Audio Engineering Society

Convention Paper

Presented at the 125th Convention
2008 October 2–5 San Francisco, CA, USA

Is My Decoder Ambisonic?

Aaron J. Heller[1], Richard Lee[2], and Eric M. Benjamin[3]

[1]*Artificial Intelligence Center, SRI International, Menlo Park, CA 94025, USA*

[2]*Pandit Littoral, Cooktown, Queensland 4895, AU*

[3]*Dolby Laboratories, San Francisco, CA 94044, USA*

Correspondence should be addressed to Aaron J. Heller (heller@ai.sri.com)

ABSTRACT

In earlier papers, the present authors established the importance of various aspects of Ambisonic decoder design: a decoding matrix matched to the geometry of the loudspeaker array in use, phase-matched shelf filters, and near-field compensation. These are needed for accurate reproduction of spatial localization cues, such as interaural time difference (ITD), interaural level difference (ILD), and distance cues. Unfortunately, many listening tests of Ambisonic reproduction reported in the literature either omit the details of the decoding used or utilize suboptimal decoding.

In this paper we review the acoustic and psychoacoustic criteria for Ambisonic reproduction, present a methodology and tools for "black box" testing to verify the performance of a candidate decoder, and present and discuss the results of this testing on some widely used decoders.

1. INTRODUCTION

This paper is about testing Ambisonic decoders. The decoder is the component of an Ambisonic reproduction system that derives the loudspeaker signals from the program signals. Unlike most other surround sound systems in which each channel of a recording is intended to drive a single loudspeaker directly, an Ambisonic recording can be played back on a variety of speaker layouts, both 2-D and 3-D, by using an appropriate decoder.

A key feature of Ambisonic theory is that it provides a mathematical encapsulation of practically all known auditory localization models, except the pinna coloration and impulsive (high-frequency) interaural time delay models. These mathematical descriptions can be used to prove theorems about surround sound recording and reproduction, predict what spatial information can and cannot be conveyed by a particular system, guide the design of decoders, and as discussed in this paper, evaluate

and validate implementations.

We assume that the reader is familiar with the basic workings of surround sound in general and Ambisonics in particular. Background material on these topics, as well as sample Ambisonic recordings, can be found at various websites [1, 2].

Our interest in determining whether or not a given decoder meets the criteria for Ambisonic reproduction is motivated by practical considerations. When we first conducted listening tests, we did what many do: obtained some recordings made with a Soundfield microphone, set up six loudspeakers in a hexagon, downloaded a decoder off the Internet, and listened with the default settings. What we experienced was quite confusing — completely ambiguous localization and severe comb filtering artifacts from slight head movements. Over the next few listening sessions, we tried other software decoders and other settings with different but equally unsatisfying results. Had we not had previous experience with good Ambisonic reproduction, we might have stopped there and written off Ambisonics as yet another failed surround sound technology.

Instead, we went to the benchmark of good Ambisonic playback, what are known informally as *Classic Ambisonic Decoders* — the hardware-based decoders designed by the original Ambisonics team [3] — and built up an offline, file-to-file decoding workflow that mimicked the processing performed by those decoders. Since each step produced an intermediate file, we were able to verify that our implementation was performing as expected. The techniques described in this paper are a formalization and extension of this verification process.

Finally, by using a playback program that provided synchronized playback of a number of files and rapid switching among them, we were able to gain an understanding of the effects of each of the key components in an Ambisonic decoder: a decoding matrix matched to the geometry of the loudspeaker array in use, phase-matched shelf filters, and near-field compensation (NFC). These listening tests demonstrated that using the correct decoder results in dramatically improved performance [4, 5].

A number of recent papers have reported on the results of Ambisonic listening tests that have used decoders that are clearly faulty or employed incorrectly. As an example, in reference [6] the authors compare various spatialization techniques, including Ambisonics. The methodology used was well thought out, but unfortunately the software used to decode the Ambisonic program material may not have been the most appropriate:

> "The 'in-phase' ambisonic decoder was selected as it is recommended for larger rooms and listening areas, preventing anti-phase signals to be fed to the loudspeaker opposite to the sound source."

Later in the paper, the authors conclude that Ambisonics provides poor localization. However, given that the listening tests were performed with single listeners using a speaker array with 2-meter radius, the best (known) methodology for decoding would have been to perform exact, or velocity decoding at low frequencies, energy decoding at middle and high frequencies and use near-field compensation.

Other software decoders have many adjustments, but their authors provide little or no guidance on appropriate settings for various playback situations, making it difficult for a user to know if they are functioning correctly without extensive listening tests. We have read many accounts of "phasey", "ambiguous", or "unpleasant" Ambisonic reproduction that can be attributed to this problem. In particular, phasey reproduction will occur when exact velocity decoding is used at higher frequencies, where the wavelengths are smaller than the inter-ear distance.

The key point here is that it is not enough to simply specify that an Ambisonic decoder was used; not all decoders or decoder philosophies perform in the same way. It is also worth to noting that various workers in the field may not want to design a decoder; they simply want to verify that an existing one works properly and then use it.

Good engineering practice dictates that the behaviors of the individual components of a system under test be verified so that its overall performance can be properly characterized. While the design criteria have been outlined or implied in many papers, we have found no discussion of tools or methodologies to assess how well they have been met in a given implementation.

We confine the discussion in this paper to decoders suitable for one or two listeners.[1] In this paper we test only

[1] Design of decoders that work well over large areas is a distinct art and in general involves additional constraints that compromise their performance for small areas. [7]

horizontal, first-order Ambisonic decoders; however, the extensions for full 3-D reproduction (periphony) and arbitrary orders are straightforward.

There are a number of additional factors, any of which can have a large effect on the quality of playback but are beyond the scope of what is discussed here, such as room acoustics, accuracy of speaker positioning and matching, timing skew in multichannel D/A converters, and so forth. Simply due to the number of interconnections, speakers, and amplifiers in a typical Ambisonics playback system, the odds of making a setup error are much higher than in the case of stereo and the faults more difficult to diagnose than a channel reversal in stereo reproduction.

Due to space limitations, we test just four decoders and a single speaker configuration, the $\sqrt{3}:1$ rectangle. This configuration was preferred over a square layout in previous listening tests, as well as being easier to fit in most domestic rooms. We intend to populate our website [8] with more test results over time.

In summary, we are trying to decide if a given decoder and loudspeaker configuration meet the criteria for Ambisonic reproduction as defined by Gerzon in [9].

> A decoder or reproduction system for 360° surround sound is defined to be Ambisonic if, for a central listening position, it is designed such that
>
> i) velocity and energy vector[2] directions are the same at least up to around 4 kHz, such that the reproduced azimuth $\theta_V = \theta_E$ is substantially unchanged with frequency,
>
> ii) at low frequencies, say below around 400 Hz, the magnitude of the velocity vector is near unity for all reproduced azimuths,
>
> iii) at mid/high frequencies, say between around 700 Hz and 4 kHz, the energy vector magnitude, r_E, is substantially maximised across as large a part of the 360° sound stage as possible.

We feel that these are necessary, if perhaps not sufficient, conditions for good surround sound reproduction.

[2] Precise definitions of these are given in the next section.

2. REVIEW OF AMBISONIC CRITERIA

Gerzon defines two primitive models that are characterized by the velocity localization vector ($\mathbf{r_V}$) and energy localization vector ($\mathbf{r_E}$). These models encapsulate the primary Interaural Time Difference (ITD) and Interaural Level Difference (ILD) theories of auditory localization. The direction of each indicates the direction of the localization perception, and the magnitude indicates the quality of the localization. In natural hearing, from a single source the magnitude of each is exactly 1 and the direction is the direction to the source.

Ideally, both types of cue will be accurately recreated by a multispeaker playback environment and they will be in agreement with each other. In terms of Gerzon's models this means that $\mathbf{r_V}$ and $\mathbf{r_E}$ should agree in direction up to around 4 kHz; that below 400 Hz, the magnitude of $\mathbf{r_V}$ is near unity for all reproduced directions; and that between 700 Hz and 4 kHz, $|\mathbf{r_E}|$ is maximized over as many reproduced directions as possible. $|\mathbf{r_E}|$ achieves a maximum value of 1 for a single source and is always less than 1 for multiple sources. Gerzon observes that a value less than 0.5 "gives rather poor image stability." [10]

Following Gerzon [11], the magnitude and direction of the velocity vector, r_V and $\hat{\mathbf{r_V}}$, at the center of a speaker array with n speakers is

$$r_V \hat{\mathbf{r_V}} = \mathrm{Re}\,\frac{\sum_{i=1}^{n} G_i \hat{\mathbf{u_i}}}{\sum_{i=1}^{n} G_i} \tag{1}$$

whereas the magnitude and direction of the energy vector, r_E and $\hat{\mathbf{r_E}}$ are computed by

$$r_E \hat{\mathbf{r_E}} = \frac{\sum_{i=1}^{n} (G_i G_i^*) \hat{\mathbf{u_i}}}{\sum_{i=1}^{n} (G_i G_i^*)} \tag{2}$$

where the G_i are the (possibly complex) gains from the source to the i-th speaker, $\hat{\mathbf{u}}$ is a unit vector in the direction of the speaker, and G_i^* is the complex conjugate of G_i.

The main goal of the test protocol outlined in Sec. 3 is to recover the G_i's used by the decoder under test for a given source direction and speaker configuration. In the general case, they vary with frequency; hence, G_i and $G_i G_i^*$ can be thought of as the complex frequency and energy responses of the decoder for a particular direction.

The remaining parameters are the imaginary parts of velocity localization vector

$$\mathrm{Im}\,\frac{\sum_{i=1}^{n} G_i \hat{\mathbf{u_i}}}{\sum_{i=1}^{n} G_i} \tag{3}$$

which correspond to "phaseyness" arising from using filters whose phase responses are not matched. The most important part of this is the Y-component, the direction of the ear axis, over the frequency range 300 to 1500 Hz, and should be as close to zero as possible [11].

In general, optimizing the $\mathbf{r_V}$ and $\mathbf{r_E}$ vectors requires the use of a different decoding matrix for each frequency range. This can be accomplished with shelf filters or band-splitting filters similar to those used in loudspeaker crossovers. In either case, it is imperative that the filters are phase matched to preserve uniform frequency response over all directions.

Last, near-field compensation corrects for the reactive component of the reproduced soundfield when the listening position is within a few meters of the loudspeakers.

