A Toolkit for the Design of Ambisonic Decoders

Aaron J. HELLER
AI Center, SRI International
Menlo Park, CA, US
heller@ai.sri.com

Eric M. BENJAMIN
Surround Research
Pacifica, CA, US
ebenj@pacbell.net

Richard LEE
Pandit Litoral
Cooktown, QLD, AU
ricardo@justnet.com.au

Abstract

Implementation of Ambisonic reproduction systems is limited by the number and placement of the loudspeakers. In practice, real-world systems tend to have insufficient loudspeaker coverage above and below the listening position. Because the localization experienced by the listener is a nonlinear function of the loudspeaker signals it is difficult to derive suitable decoders analytically. As an alternative, it is possible to derive decoders via a search process in which analytic estimators of the localization quality are evaluated at each search position. We discuss the issues involved and describe a set of tools for generating optimized decoder solutions for irregular loudspeaker arrays and demonstrate those tools with practical examples.

Keywords

Ambisonic decoder, HOA, periphonic, nonlinear optimization

1 Introduction

Ambisonics is a versatile surround sound recording and reproduction system. One of the attractions is that the transmission format is independent of the loudspeaker layout. However, this means that each playback system needs a custom decoder that is matched to the loudspeaker array. The decoder creates the loudspeaker signals from the transmission signals. Ambisonics theory provides simple encapsulations of high- and low-frequency auditory localization that can be used to design decoders, as well as theorems that ease the design of decoders for regular polygonal and polyhedral loudspeaker arrays.

In earlier papers, the present authors have discussed the design and testing of first-order decoders for regular horizontal loudspeaker layouts [Heller et al., 2008] as well as the use of nonlinear optimization to design decoders for ITU 5.1 arrays [Heller et al., 2010]. In this paper, we extend that work to full periphonic (3-D) arrays and higher-order Ambisonics (HOA). The techniques are implemented as a MATLAB [2011] and GNU Octave [Gnu, 2011] toolkit that makes use of the NLOpt library [Johnson, 2011] to perform the optimization.

We use the term *decoder* to mean the configuration for a decoding engine that does the actual signal processing. Examples are Ambdec [Adriaensen, 2011] that operates in real time, as well as an offline decoder we have implemented as part of this toolkit.[1]

In this paper, we use bold roman type to denote vectors, italic type to denote scalars, and sans serif type to denote signals. A scalar with the same name as a vector denotes the magnitude of the vector. A vector with a circumflex ("hat") is a unit vector, so, for example, $\hat{\mathbf{r}}_{\mathbf{E}} = \mathbf{r}_{\mathbf{E}}/r_E$.

We start with a discussion of the design process and the tradeoffs involved, then the specifics of the optimization process, and finally results of two arrays, a third-order decoder for the 22-loudspeaker CCRMA array, and a second-order decoder for the 12-loudspeaker 30° tri-rectangle array.

2 Designing Ambisonic Decoders

Ambisonics represents a sound field with a group of signals that are proportional to spherical harmonics. The original Ambisonic systems were first order only, but more recently, higher-order sys-

[1] Another important function of an Ambisonic decoder is to provide *near-field compensation*. This compensates for the curvature of the wavefronts due to the finite distance to the loudspeaker and is strictly a function of distance of the speaker from the center of the array and the order of reproduction. Ambdec and the offline decoder in this toolkit provide such filters and they will not be discussed further in this paper.

tems have been implemented. In first-order Ambisonics the zeroth-order component represents the sound pressure, and the three first-order components represent the acoustic particle velocity. If these components are reproduced exactly, then the sound will be correct at the center. However, it is not possible to get the first-order components correct except at a single point and not practical to get them correct at higher frequencies, where the wavelengths become smaller than the size of the human head. The task of the decoder is to create the best *perceptual impression* that the soundfield is being reproduced accurately given the loudspeaker array being used.

In practical terms, the following are necessary:

- Constant amplitude gain for all source directions
- Constant energy gain for all source directions
- At low frequencies, correct reproduced wavefront direction and velocity
- At high frequencies, maximum concentration of energy in the source direction
- Matching high- and low-frequency perceived directions

These criteria may, themselves, have different interpretation or importance depending on the source material and the intended use. We can identify three distinct types of program:

- Natural recordings made with a first-order soundfield microphone.
- Natural recordings made with higher order microphones. As of this writing, such microphones are just becoming available commercially, but practical constraints will mean that these are still first order at lower frequencies.
- Artificial recordings. First order as well as Higher Order Ambisonic (HOA) program material.

The first case is Ambisonic's greatest strength. Good first-order Ambisonic reproduction is probably the closest to recreating a virtual sound environment, whether the buzz of a busy Asian marketplace or the sound of a concert in a good hall in your living room. It will most likely be used to create realistic atmosphere even if more precise methods like HOA are used for special sounds.

To preserve this advantage requires the preservation of a good facsimile of the diffuse field. Energy gain that varies with direction and "bunching" of directions, particularly in the horizontal plane, are all detrimental, as is "speaker detent" where individual loudspeakers draw attention to themselves.

2.1 Auditory Localization

Due to the range of wavelengths involved, the human auditory localization mechanism utilizes different directional cues over different frequency regimes. At low frequencies, localization depends on the detection of Interaural Time Differences (ITDs), but at high frequencies there is an ambiguity because a human head is multiple wavelengths across above about 1 kHz. For this reason, localization switches abruptly, depending on Interaural Level Differences (ILDs) above that frequency. One way to predict localization would be to use Head Related Transfer Functions (HRTFs) to calculate the actual ear signals of a listener, but this turns out to be computationally difficult and would vary from listener to listener.

