-front-up*, etc.

Figure 2: A tetrahedral mic (ST250).

was later acquired by Soundfield Ltd.[2] These are quite expensive microphones, not only because of their high quality, but also because they need a hardware processing unit to perform the A to B-format conversion and filtering mentioned above, and in particular because for best results each microphone needs to be calibrated individually and provided with a matched A-B matrix and/or filters.

The Danish microphone manufacturers DPA[3] were the first to offer a tetrahedral mic without a processing unit. It was a very high quality mic, but it is unfortunately no longer available. Soundfield Ltd. also produce such an A-format microphone, the SPS200-A.

Recently, Core Sound[4] has announced the relatively inexpensive *Tetramic*. It should be available when this paper is presented, and its price is expected to be below 1000 USD. It comes without a controller, and relies on a software solution for A to B-format conversion.

Given such an application, this mic provides an affordable solution for Ambisonic recording. While it was this announcement that triggered the development of the *Tetraproc* software presented in this paper, it should be pointed out that this software can be used with any tetrahedral microphone. It is published under the GPL license, and in no way, either technically or commercially, linked to products of Core Sound or any other manufacturer.

2 Tetraproc processor architecture

Apart from the basic A to B format conversion, Tetraproc also provides some convenient monitoring functionality. Ambisonic microphones are often used for live recording, and in these circumstances one wants a system that is easy to set up and use, and that allows for verification of the recording chain. In practice the only other software needed should be the JACK server and a recording application such as Ardour.

Figure 3 (next page) shows the processing chain implemented in Tetraproc. The high pass filters, the mute and invert switches, and the monitoring functions are controlled by the graphical user interface. All other modules are set up using a separate configuration window, and these settings are saved into configuration files. These config files are also generated by a separate calibration program discussed in section 4.

2.1 A to B format conversion

Going from the A-format microphone inputs to the B-format output, the following processing steps are performed:

- **High pass filtering.** This has an adjustable cutoff frequency and a slope of 24 dB/oct. This is really an essential feature. Figure-of-eight microphones having a good low frequency response are also excellent detectors of earthquakes, passing underground trains, waggling mic stands, slamming doors and air currents. These can result in rather large amplitude signals, and cutting off low frequencies is the only way to get rid of them. Ideally this should be done before AD-conversion, but not all audio interfaces provide such filters.
- **Mute switches.** For testing connections and verifying correct operation of the microphone it is convenient to be able to listen to selected inputs, hence the mute switches which are provided on the GUI.
- **Low frequency parametric filtering.** This is provided to adjust the frequency response of the microphones in this region. The parameters provided are centre frequency, bandwidth and gain. The same filtering is applied to all four channels.
- **A-format FIR filters.** These are implemented using fast FFT-based convolution and can be used to correct the frequency and phase response of the four microphones using filters calculated from measured impulse responses. This will be mainly im-