3. TEST PROTOCOL

It is not necessarily straightforward to determine whether a decoder is operating optimally simply by inspecting the software or listening to the output. They must be tested in order to verify that the desired characteristics have been achieved.

To do that, a test signal was created consisting of unit impulses at 2^{16}-sample intervals. This signal was encoded according to the B-format conventions (see Appendix 2) to create a series of unit impulses from varying source directions. This test signal is the equivalent to the output of a virtual soundfield microphone with a virtual source that is moved from one angular position to the next. The original series of impulses is included on an additional channel in the test file to act as a sync signal to simplify the later analysis. A plot of the test file is shown in Fig. 1. At 48 kHz sample frequency, the playing time of this file is 109.2 seconds. [3]

This file is then applied directly to the input of a software decoder, or played out though a multichannel soundcard into a hardware decoder, and the output recorded for subsequent analysis. In either case the intermediate output of the testing process is a file containing the resultant loudspeaker feeds derived by the decoder for the particular speaker configuration. The sync signal is recorded directly into the output file, without passing through the decoder under test. A screen capture showing this process is shown in Fig. 2.

[3] Matlab code to generate this test file, along with the code discussed in Sec. 3.1 is available on our website [8].

In the current work, 72 horizontal directions are used and the four loudspeaker feeds captured to the file, resulting in 288 impulse response (IR) measurements for each decoder configuration tested.

To perform the analysis, the complex frequency and energy responses are computed for each IR, yielding the G_is needed to compute $\mathbf{r_V}$ and $\mathbf{r_E}$ according to Eqns. 1 and 2. By examining these results, we can evaluate the decoder against Gerzon's criteria for Ambisonic reproduction as well as our other criteria.

It is worth noting that a number of methods can be used to measure the impulse response of a system. A survey of these techniques can be found in Stan, *et al.* [13]. Any of these techniques should work for this analysis. For our current purposes, we have selected the simplest one since it provides adequate signal-to-noise (S/N) ratio for direct testing of software decoders and removes the deconvolution process as another potential source of errors. To test hardware decoders, more sophisticated IR measurement techniques, such as MLS or Sine Sweep, are needed to deal with the lower S/N ratio and possibly higher distortion levels found in analog circuitry.

3.1. Analysis

We have created a set of tools in Matlab to analyze the recorded impulse responses and produce various plots, which give us insight into the behavior of the decoders. All the Matlab code used in this paper is available on our website [8].

The first function, `read_spkr_imps`, reads the speaker feeds file, normalizes the range of the data to fullscale = 1.0, extracts the sync pulses from the sync track, and then uses them to extract the individual impulses. It returns a $2^{16} \times 72 \times 4$ real-valued array, called `imps_dir_spkr` in this example. The indices to each dimension represent sample number, source direction in 5 degree increments, and speaker. It also returns the sample rate of the file, `Fs`. Optional return values are the raw samples and an array containing the locations of the sync pulses.

```
[ imps_dir_spkr Fs ] = ...
  read_spkr_imps( file );
```

The next function, `compute_ffts_imps`, takes the `imps_dir_spkr` array as input and computes the FFT of each impulse. This returns a complex valued array, with the same indices as above, but with frequency instead of sample number. It also returns the length of the FFT. By slicing though this array along various dimensions, we obtain the data we need to compute the parameters of interest.

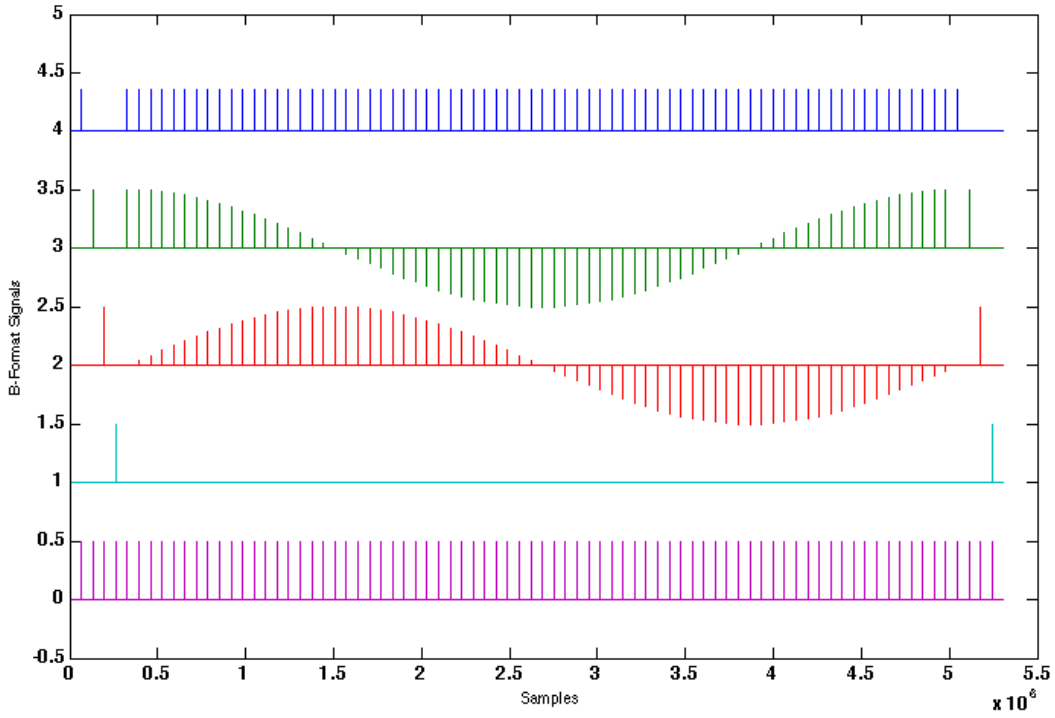


Fig. 1: A plot of the B-format impulses file encoding an impulse from 72 source directions in the horizontal plane. The signals from top to bottom are W, X, Y, Z, and original impulses. The signals are offset for clarity. The original impulse is included as a sync pulse that is recorded directly to the output file to simplify the analysis process. The first and last four impulses in the file excite each B-format component of the decoder independently. These are not used in the current analysis.

```
[ ffts_dir_spkr NFFT ] = ...
  compute_ffts_imps( imps_dir_spkr );
```

Next, the speaker positions are specified. In the $\sqrt{3}:1$ rectangular array used here, there are four loudspeakers, with 60 degree separation between the front and rear pairs. It is important that the order of these corresponds to the order of the recorded speaker signals in the data file.

```
phi = pi/6;  % frontal spkrs half-angle
speaker_weights = [ ...
  1, cos(   phi), sin(   phi), 0 ;
  1, cos(pi-phi), sin(pi-phi), 0 ;
  1, cos(pi+phi), sin(pi+phi), 0 ;
  1, cos(  -phi), sin(  -phi), 0 ];
```

The next functions compute the unnormalized components (Pressure, X, Y, and Z) of $\mathbf{r_V}$ and $\mathbf{r_E}$, by summing the ffts and the ffts squared weighted by the speaker locations. `Vpxyz` and `Epxyz` are indexed by frequency, direction, and component. The values in `Vpxyz` are complex, those in `Epxyz` are real.

```
Vpxyz = sum_pxyz( ffts_dir_spkr, ...
                  speaker_weights );

Epxyz = sum_pxyz( ffts_dir_spkr .* ...
                  conj(ffts_dir_spkr), ...
                  speaker_weights );
```

The absolute value of `Vpxyz` is used to examine the frequency response of the pressure and velocity components to determine whether or not the decoder under test implements near-field correction and dual-band processing. We also compare the frequency and phase responses in various directions to check that they are consistent.

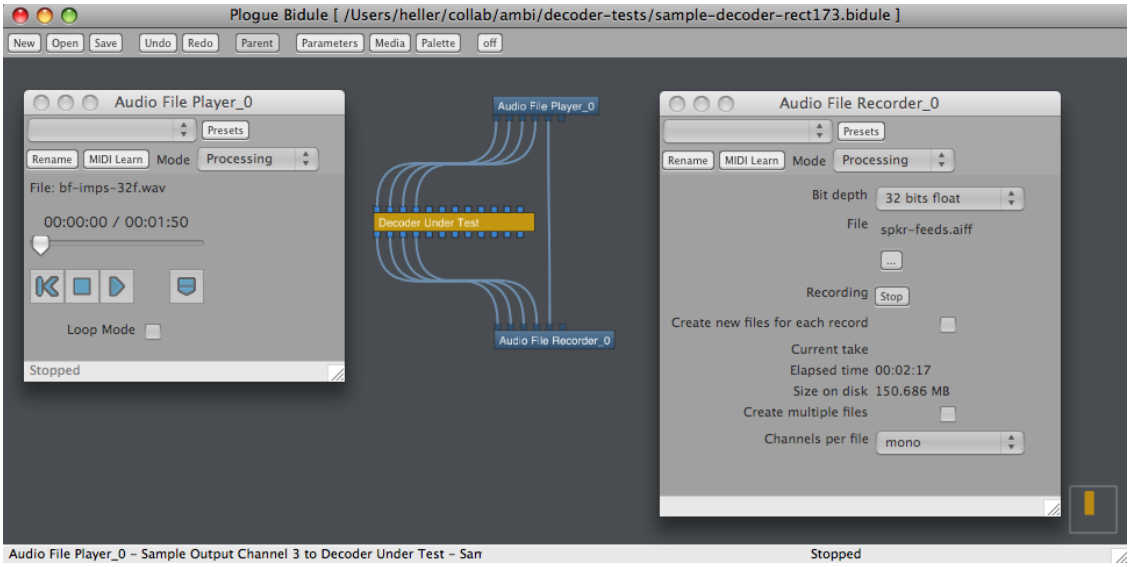


Fig. 2: A screen capture of Plogue Bidule [12] being used as a test harness for a decoder available as a VST plug-in. This works for Windows and MacOSX, which also supports Apple Audio Unit plug-ins. A similar setup using Jack and Ecasound is used to test Linux-hosted decoders.

$\mathbf{r_V}$ and $\mathbf{r_E}$ are now computed by normalizing by the pressure component and converting to spherical coordinates to yield the direction and magnitude. `rVcart` is complex. The real parts comprise $\mathbf{r_V}$. The imaginary parts, and in particular the one parallel to the Y-axis, indicate phaseyness due to use of filters that are not phase matched.

```
[ rVsph rVcart ] = r_from_pxyz( Vpxyz );

[ rEsph rEcart ] = r_from_pxyz( Epxyz );
```

At this point, polar plots of $\mathbf{r_V}$ and $\mathbf{r_E}$ are created at various frequencies and evaluated according to the Ambisonic criteria discussed in Sec. 2.