Gerzon developed a series of metrics for predicting localization that are simpler than using the HRTFs [Gerzon, 1992]. The simplest of these metrics are the velocity localization vector, $\mathbf{r_V}$, and the energy localization vector, $\mathbf{r_E}$. The direction of each indicates the direction of the expected localization perception, while the magnitude indicates the quality of the localization. In natural hearing from a single source, the magnitude of each vector should be exactly 1, and the direction of the vectors is the direction to the source. It should be noted that, while $\mathbf{r_V}$ is proportional to the physical quantity of the acoustic particle velocity, $\mathbf{r_E}$ is an abstract construct.[2]

Following Gerzon [1992], the pressure (amplitude gain), P, and total energy gain, E, are

$$P = \sum_{i=1}^{n} G_i \tag{1}$$

[2]Note that these metrics are not specific to Ambisonics; they can be used to predict the quality of the phantom images produced by any multispeaker reproduction system, regardless of the panning laws used, including plain old two-channel stereo. Gerzon shows this for several well-known stereo phenomena [Gerzon, 1992].

$$E = \sum_{i=1}^{n} (G_i {G_i}^*) \tag{2}$$

The magnitude and direction of the velocity vector, r_V and $\hat{\mathbf{r_V}}$, at the center of an array with n loudspeakers is

$$r_V \hat{\mathbf{r_V}} = \frac{1}{P} \mathrm{Re} \sum_{i=1}^{n} G_i \hat{\mathbf{u_i}} \tag{3}$$

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

$$r_E \hat{\mathbf{r_E}} = \frac{1}{E} \sum_{i=1}^{n} (G_i {G_i}^*) \hat{\mathbf{u_i}} \tag{4}$$

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

The velocity vector points in the same direction and is proportional to the acoustic particle velocity. It has been shown that the velocity vector predicts the ITDs very accurately [Benjamin et al., 2010]. The energy vector predicts the ILDs, but in practice it is not possible to get $r_E = 1$ unless the sound is coming from just one loudspeaker. This is representative of a pervasive problem in multichannel sound reproduction. The maximum average value of r_E that can be obtained for a given Ambisonic order is shown in Figure 1. The formulas to compute these are given in Appendix A.

Because different sets of gains are needed to satisfy the low- and high-frequency models, many ambisonic decoders split the audio into two bands, apply different decoder matrices, and then recombine to produce the loudspeaker signals.[3] Daniel has suggested that a three-band decoder may provide better reproduction under some listening conditions [Daniel, 2001]. This remains an open question at this time.

2.2 Computing the Low-Frequency Matrix

The low-frequency matrix provides gains from each channel of the ambisonic program material

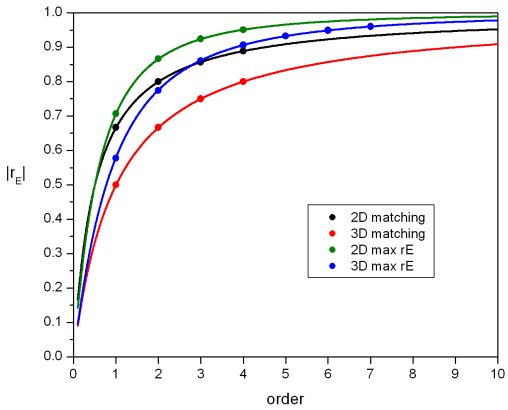


Figure 1: Maximum average r_E depending on order and type. "matching" and "max r_E" refer to the decoder matrices described in Sections 2.2 and 2.3, respectively.

to each loudspeaker that are needed to optimize localization as predicted by the velocity localization vector, $\mathbf{r_V}$. Numerous authors have provided derivations of the low-frequency solution for a given loudspeaker array, and thus a number of different terms are used to refer to it, including "velocity", "matching", "basic", "exact", "mode matching", "re-encoding" and so forth.

In practice, these reduce to projecting (or encoding) the loudspeaker directions onto the selected spherical harmonic basis set,[4] assembling these vectors into an array, and computing the Moore-Penrose pseudoinverse of the array [Weisstein, 2008]. Examples of this can be found in Appendix A of [Heller et al., 2008]. In general, there are an infinite number of solutions and this procedure provides the solution with the minimum L2-norm (i.e., the least-squares fit), which has the desirable property of requiring the minimum total radiated power.[5]

[3] This places certain constraints on the phase response of the band splitting filters. We discuss the design and implementation of suitable filters in Appendix B of [Heller et al., 2008] and note that the filters in Ambdec meet these requirements.

[4] The toolkit is neutral as to the conventions for component ordering and normalization. These conventions are encapsulated in a single function. The current implementation supports the Furse-Malham set [Malham, 2003], but others can be added easily.

[5] Recently, some authors, drawing upon compressive sending theory, have suggested that the L1-norm may be more suitable [Wabnitz et al., 2011; Zotter et al., 2012]. L1-norm minimizes the sum of absolute errors. Compared to least-squares, it allows larger maximum errors in exchange for more zero errors.

Except in the case of degenerate configurations, where all the loudspeakers lie in the null of one or more of the spherical harmonics, this procedure will result in a decoder matrix that satisfies the low-frequency localization criteria exactly; however, it may utilize a great deal of power to get them correct in directions where there is a large angular separation between the loudspeakers in the array. This will result in low r_E values in those directions. As we shall see, except in the case of regular polyhedra and polygons, it is impossible to fully satisfy all the ambisonic criteria simultaneously. This implies that while the ambisonic transmission format is independent of the loudspeaker array, not all loudspeaker array geometries perform equally well.