[2] <http://www.soundfield.com>

[3] <http://www.dpamicrophones.com>

[4] <http://www.core-sound.com>

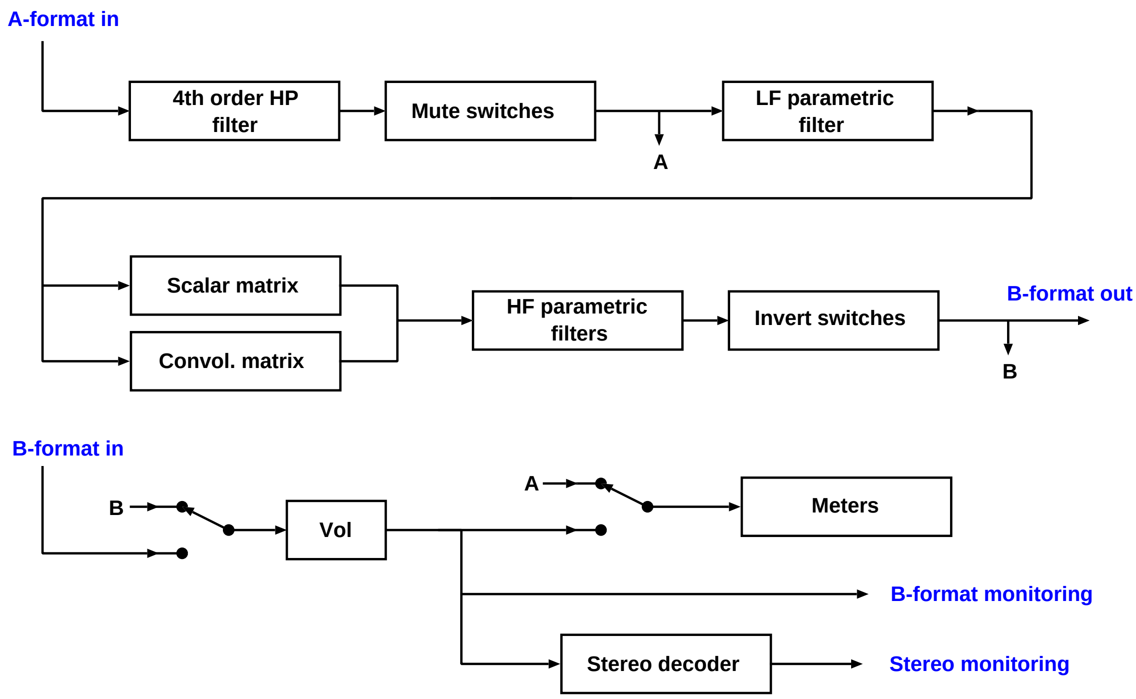

Figure 3: The Tetraproc processing chain

portant in the medium and high frequency regions.

- **A-B matrixing.** This performs the A-B transformation already described in the previous section. The actual matrix coefficients are modified to compensate for small gain and directivity mismatches between the four microphones. They are calculated by the calibration program described in section 4.
- **B-format FIR filters.** Again using fast convolution, these may be inserted to implement Angelo Farina's method (see next section) for obtaining the post-matrix filtering. They can be used together with the parametric sections that follow.
- **HF parametric filtering.** Two sections are provided in each channel to realize the required post-matrix filters. Parameters for these filters will be preset to sensible defaults and can be tweaked for optimum performance during the calibration of a tetrahedral microphone.
- **X, Y, Z inversion switches.** In some cases it is required to invert some of the first order signals, for example when the microphone is used upside down, hanging from its cable.

The output at this point is the B-format signal that will be recorded.

2.2 Monitoring functions

Monitoring can be switched between the B-format signal being recorded, or one being played back. A virtual stereo microphone with adjustable azimuth, elevation, microphone angle and directivity is provided for stereo monitoring. This module also provides a volume control and optional low-frequency crosstalk for headphone listening.

Four bargraph meters are provided on the GUI. These show either the A-format signals, or the B-format signal being monitored.

2.3 DSP implementation issues

None of the processing steps above present any real difficulty, but some attention to detail is required in order to obtain the highest quality.

The high pass filtering is implemented using a filter architecture optimised for low frequency filters described in [Adriaensen, 2006]. It is used to avoid problems with filter coefficient and signal quantisation which may arise with some standard digital filter structures.

When both the A-format and B-format FIR filters are enabled, they are combined together with the A-B matrix into a single four by four convolution process. This doubles the number of convolutions, but is both more efficient and more accurate than using eight separate ones.

The parametric filter sections use the Mitra-Regalia architecture.

3 Choice of the A-B matrix filters

During the study phase for the Tetraproc software it became clear that the choice of the post A-B matrix equalisation filters for a tetrahedral mic is sort of a black art. Theoretical analysis is possible, but due to conflicting requirements it does not lead to an obvious best solution. The problem gets more complicated again when the non-ideal characteristics of real microphones, in particular at high frequencies, and diffraction effects are taken into account — assuming the details of these are known at all.