4. KEY COMPONENTS OF AN AMBISONIC DECODER

All decoders must perform the fundamental function of forming suitable linear combinations of the B-format signals for each loudspeaker in the array that reproduces the pressure and particle velocity at the central position in the array. This set of linear combinations is called the *exact* or *matching decoder matrix.* It is also called the *basic solution* of the speaker array or simply *the velocity decode.* Regardless of what it is called, it is unique for each loudspeaker array geometry.

In general, there are three types of loudspeaker arrays:

1. regular polygons and polyhedra, such as square, hexagon, cube, dodecahedron
2. irregular but with speakers in diametrically opposite pairs, such the $\sqrt{3}:1$ rectangle tested here
3. general irregular arrays, such as an ITU 5.1 array

In all cases the number of loudspeakers must exceed the number of B-format signals.

A method for deriving the decoder matrix for the first two types is given in Appendix 1.[4] In the case of regular arrays, this reduces to the result that the decoding and encoding matrices are identical, with the speaker positions substituted for the source positions. *The single most pervasive error in Ambisonic decoder design and use is assuming that also holds for irregular arrays.* Sec. 5.3 discusses the effect of this error. In our experience, most software Ambisonic decoders that can be downloaded are of this type.

The exact decoder matrix recreates the pressure and velocity at the central position under the assumption that

[4] Methods for the third type remain an area of open research [14].

the wavefronts are planar, i.e., sources and loudspeakers at an infinite distance. Sources and loudspeakers at finite distances produce wavefronts with a "reactive" (or imaginary) component, which is perpendicular to the direction of propagation, in addition to the "real" component, which is parallel to the direction of propagation. This results in the well-known bass-boosting proximity effect in directional microphones. It is important to understand that this is an actual physical effect, not a design flaw in the microphone or loudspeaker.[5] For point sources, the frequency at which the reactive and real components are equal is given by [15]

$$f = \frac{c}{2\pi d} \tag{4}$$

where c is the speed of sound and d is the distance from the loudspeaker.

In terms of the velocity localization vector, this makes $r_V > 1$ at low frequencies, which has the effect of widening the source images. This artifact is most apparent in recordings of string trios and quartets, where the cello sounds as if it is somewhat larger and closer than the other instruments. To compensate for this, a single-pole high-pass filter is applied to the velocity signals in the decoder. We call this *near-field compensation*. The design of this filter is covered in Appendix 1.

This exact reproduction of acoustic pressure and velocity is equivalent to the condition $r_V = 1$ in Gerzon's velocity localization model. In theory that happens at only a single point in space; however, in practice, it is good enough up to roughly one-half wavelength from the central position. If we desire reconstruction over an area of 0.5-meter, the exact decoder matrix can be used up to about 300 Hz and corresponds to the frequency regime of ITD-based auditory localization. If it is used beyond that frequency, comb filter artifacts and in-head localization effects will be experienced by the listener. *This is probably the second most common error in Ambisonic reproduction.*

At higher frequencies, say 700 to 4000 Hz, ILD-based auditory localization models are appropriate, which Gerzon encapsulates in the energy localization vector, $\mathbf{r_E}$. The one parameter that can be changed in the exact solution without changing the direction of the velocity localization vector, $\hat{\mathbf{r_V}}$, is the velocity-to-pressure ratio (i.e, the gain of X, Y, and Z vs. W), usually denoted by k.[6] Writing the magnitude of the energy localization vector, r_E, as a function of k, for any regular 2-D polygonal array with at least four speakers, we get

$$r_E(k) = \frac{2k}{2k^2+1} \tag{5}$$

which attains its maximum value at $k = \frac{\sqrt{2}}{2} \approx 0.7071 \approx -3.01$ dB.[7] In the case of a regular 3-D polyhedral array, with at least six speakers, we get

$$r_E(k) = \frac{2k}{3k^2+1} \tag{6}$$

which attains its maximum value at $k = \frac{\sqrt{3}}{3} \approx 0.5774 \approx -4.77$ dB. Fig. 3 shows graphs of these equations. Solutions with these values of k are often called "Max-r_E" or "energy decodes."

Next we must apportion the total between boost for pressure and cut for velocity in such a way that the overall loudness and balance between low and high frequencies is maintained. One approach is preserving the root-mean-square (RMS) level. In the 2-D case

$$W^2 + X^2 + Y^2 = 3 \qquad \textit{at both LF \& HF} \tag{7}$$

$$\frac{X}{W} = \frac{Y}{W} = \frac{\sqrt{2}}{2} \qquad \textit{at HF only} \tag{8}$$

solving for the HF gains

$$W = \sqrt{\frac{3}{2}} \approx +1.76\,\text{dB} \tag{9}$$

$$X = Y = \sqrt{\frac{3}{4}} \approx -1.25\,\text{dB} \tag{10}$$

In the 3-D case

$$W^2 + X^2 + Y^2 + Z^2 = 4 \qquad \textit{at both LF \& HF} \tag{11}$$

$$\frac{X}{W} = \frac{Y}{W} = \frac{Z}{W} = \frac{\sqrt{3}}{3} \qquad \textit{at HF only} \tag{12}$$

solving for the HF gains

$$W = \sqrt{2} \approx +3.01\,\text{dB} \tag{13}$$

$$X = Y = Z = \sqrt{\frac{2}{3}} \approx -1.76\,\text{dB} \tag{14}$$

[5] The implication for B-format signal encoding is that the X, Y, and Z signals must have a low-frequency boost and phase shift relative to the W signal, the amount of which is a function of the source distance. For natural acoustic sources, *a properly aligned Soundfield-type microphone does this by virtue of accurate transduction of the incident wavefronts,* and thereby encodes distance. For synthetic sources, this must be included in the encoding equations. Further discussion of this is in Appendix 2.

[6] k is equivalent to the inverse of the acoustic impedance.

[7] found by setting the derivative with respect to k equal to zero and finding the roots

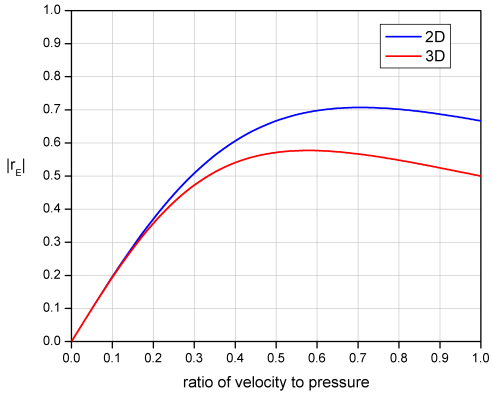


Fig. 3: Plots of r_E as a function of the velocity-to-pressure ratio k. The top curve shows the 2-D case, the bottom curve shows the 3-D case. The maximum values are $\frac{\sqrt{2}}{2}$ and $\frac{\sqrt{3}}{3}$, respectively.

For irregular diametric arrays ("type 2"), the magnitude of $\mathbf{r_E}$ varies in direct proportion to the angular density of the loudspeakers in a given direction, but for first-order Ambisonics the average value cannot exceed $\frac{\sqrt{2}}{2}$ for horizontal arrays and $\frac{\sqrt{3}}{3}$ for 3-D arrays. Gerzon notes that $\frac{\sqrt{3}}{3}$ is "perilously close to being unsatisfactory" [10]. However, in most periphonic (with height) systems, practical and domestic considerations often dictate that there will be more speakers in near horizontal than vertical directions. Localization is better in directions with more speakers — hence, our preference for a rectangle horizontal array over a square for predominantly frontal source material. However, Ambisonic systems have a clear advantage over other surround systems in that ambient/diffuse sounds are still perceived realistically even from directions with "poor localization."

A physical interpretation of the energy decode is that for a square array, a source directly ahead (azimuth zero), is reproduced with equal gain in the two front speakers and with zero gain in the two rear speakers. That is, the virtual microphone pattern formed by the gains from a source to the speakers is a near-supercardoid, with the two nulls at the angular locations of the rear speakers. The same is true in 3-D of a cube array; a frontal sound with azimuth and elevation zero, is reproduced with equal gain in the front speakers and with zero gain in the rear speakers.

This suggests that for optimal reproduction two decoding matrices are needed, one for low frequencies with $k = 1$ and another for high frequencies, with $k = \frac{\sqrt{2}}{2}$ or $\frac{\sqrt{3}}{3}$ for the 2-D and 3-D cases, respectively. Classic Ambisonic Decoders used phase-matched shelf filters to "morph" the exact solution into the energy solution. A more flexible strategy, first suggested by Barton [9], is to split each B-format signal into two bands so that two independent solutions can be used. We call this a *dual-band decoder*. It requires the use of phase-matched band-splitting filters similar to those used in loudspeaker crossover networks. The design of such filters is discussed in Appendix 1.

In summary, all the key components

- decoding matrices matched to the speaker array geometry
- near-field compensation
- frequency-frequency-dependent gains (shelf filters or a dual-band) optimizing for ITD and ILD cues using phase-matched filters

are needed to satisfy various localization mechanisms. It is this compensation for important psychoacoustic phenomena by simple means that makes a decoder Ambisonic.

Anecdotal evidence suggests that using shelf or band-splitting filters with small phase-matching errors is better than omitting frequency-dependent processing all together. This remains to be confirmed with formal listening tests, however. Finally, if one *must* use a decoder without frequency-dependent processing, the best result will be obtained using the energy-optimized values of k for all frequencies [4].

5. EXAMPLES

We review general classes of decoders and discuss the results of testing on four samples. Two perform well and two do not perform well for the given speaker configuration, the $\sqrt{3} : 1$ rectangle.

5.1. Types of Decoders

Beyond the issues of operating system, audio and user interfaces, the main distinguishing attributes of decoders are which of the three necessary components discussed in Sec. 4 are implemented and how the decoding matrix is specified to the decoder.

Some decoders provide presets such as square, rectangle, pentagon, hexagon, cube, dodecahedron, and so forth. With others the angular position of the speaker is entered along with the directivity of a virtual microphone or gains of the various orders of spherical harmonics. In the third type, the decoding matrix is specified directly.