2.3 Computing the High-Frequency Matrix

The high-frequency matrix provides gains from each channel of the ambisonic program material to each loudspeakers that are needed to optimize localization as predicted by the energy localization vector, $\mathbf{r_E}$. Gerzon proved two theorems for first-order reproduction, the polygonal decoder theorem and the diametric decoder theorem. They state that in an array with a minimum of four loudspeakers for 2-D and six speakers for 3-D, where the loudspeakers are spaced in equal angles or in diametrically opposed pairs, $\mathbf{r_E}$ is guaranteed to point in the same direction as $\mathbf{r_V}$. The polygonal decoder theorem also holds for higher-order Ambisonic reproduction, provided there is an adequate number of loudspeakers in the array to support the desired order. This simplifies the task of designing the high-frequency matrix to that of selecting the gain for each order such that the overall magnitude of $\mathbf{r_E}$ is maximized. For first-order decoders, Gerzon provided the values of $\frac{\sqrt{2}}{2}$ for horizontal arrays and $\frac{\sqrt{3}}{3}$ for periphonic arrays. Daniel derived general formulas for these gains [Daniel, 2001], which are given in Tables 1 and 2. (See Appendix A for programs that compute the values in these tables.)

As we will see in the example in Section 2.5, once the array deviates from having equal angles, there is no longer a guarantee that $\mathbf{r_E}$ and $\mathbf{r_V}$ point in the same direction or that there is a single set of gains that maximize r_E in every direction. Because of this, we must trade off the various criteria and due to the nonlinear nature of the criteria, numerical optimization is needed to compute the solutions, which will be discussed in Section 3.

2.4 Merging the LF and HF Matrix

The existence of different optimum decoder coefficients for optimum $\mathbf{r_V}$ and $\mathbf{r_E}$ would typically mean having to make a choice or compromise between the two. In this case, however, both can be had. The decoder that optimizes $\mathbf{r_V}$ can be used at low frequencies and the decoder that maximizes $\mathbf{r_E}$ can be used at high frequencies, by the simple expedient of using filters to cross over between the two. This is typically done at around 400 Hz.

Because the higher-order components are reduced in order to maximize $\mathbf{r_E}$, this causes a reduction in the total signal level of the high-frequency decoder outputs, and thus a reduction in the high frequencies heard by the listener. The gains that maximize r_E specify the relationship among the signals of different order, but not how that gain should be apportioned between high-frequency cuts and low-frequency boosts. There are three possibilities:

1) Preservation of the amplitude. That is, simply use the gains produced by the optimizer or those given in Tables 1 and 2.
2) Preservation of the root-mean-square (RMS) level. This is what Gerzon [1980] suggests and is what is implemented in older analog decoders.
3) Conservation of the total energy. Daniel [2001] suggests this, and the configuration files included with Ambdec follow this recommendation. This method results in more high frequencies with more speakers.

The calculations involved are given in Appendix B. In listening tests, we have found that preservation of the RMS level works well for small arrays. We have also found that using the conservation of energy approach on large 3-D arrays results in overemphasizing high frequencies and near-head imaging artifacts and nulls. In practice, we the set the LF/HF balance by ear, comparing the balance of the two-band decoder to that of a single-band r_E-max decoder. More work is needed to find a procedure for this that does not involve tuning by ear.

Order	Max r_E	Gains
1	0.707107	1, 0.707107
2	0.866025	1, 0.866025, 0.5
3	0.92388	1, 0.92388, 0.707107, 0.382683
4	0.951057	1, 0.951057, 0.809017, 0.587785, 0.309017
5	0.965926	1, 0.965926, 0.866025, 0.707107, 0.5, 0.258819

Table 1: Per-order gains for max-r_E decoding with 2-D regular polygonal arrays.

Order	Max r_E	Gains
1	0.57735	1, 0.57735
2	0.774597	1, 0.774597, 0.4
3	0.861136	1, 0.861136, 0.612334, 0.304747
4	0.90618	1, 0.90618, 0.731743, 0.501031, 0.245735
5	0.93247	1, 0.93247, 0.804249, 0.62825, 0.422005, 0.205712

Table 2: Per-order gains for max-r_E decoding with periphonic regular polyhedral arrays.

2.5 Selection of a speaker array

Due to symmetry, regular loudspeaker arrays have the advantage of uniformity in the localization predictors $\mathbf{r_V}$ and $\mathbf{r_E}$. As noted above, practical difficulties usually prevent the attainment of a completely regular array. There will be a tendency for r_E to be greater in the directions where the angular density of loudspeakers is greater, and less in the directions where there are few loudspeakers and $\mathbf{r_E}$ will tend to point in the directions of concentrations of loudspeakers.

It should be noted at this point that it is impossible to get r_E to be larger in the direction between loudspeakers than the value achieved simply by driving the loudspeakers nearest to the gap equally. This means that the best that can be achieved by an Ambisonic decoder is to have a smooth transition between areas where the performance is good (large number of loudspeakers, high magnitude of r_E and $\mathbf{r_E}$ points in the intended direction) and areas where the performance is less good (fewer loudspeakers, $\mathbf{r_E}$ has small magnitude and points in an incorrect direction). As such, we must be careful in choosing the decoder parameters so that the performance in the good directions is good enough, and the performance in the poor directions is not too bad.

A simple example of a four-speaker array will illustrate these difficulties. A square horizontal array has a basic decoder solution of

$$\begin{bmatrix} \mathsf{LF} \\ \mathsf{RF} \\ \mathsf{RR} \\ \mathsf{LR} \end{bmatrix} = \frac{\sqrt{2}}{4} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 \\ 1 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathsf{W} \\ \mathsf{X} \\ \mathsf{Y} \\ \mathsf{Z} \end{bmatrix} \tag{5}$$

This gives exact recovery of the pressure and velocity at the center of the array: $|\mathbf{r_V}| = 1$ and points in the intended direction. But because the angular separation of the loudspeakers is 90°, $r_E = \frac{2}{3}$. However, if we investigate what happens as the ratio of pressure (W) and velocity (X, Y and Z) is varied, it develops that r_E is maximum for the case where the first-order components are reduced to $\frac{\sqrt{2}}{2}$ of their original value. This gives a magnitude of $\mathbf{r_E}$ of $\frac{\sqrt{2}}{2}$.