The filters are necessary because at higher frequencies the size of the microphone array becomes comparable to the wavelength, and the microphones can no longer be considered to be coincident. The result is that while a B-format microphone has very good polar patterns at low and medium frequencies (better than most real omni or gradient mics), these will degrade at higher frequencies and break down above about 10 kHz (as they do with real microphones, except some of the very best). The frequency response of the B-format signals at high frequencies depends on the direction of the sound, and a compromise has to be found.

A second reason why these filters are necessary is to ensure that the B-format signals remain exactly in phase over the entire useful frequency range — this phase relationship is an essential feature of the Ambisonic system.

Considering just the theoretical response assuming perfect microphone capsules, there is already the choice between equalising for flat free field response in some preferred directions (e.g. the cardinal axes), or for flat diffuse field response, considering signals arriving from all directions.

In his famous paper [Gerzon, 1975] which seems to be one of the few original and authoritative publications on the subject,[5] Michael Gerzon notes that since the effective polar patterns become quite complex at high frequencies, it would be best to equalise for flat diffuse field response, and also shows plots of the corrections that would be required to do this. Gerzon provides design parameters for some analog filters, but these do not match the diffuse field curves. The filters actually used in one of the products based on his work [Calrec Ltd., 1984] are again different.

The theoretical curves depend on the radius of the array, and even more strongly on the directivity of the microphone capsules used. The latter will have some nominal value but it will in practice not be constant over the full frequency range. The actual values at high frequencies will be unknown in most cases.

In the light of all this, it seems unwise to include only fixed post-matrix filters in an application such as Tetraproc. It is for this reason that the two sections of parametric filtering are provided. The software package will contain a number of parameter presets for these filters, corresponding to the theoretical curves for a range of array diameters and directivities, and in many cases one of these may prove to be satisfactory. But for best results the filter settings should be derived from a calibration procedure, as outlined in the next section.

While the parametric filter approach seems to be the one preferred by some specialists in this field [Lee, 2006], a rather different one was suggested by Angelo Farina. He proposes to use Kirkeby-inverted measured impulse responses not only for equalising the individual microphone capsules, but also for the the post-matrix filters [Farina, 2006]. The Tetraproc software also permits the use of this method by providing the post-matrix convolution step.

4 The calibration procedure

While it is possible to use *Tetraproc* with a default configuration for a given microphone type (the available models are slightly different in terms of geometry and polar patterns), best results will be obtained only if the A-B conversion process is calibrated for each particular microphone. This is because each mic capsule will have its own small deviations from nominal sensitivity, directivity and frequency response. For a normal mic these don't matter much, but they become significant when the signals from a number of mics recording the same sound are combined in a way that relies on cancellation, as is the case for the first order outputs of the A-B matrix.

The complete calibration procedure can be divided into three parts.

[5] The PDF file of this paper as available from the AES is a scanned version of the original typewritten document and some parts of it are difficult to read. A typeset version is available from this author on request.

- The most important part is the compensation for any mismatching in sensitivity and directivity of the four capsules that make up a tetrahedral microphone. Any errors here will result in defective polar patterns of the virtual B-format microphones over the entire frequency range. The compensation is done by adjusting the coefficients of the A-B matrix.
- A second aspect is the adjustment of the post-matrix equalisation. This EQ compensates for the fact that at medium and high frequencies the four capsules are no longer coincident, as discussed in the previous section. In particular at higher frequencies (above about 5 kHz), this involves many unknown factors, and it will always be a compromise between conflicting requirements.
- Finally, the low frequency response of the microphone can be adjusted. Almost all directional mics will show some sort of high-pass response in this region. For some applications, e.g. orchestral recordings, this should be equalised. In practice this will vary little between mics of the same type, so default equalisation parameters will do in many cases.