5.2. Decoder 1

Decoder 1 is Adriaensen's *AmbDec* [16]. Version 0.2.0 was tested on a dual AMD Athlon machine running Fedora Core 8 and the Planet CCRMA distribution of audio software [17]. It has provisions for near-field compensation, dual-band processing, and a number of other features. The decoding matrices are specified directly in the configuration file. The distribution contains presets files for common loudspeaker arrays, but does not include the $\sqrt{3}:1$ array. It was tested with decoding matrices derived by the procedure outlined in Appendix 1. Distance was set to 2.0 meters and dual-band decoding was turned on with 380 Hz crossover.

Results are shown in Fig. 4. Examining the frequency and phase response graph, we see that NFC is implemented and has the correct -3 dB point for 2 meter distance (27 Hz). The NFC (correctly) has a large effect on the low-frequency gain and phase response of the velocity signals, which makes the magnitude of $\mathbf{r_V}$ less than 1 (0.91) and introduces a phase-matching error between pressure and velocity. These are intended to be the complement of the near-field effect of the loudspeakers, so that *at the listening position, the magnitude of* $\mathbf{r_V}$ *will be exactly 1 and the phase mismatch will be 0.*

To examine the frequency and phase response of the band-splitting filter in isolation, we ran a second test with NFC turned off. These results are shown in Fig. 5. Frequency-dependent gains are implemented with the correct values of k, and the phase responses of the filters are matched. Also note that the pressure and velocity gains are identical over all source azimuths.

Examining the polar plots of $\mathbf{r_V}$ and $\mathbf{r_E}$, we see that all source azimuths are rendered correctly at both high and low frequencies. At low frequencies the magnitude of $\mathbf{r_V}$ is (almost) 1 and at high frequencies the magnitude of $\mathbf{r_E}$ varies smoothly and has the highest attainable average for a first-order decoder of $\frac{\sqrt{2}}{2}$. The magnitude of $\mathbf{r_V}$ is slightly less than 1 (0.95) because the shelf filters are already affecting the gain at 150 Hz, as seen in Fig. 5(c).

This decoder and configuration *is* Ambisonic.

5.3. Decoder 2

Decoder 2 is a VST plugin that was tested on a MacBook Pro running OS X 10.5 using Bidule as a host program.[8] The GUI has sliders used to specify azimuth, elevation, directivity, and distance for each loudspeaker. There is also a switch to turn shelf filters on and off. The shelving frequency is not listed in the documentation, nor is there any mention of near-field compensation. It was tested with shelf filters switched on and the azimuths of the four speakers entered on the sliders. All other settings were left in their default positions.

Testing results are shown in Fig. 6. No near-field compensation is provided. Shelf filters are implemented with a turnover frequency of about 415 Hz. The magnitude of $\mathbf{r_V}$ varies with source direction, and its direction does not match the source direction. For some source directions $r_V > 1.0$. Frequency response varies with source direction. At 3 kHz, the magnitude of $\mathbf{r_E}$ is < 0.5 from 45 to 135 degrees azimuth. Note that over these azimuths, the shelf filters merely make a suboptimal r_E even worse. We emphasize that shelf filters will have the intended effect only when paired with the correct decoder matrix for the speaker array geometry. There is a small (15°) error in phase matching near the crossover frequency.

In summary, this decoder is not suitable for use with this speaker configuration when set up according to the instructions provided.

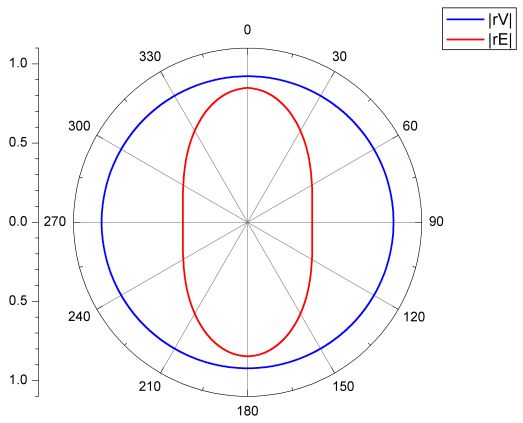
To be fair, it is possible to derive appropriate virtual microphone angles and directivities from the exact decoder matrix, enter those into decoders of this type and obtain somewhat better performance than we observe here. In the general case, the virtual microphones *will not* point at the speaker positions.

5.4. Decoder 3

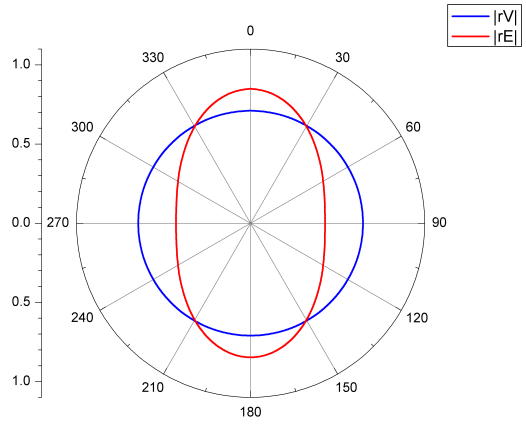
Decoder 3 is Csound's `bformdec` Opcode.[9] Csound is a computer programming language for dealing with sound, also known as a sound compiler or an audio programming language [18]. The sound processing elements are called *opcodes,* which are connected and invoked using *orchestra* and *score* files. The documentation for `bformdec` says "New in Version 5.07 (October 2007)."

[8] We do not identify Decoder 2 since many decoders appear to use the same underlying approach. We suspect that any one of them would have produced results no better than the one we happened to test.

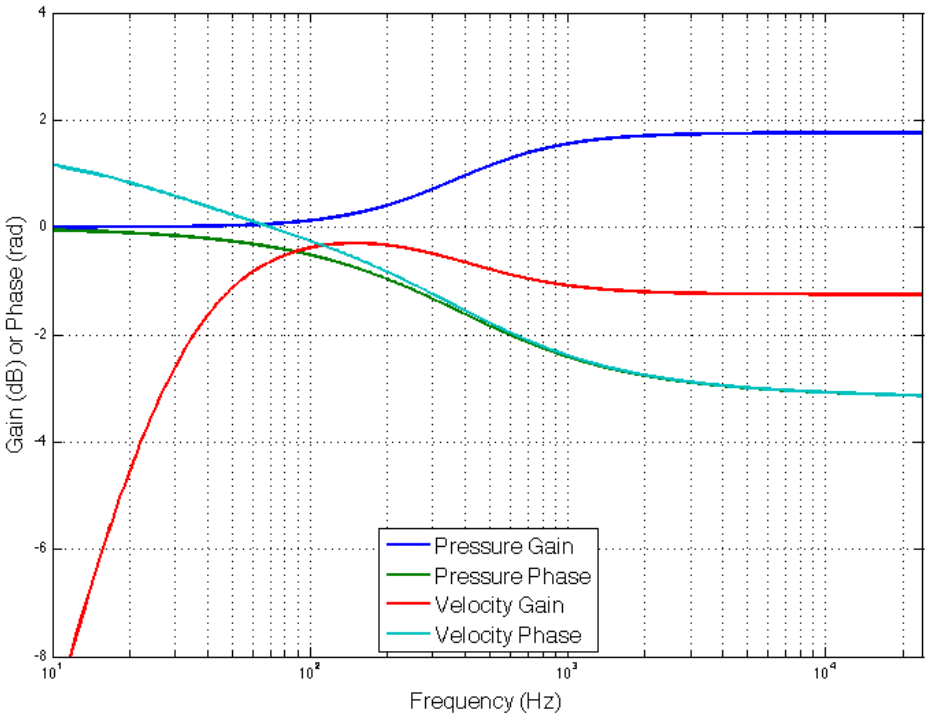
[9] Csound tests were performed by Sven Bien from Bremen University.



(a) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 150 Hz

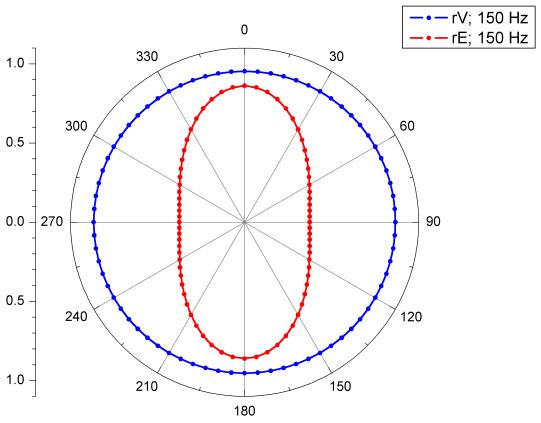


(b) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 3 kHz

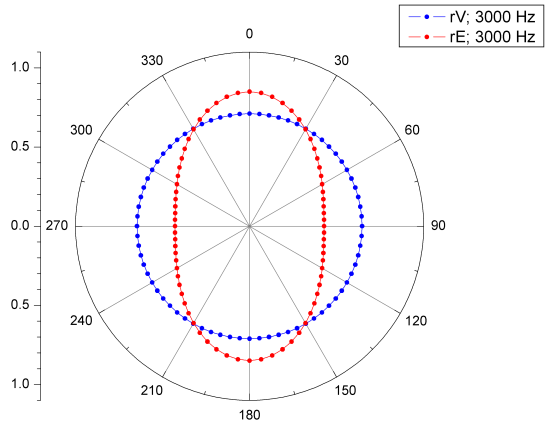


(c) frequency and phase response

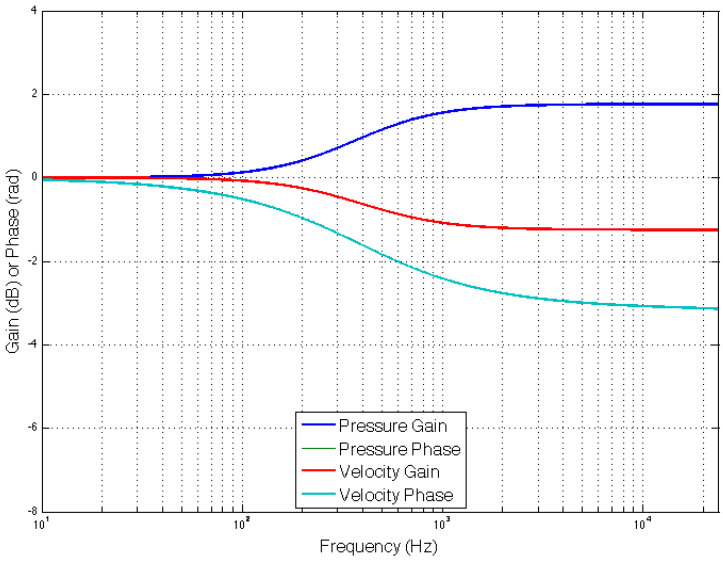
Fig. 4: *AmbDec* with configuration derived by the procedure given in Appendix 1. This is a very good result. (a) and (b) show $\mathbf{r_V}$ and $\mathbf{r_E}$ at 150 Hz and 3 kHz. Source directions are correct and matched. The magnitude of $\mathbf{r_V}$ is uniform in all directions and $\mathbf{r_E}$ at 3 kHz attains an average value of $\frac{\sqrt{2}}{2}$. (c) shows that NFC and dual-band processing is implemented. The next figure shows the same configuration with NFC switched off so that frequency and phase response can be examined in isolation.