If the square is replaced with a rectangle with an aspect ratio of $\sqrt{3} : 1$, the front and rear loudspeakers now subtend an angle of 60° and the side loudspeakers subtend an angle of 120°. This reduces r_E at the sides but increases it in the front, relative to a square. If the same gain as derived for the square ($\frac{\sqrt{2}}{2}$ for the first-order components) is applied, then there is a substantial improvement in r_E to the sides, and a very tiny decrease in r_E in the front. This is shown in Figure 2.

But is this the "optimum"? Figure 3 shows that if we further vary the ratio of the zero- and first-order components it develops that r_E, evaluated at the sides, is a maximum at a different ratio.

It is thus possible to maximize r_E in front *or* at

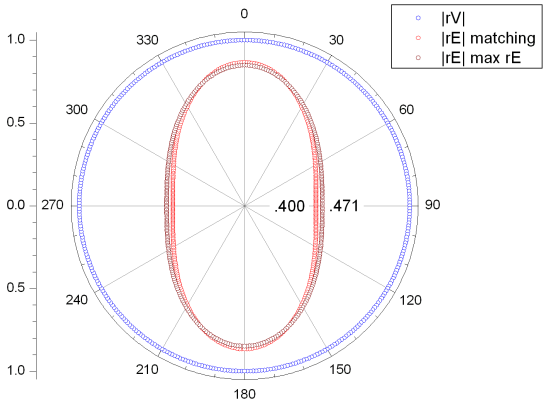


Figure 2: Locus of $\mathbf{r_E}$ for a rectangular array, matching and "max r_E" decoders.

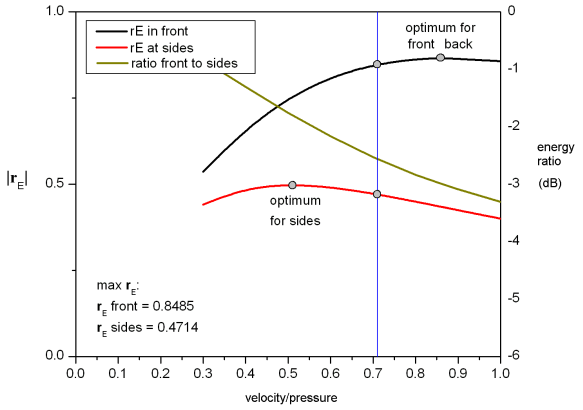


Figure 3: r_E and the energy, E, as a function of the ratio of the first-order to zero-order scaling.

the sides, but not both at once. One might wish to use a different decoder depending on whether sound images are expected at the front, or on the sides, or a decoder that gives a compromise between the two.

Thus far, only the quality and direction of the localization have been discussed. There is also an effect on the loudness of the sound, depending on direction. If the loudspeaker array is irregular, then the solution to recover pressure and velocity results in an increase in energy in the directions where the angular spacing is greatest. This results in an increase in reproduced loudness in those directions.

For the previous example of a rectangular array with a $\sqrt{3}:1$ aspect ratio, the ratio of the energy in the forward direction to the energy at the sides was calculated and is also plotted in Figure 3. r_V obtains its correct value of 1 in all directions and the pressure response is also omnidirectional. At higher frequencies, where the "max-r_E" decoder is in effect, r_E is maximized for front and back directions (where the speakers are closer together). At these frequencies, sounds from the sides (perceived as "energy") are 2.4 dB louder. This is a pervasive problem for irregular arrays and will be addressed in greater detail below. The simple $\sqrt{3}:1$ rectangle as above is used widely and is known to give good results. On the other hand, listening tests indicate that the 5.5 dB energy imbalance exhibited by some first-order decoders for ITU 5.1 arrays is too large. From this, we propose 3 dB variation in "energy" with horizontal direction as the maximum imbalance acceptable.

2.5.1 Discussion of compromises of speaker arrays

The selection of a loudspeaker array for Ambisonic reproduction is subject to a number of compromises, notably the space available to house the array and the budget for purchasing loudspeakers. It may be that the array already exists, in which case the decoder design task is one of selecting a decoder design that provides the best audible performance. In other situations, however, the design of the array has not been fixed although the number of loudspeakers may have been. In that case, there is substantial latitude to trade off between high-order performance horizontally and periphonic performance.

3 Optimizer

As noted in Section 2.5, in an irregular array, simply scaling the LF and HF matrices does not result in $\mathbf{r_V}$ and $\mathbf{r_E}$ pointing in the same direction; hence, the design procedure becomes somewhat more complex.

Because the key psychoacoustic criteria for good decoder performance are nonlinear functions of the speaker signals, we utilize numerical optimization techniques. To do this, a single objective function is formulated that takes as input the decoder matrix and produces a single figure of merit that decreases as the decoder performance improves. The nonlinear optimization algorithm

will then try different sets of matrix elements, attempting to arrive at the lowest value possible. Because there are a number of criteria, we use the weighted sum to provide an overall figure of merit. A user can adjust the weights to set the relative importance of the different criteria, say uniform energy gain (loudness) versus angular accuracy. In addition, each test direction can have its own set of weights, so that, for example, angular accuracy can be emphasized for the front, while uniform energy gain is emphasized in other directions. This might be the preferred configuration for classical music recorded in a reverberant performance hall. On the other hand, environmental recordings made outdoors have very little diffuse content, so overall angular accuracy is more important. Another application of direction weightings is in highly asymmetrical arrays, such as a dome, where few speakers are below the listener. In this case, we expect poor performance in those directions, so they are deemphasized when computing the objective function.