A separate calibration application has been developed to assist the user with this rather complicated process. This program requires a number of measurements as its input. For the simplest case, when only the sensitivity and directivity compensation is performed, this involves eight recordings of a test signal reproduced by a loudspeaker, made at intervals of 45 degrees in the horizontal plane. This is a relatively simple operation that can be performed by all users. Alternatively, it is possible to use eight impulse responses measured using a program such as *Aliki*. This procedure also requires a separate omnidirectional measurement microphone, to compensate for the frequency response of the speaker used to reproduce the sweep signals. If these IR are available it is also possible to adjust the post-matrix equalisers, or to use the B-format convolution. Using the A-format convolution to equalise each capsule separately requires another set of measured impulse responses. For adjusting the low frequency parametric equaliser, four recordings of a special test signal reproduced using a small point source speaker are required.

Some parts of the application program work fully automatically, and others require interaction with the user who is required to interpret some results presented in a graphical format. It is well beyond the scope of this paper to explain the processing performed by the calibration program in any detail. A complete description together with instructions on how to perform the required measurements will be available in the manual for this application.

5 Supporting the 4Mic interface

Together with the *Tetramic*, Core Sound also introduces an audio interface optimised for use with this microphone. The *4Mic* interface offers four phantom powered inputs, precisely tracking gain controls, and AD conversion up to 24 bit and 192 kHz. Output is via one or two SPDIF streams. The unit can be configured to multiplex the four channels onto one stereo SPDIF stream at the double sample rate. The purpose of this is to enable the use of existing portable 2-channel recorders to capture the four A-format signals. The resulting stereo WAVEX file can then be converted to the required 4-channel one by just modifying its header.

Since there seem to be few audio interfaces offering two SPDIF inputs, it seems attractive to use the multiplexed output format also when recording via *Tetraproc*. This raises the question of where to implement the demultiplexing. The following have been considered:

- **In the drivers.** This would require changes to the source code of all drivers supporting and SPDIF interface, and is therefore not a viable option.
- **In libalsa.** It would probably take a long time to get this accepted into the ALSA source tree, assuming it would be accepted at all.[6]
- **In a libalsa plug-in.** This seems to be impossible as the plug-in interface does not allow for different input and output sample rates.
- **In an application.** It would be possible to include the demultiplexing in e.g. Tetraproc, but this would mean that the entire JACK graph has to work at the

[6] The ALSA developers do not seem to be interested in the problem. The author posted several requests for information on the ALSA developers mailing list and did not even receive a single reply.

double sample rate, and the demultiplexed channels would need to be up-sampled, resulting in a useless doubling of the recorded file sizes.

- **In a JACK backend.** This is the solution that was finally chosen.

The *fourmic* backend supplied with *Tetraproc* opens the ALSA driver at twice the sample rate and period size as seen by the JACK engine. It demultiplexes a selected stereo pair to four capture ports, and optionally also up-samples up to four playback ports so they can be used for monitoring.

There is one problem with this demultiplexing. The first samples delivered by the ALSA drivers seem to correspond to a random offset into an SPDIF frame. So there is a one in two chance that this is an odd offset, resulting in the channels being swapped and two of them having a one sample delay w.r.t. the others. The ALSA API does not seem to provide any means to find out the current position in an SPDIF frame, so the only solution for this at the time of writing is to check the channel assignments and restart JACK if they are wrong.

6 Acknowledgements

The design of this software would not have been possible without the generous help of the small community of Ambisonic experts.

I would like to thank Richard Lee for sharing his ideas on tetrahedral microphone alignment and equalisation. Also many thanks to the members of the Sursound mailing list, in particular Aaron Heller, Angelo Farina, Dave Malham, David McGriffy and Eric Benjamin, and to Len Moskowitz of Core Sound.

References

Fons Adriaensen. 2006. Near field filters for higher order Ambisonics. Available from <http://www.kokkinizita.net/linuxaudio>.

Calrec Ltd. 1984. Technical manual for the Mk4 soundfield microphone.

Angelo Farina. 2006. A-format to B-format conversion. <http://pcfarina.eng.unipr.it/Public/B-format/A2B-conversion/A2B.htm>.

Michael Gerzon. 1975. The design of precisely coincident microphone arrays for stereo and surround sound. *50th Audio Engineering Society Conference*, (AES preprint 8083L20).

Richard Lee. 2006. Sound field alignment and EQ. Private communication.