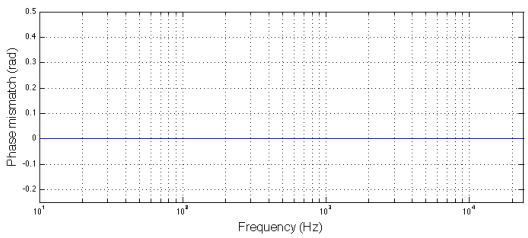
(a) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 150 Hz



(b) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 3 kHz

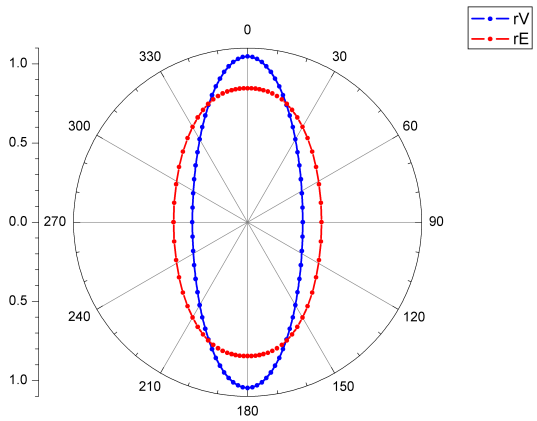


(c) frequency and phase response

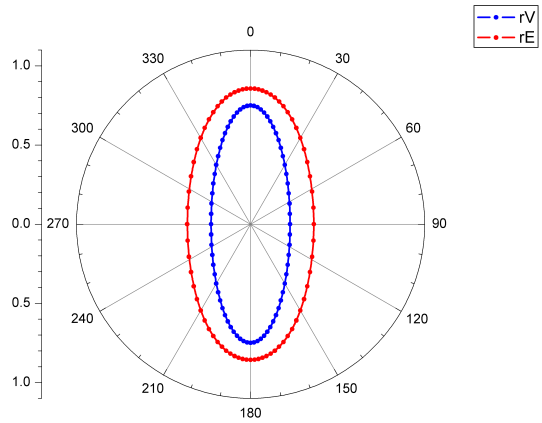


(d) phase mismatch between pressure and velocity

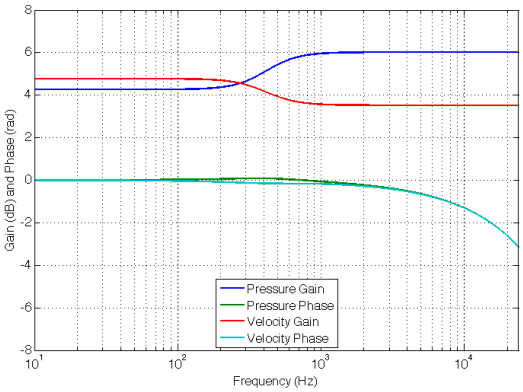
Fig. 5: *AmbDec* with NFC switched off. Only three lines are seen in (c) because the pressure and velocity phase responses are identical.



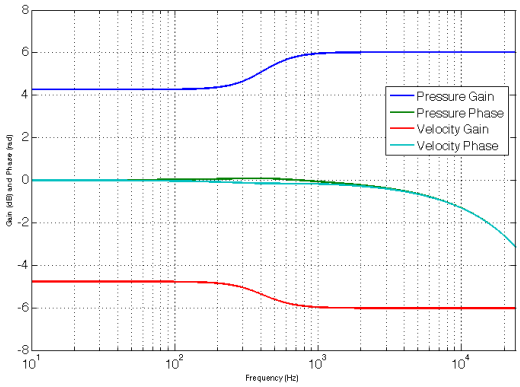
(a) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 150 Hz



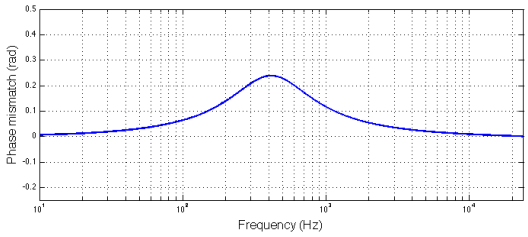
(b) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 3 kHz



(c) frequency and phase response for a 0° azimuth source



(d) frequency and phase response for a 90° azimuth source



(e) phase mismatch between pressure and velocity

Fig. 6: Decoder 2

This opcode does not have provision for the rectangular configuration we are using; however, it is still useful to include this test as an example of a decoder that is in widespread use.

The test was conducted with the following orchestra file:

```
sr = 48000
kr = 4800
ksmps = 10
nchnls = 5

   instr 1
a1 soundin "bf-1-sync.wav"
aw soundin "bf-2-w.wav"
ax soundin "bf-3-x.wav"
ay soundin "bf-4-y.wav"
az soundin "bf-5-z.wav"
a2, a3, a4, a5 bformdec 2, aw, ax, ay, az
          outc a1, a2, a3, a4, a5
   endin
```

Test results are shown in Fig. 7.

The most apparent feature of these results is that the frequency response is perfectly flat; neither NFC nor shelf filters are implemented. At low frequencies the magnitude of $\mathbf{r_V}$ is 1 and source directions are rendered correctly. However, at high frequencies it remains 1, which will result in in-head localization and comb filtering artifacts with head movement. This also puts the high-frequency $\mathbf{r_E}$ magnitude well below the optimum value. This decoder should not be used as currently implemented.

If one is forced to use this decoder, reducing the levels of X, Y, and Z by 3 dB will produce an "energy decode," which was recommended in [4] in cases where no shelf filters are available.

5.5. Decoder 4

Decoder 4 is a Mimim AD-10 decoder whose Ambisonic processing is similar to the model described here [3] and is an example of a Classic Ambisonic Decoder. It is an all-analog implementation of an Ambisonic decoder for horizontal decoding to four or six loudspeakers. It has a switch to turn NFC on and off and a "layout" adjustment to accommodate speaker arrays with aspect ratios ranging from 1 : 2 to 2 : 1. For these tests, the layout control was set to "3 o'clock", which was our best estimate of the setting for the $\sqrt{3} : 1$ array used for our analysis. NFC was switched on. The test was performed using an Audio Precision analyzer. Results are shown in Fig. 8. As can be seen from the graphs, correct shelf filters and NFC are implemented. At low frequencies, $r_V = 1$ and source directions are rendered correctly. At high frequencies, the maximum possible values of r_E are achieved, indicating that this decoder is operating near optimally for the speaker array used for these tests. The slight "egg shape" in $\mathbf{r_E}$ at low frequencies is most likely due to a gain mismatch between the front and rear loudspeaker outputs.

This decoder *is* Ambisonic.

6. LISTENING TESTS

One of the authors carried out informal listening tests in a domestic setting using some of the test files employed in earlier listening tests: voice announcements in eight directions, continuously panned pink noise, a recording of a classical chamber orchestra, and the applause that followed the performance. The last was recorded with a Calrec Soundfield Microphone MkIV, serial number 099, with original capsules and calibration.

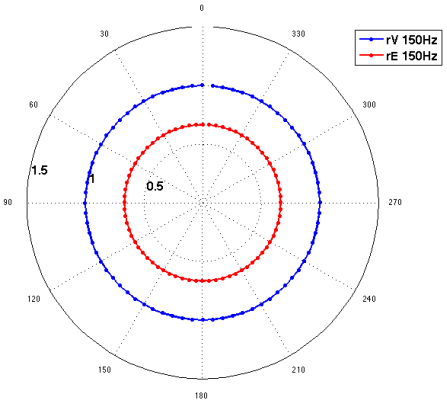
The results broadly confirm the results of the testing described in this paper.

- Adriaensen's *AmbDec* decoder provided good localization in all directions, uniform frequency response, and a good sense of envelopment, with no audible artifacts.
- With Decoder 2 all sources were localized to the front or rear, with no sense of envelopment and a very narrow "sound stage" on orchestral test material.
- We were not able to audition the Csound decoder directly, but did simulate it in Bidule using the measured parameters, and can confirm that the predicted comb-filtering and in-head localization artifacts are quite apparent.