We have employed the NLOpt library for nonlinear optimization [Johnson, 2011]. NLOpt provides a common application programming interface (API) for a collection of nonlinear optimization techniques. In particular, it supports a number of "derivative free" optimization algorithms, which are well suited to the current application where the objective function is the result of a computation, rather than an analytic function.

An earlier version of the optimizer that was limited to first-order horizontal arrays was written in C++ [Heller et al., 2010]. To extend that to higher-order and periphonic arrays required a significant rewrite, so an initial prototype was written in MATLAB, with plans to recode in C++. Because the bulk of the computation is matrix multiplication, which is handled by highly optimized code in MATLAB, it turned out that the execution speed was almost as fast as the original C++ version, so we abandoned plans for the rewrite. To make the code widely usable, it was kept compatible with GNU Octave. The key change is that GNU Octave does not support nested functions, so a number of variables need to be declared global to make them accessible to the objective function.

3.1 Optimization Criteria

For each test direction, the following are computed: amplitude gain, P, energy gain, E, the velocity localization vector, $\mathbf{r_V}$, and the energy localization vector, $\mathbf{r_E}$. From these, we compute the pairwise angles between the test direction, $\hat{\mathbf{r_V}}$, and $\hat{\mathbf{r_E}}$. These are summarized with the following figures of merit: deviation of amplitude gain from 1 along the x-axis, minimum, maximum, and RMS values of amplitude gain, energy gain, magnitude of $\mathbf{r_V}$, magnitude of $\mathbf{r_E}$, and the pairwise angular deviations.

It is important that the criteria are "well behaved" near zero, so as not to trigger oscillating behavior in the optimizer. They should be continuous and have first derivatives. In practical terms, absolute value and thresholds should not be used; squaring can be used for the former and the exponential function for the latter cases.

Finally, directional weightings are applied to each criteria and then an overall weighted sum produces the single figure of merit for that particular configuration.

3.2 Test Directions

As mentioned in the previous section, each candidate set of parameters is evaluated from a number of directions. For 2-D speaker arrays, 180 or 360 evenly spaced directions are often used [Wiggins et al., 2003; Moore and Wakefield, 2008]. For 3-D arrays, the situation is more complex because no more than 20 points can be distributed uniformly on a sphere (a dodecahedron).

Lebedev-Laikov quadrature defines sets of points on the unit sphere and weights with the property that they provide exact results for integration of the spherical harmonics [Lebedev and Laikov, 1999]. The current implementation provides a function that returns Lebedev grids of points and corresponding weights for as many as 5810 directions. Our current experiments have used a grid with 2702 points, which corresponds roughly to 3°. The toolkit also has functions providing 2-D and 3-D grids that are sampled in uniform azimuth and elevation increments, which are useful for visualization of the results.

3.3 Optimization Behavior

As part of the optimization setup, the user supplies a set of stopping criteria. This can be specified as a threshold on the absolute and relative

changes in the parameters and/or the objective function, as well as a maximum running time and maximum number of iterations. The default values in the current implementation are 1×10^{-7} for the parameters and objective function.

For small 2-D arrays (say, 12 to 24 parameters), the optimizer typically converges in less than 1 minute, examining 40,000 to 1,500,000 configurations. For large high-order arrays (say, 200 to 400+ parameters), it typically converges in less than an hour. These timings were done with Octave version 3.2.4-atlas on a 2.66 GHz Intel Core i7 with 8 GB of memory. The bulk of the computation comprises matrix multiplications and is therefore suitable for parallel implementations. The timings in MATLAB were approximately 2x faster than Octave since it can make use of the multiple cores in the i7 processor.

With large optimization problems, using a local optimization algorithm and providing an initial solution that is near to the optimum is important for reliable convergence. The toolkit currently supports three strategies:

- Using the low-frequency solution modified with the per-order gains that would provide the max-r_E solution for a uniform array.
- "Musil Design" where additional "virtual" loudspeakers are inserted into the array to make the spacing more uniform, and hence more suited to a pseudo-inverse solution. After the optimization is complete, the signals for the virtual speakers are either ignored or distributed to the adjacent speakers [Zotter et al., 2010].
- A hierarchical approach, decomposing the optimization by establishing a solution for each order consecutively, freezing the individual coefficients for orders below the current one, but allowing an overall adjustment on the gain of the lower orders.

4 Examples

The software tools described above were applied to the derivation of decoders for several real-world systems. The examples given here are the CCRMA listening room[6] and a tri-rectangle with the upper and lower loudspeakers at ±30° with respect to horizontal.

4.1 CCRMA Listening Room

The described software was applied to deriving decoders for the Listening Room at CCRMA (Center for Computer Research in Music and Acoustics) at Stanford. This facility consists of 22 identical loudspeakers arranged in five rings. There is a horizontal ring of eight loudspeakers, two rings of six loudspeakers, one 50° below and one 40° above horizontal, and one loudspeaker directly above and one directly below the listening position. The two hexagonal rings are thus not exactly horizontally opposed. A schematic of the array is shown in Figure 4a.

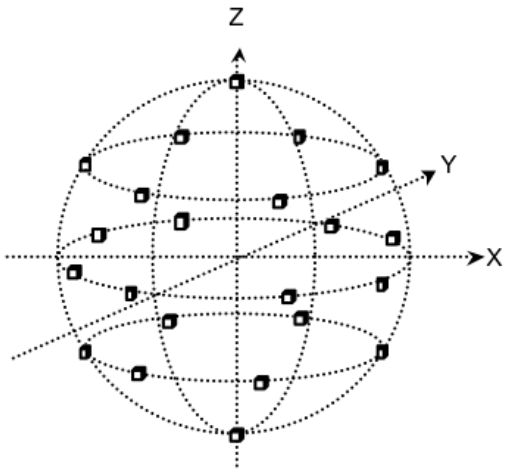
An initial solution was derived by calculating the pseudoinverse of the loudspeaker projection matrix as described above. The decoder was modified to optimize the magnitude of r_E at high frequencies by applying the weighting factors given in Table 2 to the gains of the signal components of each order. Given that the theoretical maximum average value for r_E can be no greater than 0.866 at third order, the average value of 0.850 for the third order decoder given here does not leave a great deal of margin for improvement. Nonetheless, the optimization software was applied to the problem. Figure 5 shows the performance of the initial solution and the optimized result. Average r_E was increased slightly, and maximum directional error reduced by a factor of 5.

An informal listening test comparing this decoder to the existing one was conducted using third-order test signals and studio recordings, as well as first-order acoustic recordings. The general impression was that the new decoder did a better job of keeping horizontal sources in the horizontal plane, whereas the existing decoder rendered such sources above the horizontal plane.

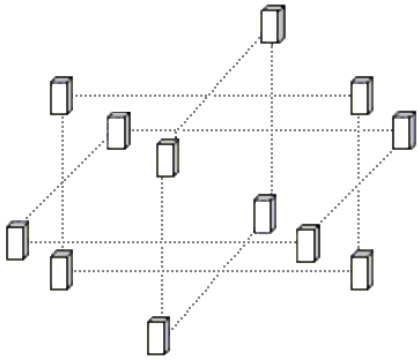
4.2 The 30° Tri-Rectangle

As discussed elsewhere, a dodecahedron or other large regular array is difficult to fit into normally dimensioned spaces. One large array that does fit into normal spaces is the so-called *tri-rectangle*, patterned after a suggestion by Gerzon. A schematic is shown in Figure 4b. It consists of three interlocking rectangles of loudspeakers, one in the horizontal plane, one in the XZ plane, and one in the YZ plane. The projection of the loud-

[6] See https://ccrma.stanford.edu/room-guides/listening-room/

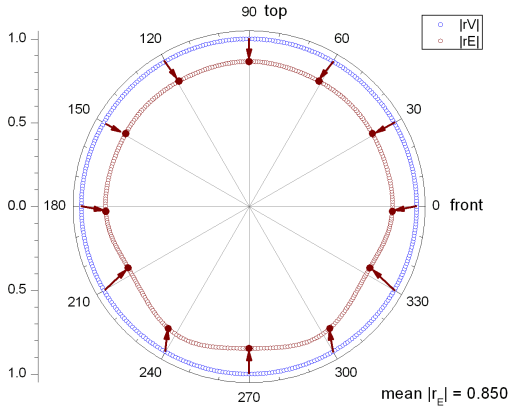


(a) The 22-loudspeaker array at CCRMA

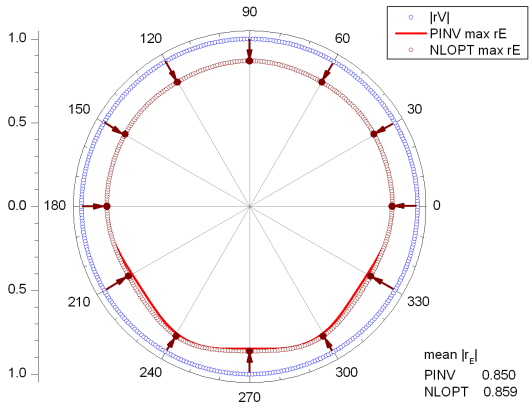


(b) The 12-loudspeaker tri-rectangle.

Figure 4: Schematics of the loudspeaker arrays used in the examples.



(a) The initial solution calculated by pseudoinverse and max-r_E gains.



(b) The optimized solution.

Figure 5: $\mathbf{r_E}$ in the vertical plane for the CCRMA array before and after optimization. The arrows show the directional error between the low- and high-frequency matrices. In this case, average r_E was increased slightly, from 0.85 to 0.86 and the maximum directional error reduced by a factor of 5.

speakers into any plane is an octagon, which hints at its utility for reproducing second-order program material. However, to enable it to fit into typical spaces the vertical rectangles must be squashed to an approximate $\pm 30°$ vertical angle. This gives a solid angle of $120°$ above and below the listening position with no loudspeakers. Naturally, this has a profound effect on the localization for sources above and below the listening position.

Performance of the initial solution by inversion is shown in Figure 6. The magnitude of r_E is in red, with both the horizontal and vertical (in the XZ plane) shown. The horizontal shape is essentially circular, with perfect direction (not shown in the figure), but the magnitude of $\mathbf{r_E}$ decreases dramatically for sources above or below about $\pm 30°$ of elevation. Furthermore, there is an increasing error in the direction of $\mathbf{r_E}$ indicating that high-

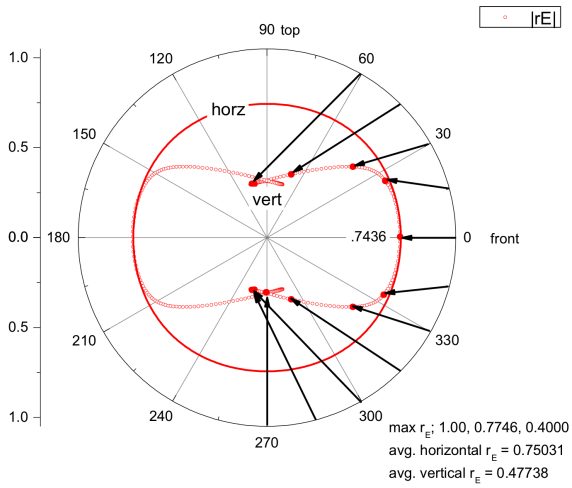


Figure 6: $\mathbf{r_E}$ in the horizontal and vertical planes for the initial second-order decoder.