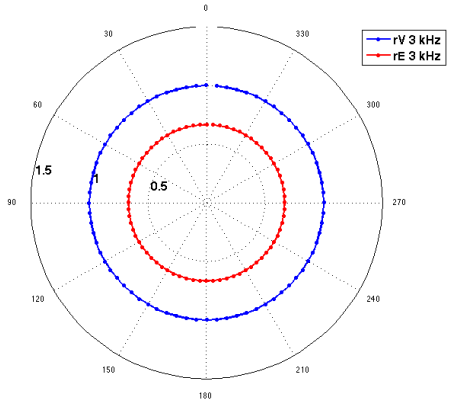
7. DISCUSSION

From the measurements reported above, it may be safely concluded that not all available Ambisonic decoders perform according to the requirements set down in Sec. 2, and in fact most do not! The most common problems that were found in decoders are

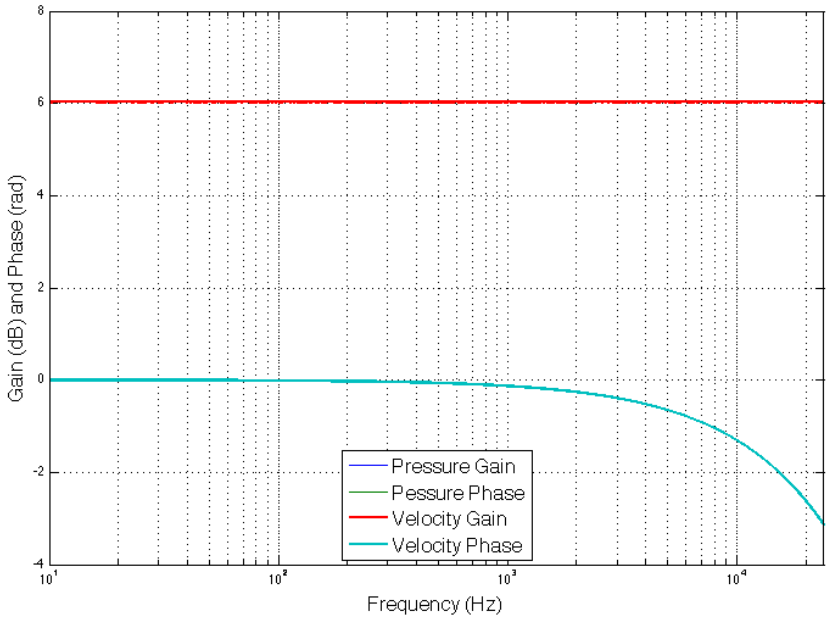
- Incorrect coefficients for rectangular or other non-regular polygonal loudspeaker arrays



(a) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 150 Hz

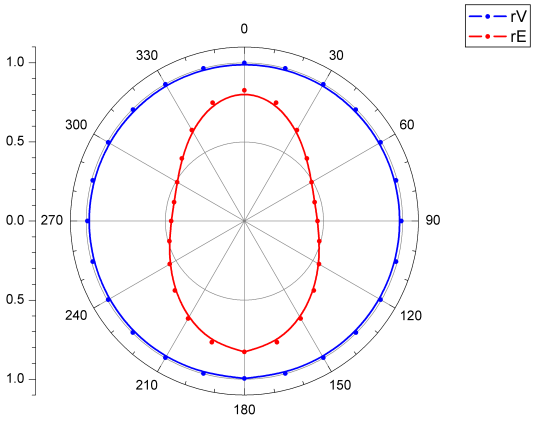


(b) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 3 kHz

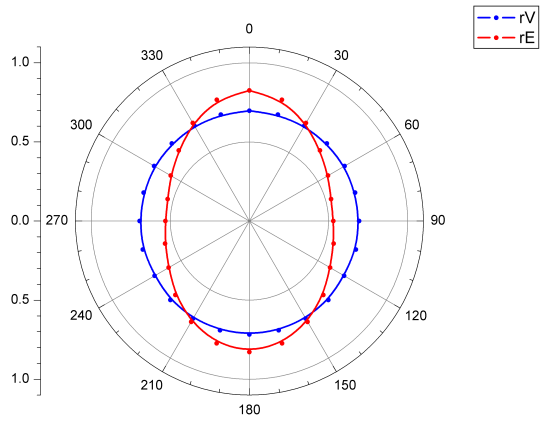


(c) frequency and phase response for all source azimuths

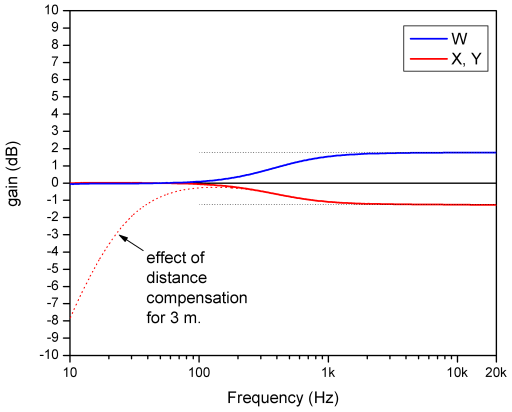
Fig. 7: Decoder 3 Csound's `bformdec` Opcode. Only two of the four lines are visible in (c) because the pressure and velocity responses are identical.



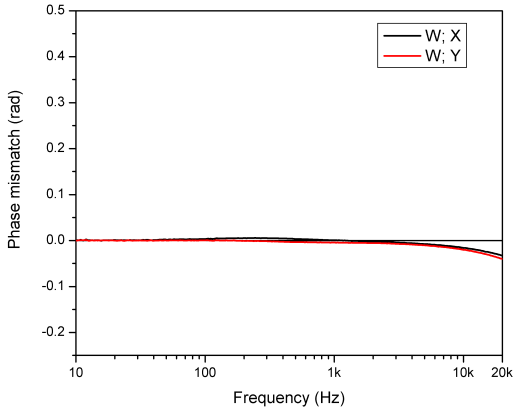
(a) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 150 Hz



(b) $\mathbf{r_V}$ and $\mathbf{r_E}$ measured at 3 kHz



(c) frequency response with and without NFC



(d) phase mismatch between pressure and velocity

Fig. 8: Decoder 4 Minim Analog Ambisonic Decoder.

- Lack of dual-band decoding
- Lack of near-field compensation

These omissions or errors cause the audio reproduction to suffer from poor localization behavior, phasiness, and other artifacts.

The regular polygon dilemma

Gerzon described, in *General Metatheory*, a "naïve decoder for a regular polygon loudspeaker Layout," with the following form:

$$W + X\cos\theta + Y\sin\theta \tag{15}$$

for which he proved that "the Makita and Energy vector localizations coincide, and the energy vector magnitude, r_E, cannot exceed $\frac{\sqrt{2}}{2}$. Unfortunately, that statement is true only for the case of regular polygons, and following this form for nonregular polygons gives an incorrect result. Specifically, if a regular array is narrowed, then the naïve decoder gives values for the coefficients that are larger for X and smaller for Y, relative to the regular polygon. But intuition tells us that narrowing the array would require less X and more Y to achieve the same localization vectors.

The correct decoder for the rectangle case is

$$W \pm \frac{X}{\sqrt{2}\cos\phi} \pm \frac{Y}{\sqrt{2}\sin\phi} \tag{16}$$

where 2ϕ is the angle subtended by the front two loudspeakers.

It may be argued, and in fact we do argue, that the rectangle decoder (and its 3-D extension, the bi-rectangle) is the single most important case for actual applications in reproduction of first-order Ambisonic recordings. The reason for this importance is that rectangular arrays fit better into ordinary rooms, and in addition that they give a needed improvement in mid/high-frequency localization toward the front and back when the correct decoder is used.

Many of the available software Ambisonic decoders do not implement dual-band decoding, presumably because of lack of knowledge about how to design IIR filters. When dual-band decoding is implemented, it frequently is found to utilize shelf filters that are not phase matched between the filter for W and the filters for X/Y/Z. Since that phase mismatch occurs only in the frequency range around the transition frequency (see Fig. 6(e)), it is difficult to evaluate the seriousness of the error. Clearly, any errors will be program dependent, depending on the spectral density in that particular frequency range. However, it is relatively easy to do it correctly and it *should* be done correctly. See Appendix 1.

Previous listening tests by the authors of this paper, and informal listening tests done during the writing of the present paper, have shown the importance of all the features of an Ambisonic decoder. The use of Ambisonic decoders that are inappropriate to the venue, or have incorrect decoding coefficients, or lack the important features of dual-band decoding and NFC will give results that are inferior to what would be obtained with a correct Ambisonic decoder and which will prejudice the results of comparative listening tests.

In this paper, we have made every effort to "tell it all" as clearly and plainly as we can without oversimplification, and back that up with examples, test files, and sample code that can be downloaded and used. We hope that researchers conducting experiments in audio localization will adopt these or similar techniques to validate their experimental setups and that decoder writers will now have necessary knowledge and tools to write better decoders.

A decade ago, lack of program material (B-format recordings) was the biggest problem with Ambisonics. Now that downloads of B-format recordings [2], relatively low-cost B-format microphones [19], and pocket-sized multichannel digital audio recorders are available, suitable program material is somewhat more plentiful.

The next hurdle is the creation of easy-to-use playback software that runs on popular computing platforms, about which we can say: These *are* Ambisonic.

8. ACKNOWLEDGMENTS

The authors thank Sven Bien from Bremen University and his advisor Jörn Loviscach at Hochschule Bremen, University of Applied Sciences, for help with Csound, and in particular Sven for running the tests under very tight deadline constraints. We also thank the participants in the *sursound* email discussion group for their encouragement and comments, Don Drewecki for providing a steady flow of top-notch B-format concert recordings for our listening tests, and Jim Mastracco for many hours of stimulating discussion on microphones and acoustics.

9. REFERENCES

[1] Richard Elen. Ambisonic.net - where surround-sound comes to life. http://www.ambisonic.net/.

[2] Etienne Deleflie. Ambisonia: The next generation surround sound experience, by the enthusiasts, for the home theatre. http://www.ambisonia.com/.

[3] Michael Gerzon. Multi-System Ambisonic Decoder. *Wireless World*, 83(1499):43–47, July 1977. Part 2 in issue 1500. Available from http://www.geocities.com/ambinutter/Integrex.pdf (accessed June 1, 2006).

[4] Eric Benjamin, Richard Lee, and Aaron Heller. Localization in Horizontal-Only Ambisonic Systems. In *Preprints from the 121st Audio Engineering Society Convention, San Francisco*, number 6967, October 2007.

[5] Richard Lee and Aaron J Heller. Ambisonic localisation - part 2. *14th International Congress on Sound and Vibration*, 2007.

[6] Catherine Cuastavino, Veronique Larcher, Guillaume Gatusseau, and Patrick Bossard. Spatial audio quality evaluation: Comparing transaural, ambisonics and stereo. *Proceedings of the 13th International Conference on Auditory Display*, 2007.

[7] David G Malham. Experience with large area 3-d ambisonic sound systems. *Proceedings of the Institute of Acoustics Autumn Conference on Reproduced Sound 8*, 1992.

[8] Aaron Heller. Ambisonics reference material. http://www.ai.sri.com/ajh/ambisonics.

[9] Michael A. Gerzon and Geoffrey J. Barton. Ambisonic Decoders for HDTV. In *Preprints from the 92nd Convention of the Audio Engineering Society, Vienna*, number 3345, March 1992. AES E-lib http://www.aes.org/e-lib/browse.cfm?elib=6788.

[10] Michael A. Gerzon. Practical Periphony: The Reproduction of Full-Sphere Sound. In *Preprints from the 65th Audio Engineering Society Convention, London*, number 1571, February 1980. AES E-lib http://www.aes.org/e-lib/browse.cfm?elib=3794.