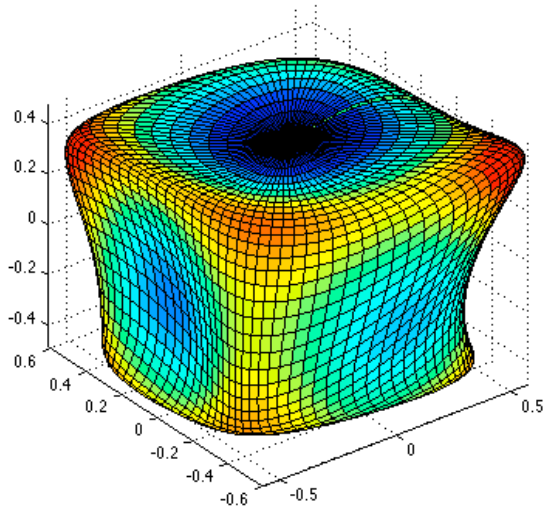


Figure 7: 3-D plot of $\mathbf{r_E}$ from an unconstrained optimization of the second-order decoder for the 30° tri-rectangle.

frequency sounds will be perceived as coming from near the poles. The extreme errors in the direction of r_E are compounded by the low values, making localization in the up and down directions vague in any case.

The large angle subtended by the loudspeaker placement with respect to the vertical axis makes it impossible to get precise localization for sources directly above or below the listening position. It may, however, be possible to improve the localization for sources near horizontal by correcting the direction of r_E.

Running the optimizer with this configuration as an initial solution resulted in a highly distorted solution where the sounds are drawn strongly to the loudspeakers. A 3-D plot of $\mathbf{r_E}$ for this solution is shown in Figure 7. As can be seen, the performance is very non-uniform (a sphere would be ideal) and the maximum angular error is over 30°.

Next, a Musil design was attempted. Virtual loudspeakers were inserted into the array directly above and below the center. This was optimized and then the signals for the virtual loudspeakers reassigned to the nearest real loudspeakers. This resulted in improved r_E in the horizontal plane, as well as elevations as high as ±30°; however, it suffers from directional errors as large as 31°. Figure 8a shows the optimized solution.

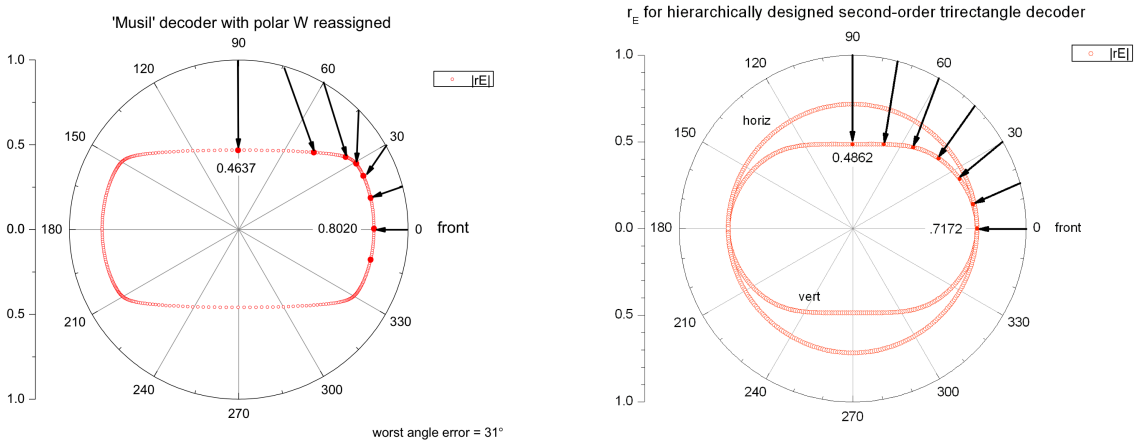
Finally, a hierarchical design was attempted, where each subsequent order is optimized separately. This resulted in a slightly lower r_E, but significantly reduced angular error in the vertical plane. Figure 8b shows the optimized solution.

5 Conclusions

An open source package for the design of ambisonic decoders has been presented. The software allows the derivation of decoders for arbitrary loudspeaker arrays, 2-D or 3-D. The software operates under Octave or MATLAB, with the nonlinear optimization performed by the open source package NLOPT. Auditory localization at middle and high frequencies is a nonlinear function of the loudspeaker signals, which necessitates the finding of solutions that work well for those frequencies via an optimization process.

Two example systems were solved. The first was a third-order decoder for the 22-loudspeaker CCRMA listening room. That system is nearly regular, and it was found that a solution obtained by inversion of the loudspeaker matrix, with per-order gains, was nearly as good as one obtained by the nonlinear optimization process. Nonetheless, the magnitude of $\mathbf{r_E}$ was improved and the angle error was reduced.

The second system was a 12-loudspeaker tri-rectangle, with the upper and lower loudspeakers at 30° above and below the horizontal plane. A decoder derived for that system via the technique



(a) Musil design.

(b) Hierarchical design.

Figure 8: $\mathbf{r_E}$ in the vertical plane for second-order decoders for the 30° tri-rectangle.

of inversion followed by per-order gains shows high magnitudes of r_E in the horizontal plane but low magnitudes in the polar regions and large errors in the direction of $\mathbf{r_E}$.

Two additional methods were tried in a search for a superior solution. The first was the Musil decoder in which the array was filled out with virtual loudspeakers at the poles and the signals for those speakers are routed to the nearest real speakers.

The second method was a hierarchical one in which a solution for each order was established consecutively, such that a higher-order decoder is also optimum for lower-order program sources. This results in a very well behaved decoder, but with slightly lower values of r_E.

6 Acknowledgements

The authors thank Fernando Lopez-Lezcano for encouraging us to write this paper and for posing the problem of deriving a third-order decoder for the CCRMA listening room; Andrew Kimpel and Elan Rosenman for stimulating discussion and access to their second-order periphonic and fifth-order horizontal loudspeaker arrays; and Isaac Heller for the idea of using the exponential function to implement soft thresholds.

References

Fons Adriaensen, 2011. *AmbDec User Manual*, 0.4.3 edition. http://kokkinizita.linuxaudio.org/ambisonics/. 1

Eric Benjamin, Richard Lee, and Aaron Heller. 2010. Why Ambisonics Does Work. In *AES 129th Convention*, San Francisco, November. 3

Jerome Daniel. 2001. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*. Ph.D. thesis, University of Paris. 3, 4

Michael A. Gerzon. 1980. Practical Periphony: The Reproduction of Full-Sphere Sound. In *65th Audio Engineering Society Convention Preprints*, number 1571, London, February. AES E-lib http://www.aes.org/e-lib/browse.cfm?elib=3794. 4

Michael A. Gerzon. 1992. General Metatheory of Auditory Localisation. In *92nd Audio Engineering Society Convention Preprints*, number 3306, Vienna, March. AES E-lib http://www.aes.org/e-lib/browse.cfm?elib=6827. 2

2011. Octave. http://www.octave.org/ (accessed July 1, 2011). 1

Aaron J. Heller, Richard Lee, and Eric M. Benjamin. 2008. Is My Decoder Ambisonic? In *AES 125th Convention*, number 7553. 1, 3

Aaron Heller, Eric Benjamin, and Richard Lee. 2010. Design of Ambisonic Decoders for Irregular Arrays of Loudspeakers by Non-Linear Optimization. In *AES 129th Convention*, San Francisco. 1, 7

Steven G. Johnson. 2011. The nlopt nonlinear- optimization package. http://ab-initio.mit.edu/nlopt. 1, 7

V.I. Lebedev and D.N. Laikov. 1999. A Quadrature Formula for the Sphere of the 131st Algebraic Order of Accuracy. *Doklady Mathematics*, 59(3):477–481. 7

David G Malham. 2003. Space in music - music in space: Higher order ambisonic systems. Master's thesis, University of York. 3

MATLAB. 2011. *version 7.13.0 (R2011b)*. The MathWorks Inc., Natick, Massachusetts. 1

David Moore and Jonathan Wakefield. 2008. The Design of Ambisonic Decoders for the ITU 5.1 Layout with Even Performance Characteristics. In *124th Audio Engineering Society Convention Preprints*, number 7473, Amsterdam, May. 7

Andrew Wabnitz, Nicolas Epain, A van Schaik, and C Jin. 2011. Time Domain Reconstruction of Spatial Sound Fields using Compressed Sensing. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 465–468. IEEE. 3

Eric W. Weisstein. 2008. Moore-Penrose Matrix Inverse. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html. 3

Bruce Wiggins, Iain Paterson-Stephens, Val Lowndes, and Stuart Berry. 2003. 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. Availible from http://sparg.derby.ac.uk/SPARG/PDFs/SPARG_UKSIM_Paper.pdf (accessed May 15, 2006). 7

Franz Zotter, H. Pomberger, and M. Noisternig. 2010. Ambisonic Decoding with and without Mode-Matching: A Case Study Using the Hemisphere. *2nd Ambisonics Symposium, Paris*. 8

Franz Zotter, H. Pomberger, and M. Noisternig. 2012. Energy-Preserving Ambisonic Decoding. *Acta Acustica united with Acustica*, 98(1):37–47, January. 3

A Formulas for maximum average r_E and per-order gains

A.1 Horizontal Arrays

For regular horizontal arrays (Table 1), the maximum value of r_E and the gains for each Ambisonic order, M, are given by

$$r_E = \text{largest root of } T_{M+1}(x) \tag{6}$$

$$g_m = T_m(r_E),\ m = 0 \dots M \tag{7}$$

where T_m is the m^{th} Chebyshev polynomial of the first kind. In Mathematica, this can be written as[7]

```
Table[ChebyshevT[Range[0, M],
  x /. FindRoot[ChebyshevT[M+1,x], {x,1}]],
  {M, 1, 5}]
```

A.2 Periphonic Arrays

For regular periphonic arrays (Table 2), the maximum value of r_E and the gains for each Ambisonic order, M, are given by

$$r_E = \text{largest root of } P_{M+1}(x) \tag{8}$$

$$g_m = P_m(r_E),\ m = 0 \dots M \tag{9}$$

where P_m is the m^{th} Legendre polynomial. In Mathematica, this can be written as

```
Table[LegendreP[Range[0, M],
  x /. FindRoot[LegendreP[M+1,x], {x,1}]],
  {M, 1, 5}]
```

B LF/HF Matching

As mentioned in Section 2.4, there are three approaches to adjusting the g_m to match LF/HF loudness, $g'_m = g'_0\ g_m$. For approach 1, $g'_0 = 1$. For approaches 2 and 3, g'_0 is calculated as

$$E_{\{g_m\}} = \sum_{m=0}^{M} C_m\, g_m^2 \tag{10}$$

$$g'_0 = \sqrt{N / E_{\{g_m\}}} \tag{11}$$

where C_m is the number of signals in the m^{th} order component. In 3-D, $C_m = m^2 + 1$; in 2-D, $C_1 = 1$ and $C_{m>1} = 2$. For approach 2, N is the total number of components: in 3-D, $(M+1)^2$; in 2-D, $2M + 1$. For approach 3, N is the number of loudspeakers in the array.

[7] For those without access to Mathematica, these can also be computed interactively using the Wolfram|Alpha online service, http://alpha.wolfram.com.