[11] Michael A. Gerzon. General Metatheory of Auditory Localisation. In *Preprints from the 92nd Audio Engineering Society Convention, Vienna*, number 3306, March 1992. AES E-lib http://www.aes.org/e-lib/browse.cfm?elib=6827.

[12] Plogue Art et Technologie Inc. Bidule: modular audio software. http://www.plogue.com/.

[13] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio Engineering Society*, 50(4):249–262, Apr 2002.

[14] Bruce Wiggins, Iain Paterson-Stephens, Val Lowndes, and Stuart Berry. The Design and Optimisation of Surround Sound Decoders Using Heuristic Methods. In *Proceedings of UKSim 2003, Conference of the UK Simulation Society*, pages 106–114. UK Simulation Society, 2003. Availible from http://sparg.derby.ac.uk/SPARG/PDFs/SPARG_UKSIM_Paper.pdf (accessed May 15, 2006).

[15] Philip M. Morse and K. Uno Ingard. *Theoretical Acoustics*, chapter 7, page 311. Princeton University Press, 1986. ISBN: 0691024014.

[16] Fons Adriaensen. *AmbDec User Manual*, 0.2.0 edition, 2008. http://www.kokkinizita.net/linuxaudio/downloads/ambdec-manual.pdf.

[17] Fernando Lopez-Lezcano. Planet ccrma at home. http://ccrma.stanford.edu/planetccrma/software/.

[18] Wikipedia. Csound — wikipedia, the free encyclopedia, 2008. [Online; accessed 11-August-2008].

[19] Core sound tetramic. http://www.core-sound.com/TetraMic/1.php.

[20] Eric W. Weisstein. Moore-Penrose Matrix Inverse. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html.

[21] Wikipedia. Singular value decomposition — wikipedia, the free encyclopedia, 2008. [Online; accessed 10-August-2008].

[22] Wikipedia. Digital biquad filter — wikipedia, the free encyclopedia, 2008. [Online; accessed 11-August-2008].

[23] Leo Beranek. *Acoustic Measurements*, pages 56–64. John Wiley, New York, 1949.

[24] Philip Cotterel. *On the Theory of the Second-Order Soundfield Microphone*. PhD thesis, Dept. of Cybernetics. Reading University, Febuary 2002.

[25] Jerome Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format. *Preprints 23rd AES International Conference, Copenhagen*, 2003.

APPENDICES

1. DECODER DESIGN

We present some "cookbook" procedures for the design of the key decoder components, along with some digressions into the underlying theory and mathematics. Examples and implementations of all three can be found on the authors' website [8].

1.1. Exact-Solution Decoder Matrix

This method is based on the idea of *inversion* where we write down the direction of propagation of the acoustic impulse created by each loudspeaker in the array, decompose that into the selected set of spherical harmonics and use generalized inversion to derive the decoder matrix that recreates the original impulse. This is a generalization of Figure 12 ("The Design Mathematics") in [10] and can be shown to be equivalent to the least-squares solution. If the problem is under-constrained (many solutions), as it is for typical Ambisonic speaker arrays (more speakers than signals), it will give the solution that requires the minimum overall radiated energy, which also will yield the largest $|\mathbf{r_E}|$'s.

While this procedure provides a solution for any loudspeaker array, only regular arrays and those with speakers in diametrically opposed pairs (Type 1 and 2 from Sec. 4) will result in the directions of $\mathbf{r_V}$ and $\mathbf{r_E}$ agreeing for all source directions, which is one of the basic criteria for Ambisonic reproduction. As noted earlier, solution of general irregular arrays (Type 3) that satisfy Ambisonic criteria is beyond the scope of this paper.

An additional constraint is that all the speakers in the array are equidistant from the listening position. While this can be relaxed by introducing delays, "$1/r$"-level adjustments, and per-speaker NFC, it is also beyond the scope of this paper.

We assume that each loudspeaker in the array produces a planar wave front propagating towards the center of the array that is parallel to the direction given by its position relative to the center. For the i^{th} loudspeaker

$$\mathbf{L}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix} \tag{17}$$

where ${x_i}^2 + {y_i}^2 + {z_i}^2 = 1$, that is they are the direction cosines of the of the vector from the center of the array to the i^{th} loudspeaker. In spherical coordinates this is

$$\mathbf{L}_i = \begin{bmatrix} \cos\theta\cos\varepsilon & \sin\theta\cos\varepsilon & \sin\varepsilon \end{bmatrix} \tag{18}$$

where θ is the counterclockwise azimuth from directly ahead, and ε is the elevation from horizontal.

Next, we select a set of spherical harmonic functions, up to the desired order, that form an orthogonal basis and then project the speaker directions onto it. For first-order Ambisonics there is single choice, the B-format definitions.[10] In Cartesian coordinates, each $\mathbf{L}_i$ becomes

$$\mathbf{K}_i = \begin{bmatrix} \frac{\sqrt{2}}{2} & x_i & y_i & z_i \end{bmatrix}. \tag{19}$$

For the array used in this paper, the $\sqrt{3}:1$ rectangle, which has speakers at azimuths 30, 150, 210, and 330 degrees in the horizontal plane

```
K_rect30 =
   0.7071    0.8660    0.5000         0
   0.7071   -0.8660    0.5000         0
   0.7071   -0.8660   -0.5000         0
   0.7071    0.8660   -0.5000         0
```

where each row corresponds to the coefficients of a single speaker. For a cuboid array that is 2 meters wide, 3 meters deep, and 1.5 meters tall

```
K_cuboid =
   0.7071    0.5121    0.7682   -0.3841
   0.7071    0.5121   -0.7682   -0.3841
   0.7071   -0.5121   -0.7682   -0.3841
   0.7071   -0.5121    0.7682   -0.3841
   0.7071    0.5121    0.7682    0.3841
   0.7071    0.5121   -0.7682    0.3841
   0.7071   -0.5121   -0.7682    0.3841
   0.7071   -0.5121    0.7682    0.3841
```

[10] For higher-order Ambisonics, there are at least three possibilities.

We want to find **M**, the decoder matrix, that satisfies the condition

$$\mathbf{M} \times \mathbf{K} = \mathbf{I} \tag{20}$$

where **I** is the identity matrix, a matrix with 1's on the diagonal and 0's everywhere else and $\times$ indicates matrix multiplication. When that is satisfied, it means that the speaker array, **L**, in combination with the decoder matrix, **M**, can reproduce all the spherical harmonics independently (i.e., without crosstalk).

If **K** is invertible, then $\mathbf{M} = \mathbf{K}^{-1}$; however, in the case with most Ambisonic arrays (and in particular in the two examples above), it is not, so we use the least-squares solution to the system

$$\mathbf{M} \times \mathbf{K} - \mathbf{I} = \mathbf{r} \tag{21}$$

In the typical case, we have more speakers than signals, so this system is over determined and there are many solutions. By selecting the one that also minimizes the 2-norm of **M**, we obtain the one providing the highest average value of $|\mathbf{r_E}|$.

The desired solution of Eqn. 21, **M** is given by the *Moore-Penrose pseudoinverse* of **K** [20], which is available in *Matlab* and *GNU Octave* as the function `pinv()` and in many other computer mathematics systems, such as *Scilab* and *Mathematica.*

In *Matlab*,

```
>> M_rect30 = pinv(K_rect30);
>> transpose(M_rect30)
ans =
   0.3536    0.2887    0.5000         0
   0.3536   -0.2887    0.5000         0
   0.3536   -0.2887   -0.5000         0
   0.3536    0.2887   -0.5000         0

>> M_cuboid = pinv(K_cuboid);
>> transpose(M_cuboid)
ans =
   0.1768    0.2441    0.1627   -0.3254
   0.1768    0.2441   -0.1627   -0.3254
   0.1768   -0.2441   -0.1627   -0.3254
   0.1768   -0.2441    0.1627   -0.3254
   0.1768    0.2441    0.1627    0.3254
   0.1768    0.2441   -0.1627    0.3254
   0.1768   -0.2441   -0.1627    0.3254
   0.1768   -0.2441    0.1627    0.3254
```

where the function `transpose()` swaps the rows and columns of the matrix, yielding a matrix where each column corresponds to one of the B-format signals, W, X, Y, and Z and each row contains the decoder gains for the corresponding speaker for that signal.

We can now use Eqn. 21 to check the quality of the solution by examining the entries in **r**. Non-zero entries on the diagonal indicate a spatial harmonic that is not being reproduced correctly. Non-zero entries off the diagonal indicate crosstalk or aliasing between the spatial harmonics. Either condition indicates that further analysis of the array geometry is needed.

1.1.1. A further math digression...

One way to compute $\mathbf{A}^\dagger$, the pseudoinverse of **A**, is

$$\mathbf{A}^\dagger = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^* \tag{22}$$

where $\mathbf{A}^*$ indicates the transpose of **A**. With one further optimization (factoring out the *W* column, which is constant), this is what Gerzon is doing in Figure 12 of *Practical Periphony*[10], which is reproduced as Eqn. 4 in [4].

Most spreadsheet programs include basic matrix operations such as transposition and inversion, so it is possible to create spreadsheets that do these calculations, however from a numerical computing standpoint, this is not the best way to obtain the pseudoinverse.

A better way to compute $\mathbf{A}^\dagger$ is to use *singular-value decomposition* (SVD) [21]. This will also yield some insight into the underlying mechanism of the inverse method. The SVD factors any matrix (real or complex) into three other matrices, as follows

$$\mathbf{A} = \mathbf{U} \times \Sigma \times \mathbf{V}^* \tag{23}$$

where **U** and **V** are orthonormal and Σ is diagonal. Then the pseudoinverse is

$$\mathbf{A}^\dagger = \mathbf{V} \times \Sigma^\dagger \times \mathbf{U}^*. \tag{24}$$

$\Sigma^\dagger$ is trivial to compute because it is diagonal; simply substitute each non-zero entry in Σ with its reciprocal.

Informally, **V** is said to contain the "input" or "analyzing" vectors of **A**, **U** is said to contain the "output" vectors of **A**, and the diagonal entries of Σ the "gains."

In terms of solving for Ambisonic decoder matrices

1. $\mathbf{V}^*$ transforms **K**, into an orientation that is symmetric about the coordinate axes, *X*, *Y*, and *Z* in the case of first order, so each can be adjusted independently, then

2. $\Sigma^{\dagger}$ adjusts the gain of each spherical harmonic, so the pressure, velocity, and so forth, are correctly reproduced, which also assures that source directions are correctly reproduced, and finally,

3. **U** returns everything to the original orientation of the speaker array.

The non-zero entries in Σ are called the *singular values* of **A**. In analyzing Ambisonic playback systems, if the number of singular values does not equal the number of signals in use, then the speaker array is not capable of reproducing all the intended spherical harmonics. This may be a trivial result, for example that a horizontal array cannot reproduce Z, or more significantly, that the array geometry is degenerate in some other way.

The ratio of the largest to the smallest singular value is called the *condition* of the matrix. This is related to the minimum and maximum values of r_E, and hence gives us insight into the overall quality of localization the array will provide. For example, in Matlab

```
>> svd(K_rect30)
ans =
   1.7321
   1.4142
   1.0000
        0
```

indicates that one spherical harmonic will not be reproduced (Z) and that $\mathbf{r_E}$ will not be uniform in all directions.

1.2. Phase-Matched Band-Splitting Filters

Sec. 4 discussed the need for frequency-dependent processing in order to transition from the exact solution at low-frequencies (LF), to one that optimizes $\mathbf{r_E}$ at high frequencies (HF). The crossover frequency use by Classic Ambisonic Decoders and in our earlier listening tests is 380 Hz. We present the design of a suitable filter for this.

The key idea is to treat the LF-to-HF transition as one would the crossover network feeding the LF and HF units in a loudspeaker. We desire a gradual transition, so simple second-order filters are used

$$LF(s) = \frac{1}{1+2sT+(sT)^2} \tag{25}$$

$$HF(s) = \frac{(sT)^2}{1+2sT+(sT)^2} \tag{26}$$

These have the -6 dB point at the crossover frequency $\omega = 1/T$ rad/sec. If you combine these, the outputs cancel at the crossover frequency, but reversing the phase of the HF section, makes its phase match that of the LF section and there is no cancellation at the crossover frequency. The output is

$$Total(s) = \frac{1-(sT)^2}{1+2sT+(sT)^2} \tag{27}$$

$$= \frac{(1+sT)(1-sT)}{(1+sT)(1+sT)} \tag{28}$$

$$= \frac{1-sT}{1+sT} \tag{29}$$

which is a first-order all-pass network. Hence, the phase response is the same as the LF section and is maintained *regardless of the relative levels of the LF and HF sections.*

Applying the bilinear transformation to implement these as digital infinite-impulse response (IIR) filters, the second-order pole

$$H(s) = \frac{1}{1+2sT+(sT)^2} \tag{30}$$

becomes

$$H(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{a_0+a_1z^{-1}+a_2z^{-2})} \tag{31}$$

where, for the LF section

$$b_0 = \frac{k^2}{k^2+2k+1} \tag{32}$$

$$b_1 = 2b_0 \tag{33}$$

$$b_2 = b_0 \tag{34}$$

$$a_0 = 1 \tag{35}$$

$$a_1 = \frac{2(k^2-1)}{k^2+2k+1} \tag{36}$$

$$a_2 = \frac{k^2-2k+1}{k^2+2k+1} \tag{37}$$

and, for the HF section

$$b_0 = \frac{1}{k^2+2k+1} \tag{38}$$

$$b_1 = -2b_0 \tag{39}$$

$$b_2 = b_0 \tag{40}$$

with a_0, a_1, a_2 as in the LF section and

$$k = tan\frac{\pi F_c}{F_s} \tag{41}$$

and F_c is the crossover frequency in Hz and F_s is the sample rate.[11]

As an example, with $F_c = 380$ and $F_s = 48000$

```
b_lp =
  0.000589143208472
  0.001178286416944
  0.000589143208472

b_hp =
  0.952044598366767
 -1.904089196733534
  0.952044598366767

a =
  1.000000000000000
 -1.902910910316590
  0.905267483150478
```

These filters can be implemented in Plogue *Bidule* as Direct Form 1 IIRs [22] using the Recursive Function block. The HF section is specified by entering

```
( 0.952044598366767 * x) +
(-1.904089196733534 * prevX(1)) +
( 0.952044598366767 * prevX(2)) -
(-1.902910910316590 * prevR(1)) -
( 0.905267483150478 * prevR(2))
```

and the LP section by entering

```
( 0.000589143208472 * x) +
( 0.001178286416944 * prevX(1)) +
( 0.000589143208472 * prevX(2)) -
(-1.902910910316590 * prevR(1)) -
( 0.905267483150478 * prevR(2))
```

Recall that the desired phase response is obtained by subtracting the output of these sections, so after scaling according to the desired response, the output signals must be differenced, not summed.

1.3. Near-Field Compensation Filter

All that is needed here is a first-order high-pass (HP) filter

$$H(s) = \frac{s}{1+sT} \tag{42}$$

which translates to the digital IIR filter

$$H(z) = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}} \tag{43}$$

[11] Note that this k is not the same as the k used in Sec. 4.

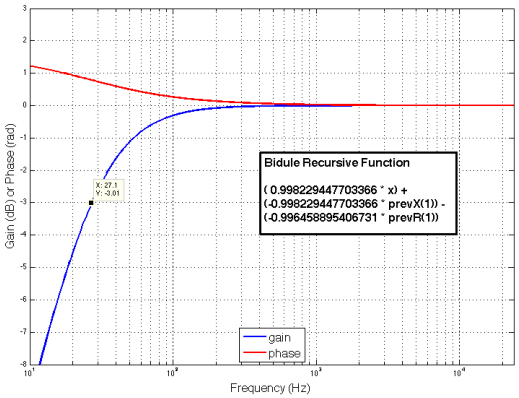


Fig. 10: Frequency and phase response of 2-meter NFC filter implemented as Direct Form 1 IIR with Bidule's recursive function block.

with

$$b_0 = \frac{1}{k+1} \tag{44}$$

$$b_1 = -b_0 \tag{45}$$

$$a_0 = 1 \tag{46}$$

$$a_1 = (k-1)b_0 \tag{47}$$

where k is given by Eqn. 41.

As an example, for 2 meters, $f_{\text{-3dB}} = 27.1$ Hz and $F_s = 48000$

```
b =
  0.998229447703366  -0.998229447703366
a =
  1.000000000000000  -0.996458895406731
```

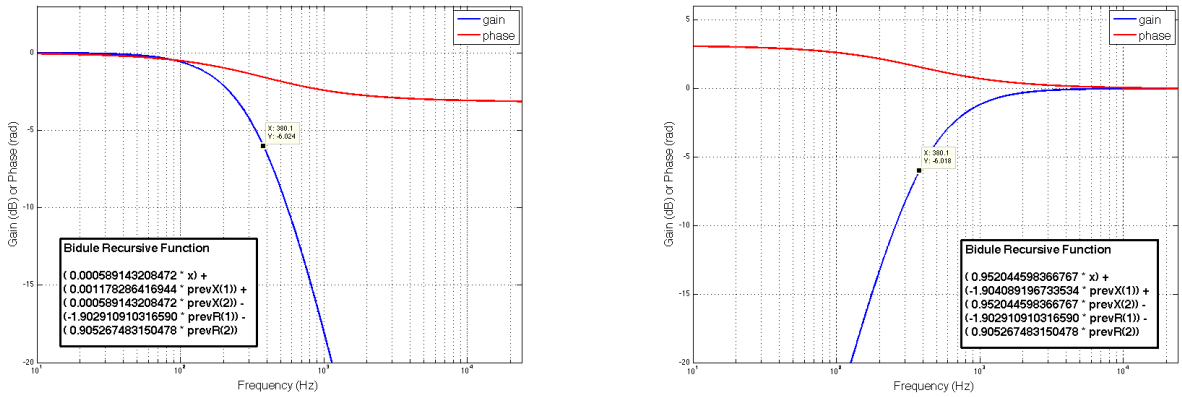
Implementing in *Bidule*'s recursive function block

```
( 0.998229447703366 * x) +
(-0.998229447703366 * prevX(1)) -
(-0.996458895406731 * prevR(1))
```

2. IS MY *ENCODER* AMBISONIC?

Many references show the first-order horizontal Ambisonic B-format encoding equations as

$$\begin{bmatrix} \mathsf{W} \\ \mathsf{X} \\ \mathsf{Y} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \cos\theta \\ \sin\theta \end{bmatrix} \mathsf{S} \tag{48}$$



(a) low-pass filter (b) high-pass filter

Fig. 9: Frequency and phase response of phase-matched 380 Hz band-splitting filters implemented as Direct Form 1 IIR with Bidule's recursive function block.

where θ is the azimuth and S is the pressure due to the source at the position of the "microphone." But this is a simplification that has led to much misunderstanding of the nature of B-format. As an example, one myth suggests that B-format does not accurately encode near or diffuse soundfields because these equations have no "phase information."

W is a perfect omni-directional pressure microphone with -3 dB gain. However, no practical microphone has a response as shown for X and Y. The correct equations regard X and Y as two perfect figure-8 particle velocity microphones with an on-axis gain of 1. As such, X and Y are subject to the variations in phase and amplitude between particle velocity and pressure encountered in real life. An important example of this is proximity [23, 15], which is a clear and unambiguous coding of distance for near sources. Cotterel [24] correctly derives XYZ as solutions of the Helmholtz wave equation. This codes near and diffuse soundfields properly and is consistent with practical implementations of the Soundfield Microphone.

The correct equation, via Beranek [23] encodes a single point source at distance d as

$$\begin{bmatrix} \mathsf{W} \\ \mathsf{X} \\ \mathsf{Y} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ D(s)\cos\theta \\ D(s)\sin\theta \end{bmatrix} \mathsf{S} \qquad (49)$$

where $D(s) = \dfrac{1+sT}{sT}$, $T = \dfrac{d}{c}$, and c is the speed of sound.

Eqn. 49 is essentially Gerzon's full expression for velocity components at the bottom of page 15, *General Metatheory of Auditory Localisation* [11], but from an encoding viewpoint. The Wave Equation precludes any practical device that implements the simplistic Eqn. 48.

Daniel [25] extends this form for higher orders but his derivation does not conveniently describe diffuse fields, standing waves or non-point sources for which we refer you to Cotterel [24]. However, a microphone with response to point sources as Eqn. 49 is necessary and sufficient to correctly encode diffuse fields, standing waves, non-point and nearby sources.