Development of Ambisonic Microphone Design Tools – Part 1

Authors : Charles J Middlicott and Bruce J Wiggins

University of Derby, UK

Correspondence should be addressed to Charles J Middlicott (c.middlicott1@derby.ac.uk)

ABSTRACT

In recent years an increase in the capture and production of ambisonic material has occurred as a result of companies such as YouTube and Facebook utilising ambisonics for spatial audio playback. There is now a greater need for affordable higher order microphone arrays. This work details the development of a set of tools which can be used to simulate and evaluate such microphone arrays, The 'Ambisonic Array Design Tool' for simulation and 'Ambisonic Array Evaluation Tool' for evaluation. The microphone capsules' position and directivity can be changed, with the effects on the synthesised spherical harmonics frequency and polar responses observed within the GUI. These scripts written in MatLab have been packaged within a GUI and will be available online.

1 Introduction

The capture of ambisonic signals was developed, to 1st order, by Gerzon and Craven in 1977 [1], where the notion of using a tetrahedral array of capsules was described. More recently, with the advent of spatial audio for virtual reality and 360° video, software tools have enabled users to synthesize higher order ambisonic material in Digital Audio Workstations (DAWs) in order to create immersive, full-sphere, sound fields. Yet the availability of microphones capable of capturing such signals has not grown at the same pace, or affordability.

A number of tools are available to design and simulate spaced microphone arrays such as MMAD [2] [3] and MARRS [4] but few tools are currently available for the design of coincident Ambisonic microphones.

This e-brief presents the development of a set of tools that can be used to simulate and evaluate the design of higher order microphone arrays, giving useful insight and performance metrics to use as a benchmark, prior to developing physical prototypes.

2 Ambisonics

2.1 Background

Ambisonics is a full-sphere isotropic surround format based around the decomposition of a 3D sound field into Spherical Harmonics (SH). These are an orthonormal set of basis functions on a sphere.

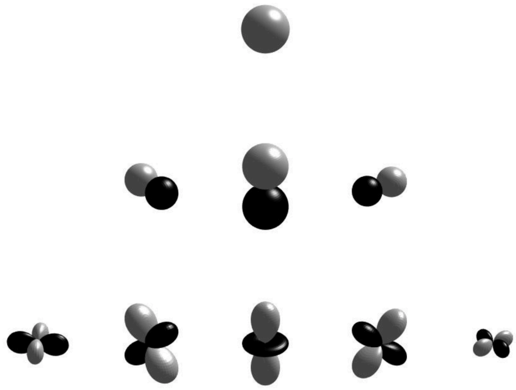

Figure 1 - Spherical Harmonics up to Order N = 2

The SH shown in Figure 1. are defined by the following equation [5].

$$Y_n^m(\theta,\phi) \equiv \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}}P_n^m(\cos\theta)\, e^{im\phi} \qquad (1)$$

Where

n is the order (0 to the max order N)
m is the degree (-n to n)
$(\cdot)!$ is the factorial function
P_n^m is the associated legendre polynomial

2.2 An Ideal Ambisonic Microphone

For an ambisonic array to exhibit an ideal response, the capsules must be coincident. In practice, it isn't possible for them to occupy the same physical space.

Additionally, a typical microphone may exhibit a flat on axis response frequency response – but for a capsule to be useful when developing an ambisonic array ideally it needs to exhibit an even response regardless of the angle of incidence. Capsules, even when matched, won't typically exhibit this desired response. This is especially true of capsules that have a low cost per unit, such as electrets.

2.3 Stages of Array Optimization

2.3.1 Pre-Filtering

Pre-filtering must occur by means of capsule calibration. To compensate for a less than ideal frequency/polar response. This optimisation also aims to correct for capsule mismatch. This has been implemented in various forms in the AAET.

2.3.2 Post-Filtering

Post filtering is the process of applying filters to the generated SH signals, typically to help compensate for the effects on the frequency response due to the non-coincident nature of the capsules and spatial aliasing

Filtering of the low frequency (LF) content varies depending on the order of the SH in question, this is especially useful as it limits excessive gain at LF which would otherwise increase the noise floor.
Spatial Aliasing occurs as a product of the sampling process, most typical schemes are alias free for order limited functions or exhibit negligible aliasing. However, in practice a sound field composed of plane waves isn't order limited. Therefore, due to the existence of higher order elements in a sound field (infinite number of spherical harmonics) spatial aliasing will occur [5].

The simplest way to overcome aliasing problems would be to increase the number of microphones, however this may not be possible. A more feasible approach would be to implement anti-aliasing filters or alias minimisation. This will be implemented in the applications at a later date.

3 Array Design / Evaluation Criteria

3.1 Current Functionality

Below a set of performance parameters have been defined, these are shown as plots that can be evaluated within the current incarnation of the *AADT* and *AAET* applications.

Individual Capsule Responses
- Time Domain Response
- Polar Response
- Frequency Response

Generated B-Format Responses
- Polar Response
- Frequency Response

These five responses were chosen as to aid the evaluation of an array at multiple stages in various domains.

4 Simulation Tool – Overview

The *Ambisonic Array Design Tool* (*AADT*) simulates the response of an array that has specific set of design attributes, chosen by the user, these are shown in Table 1.

These attributes are utilised to calculate a set of impulse responses (IRs) that can be used to evaluate an arrays theoretical performance characteristics listed in section 3.1.
As a tool, this gives the user a 'best case scenario', in regard to array performance, which subsequently developed prototype arrays can be evaluated against.

4.1 Assignable Parameters

To simulate an array using a graphical user interface (GUI) a set of user assignable parameters has to be defined, covering all the relevant array design features needed. These are used in various calculations detailed throughout this work. Table 1 details the parameters used to facilitate the generation of these responses.

List of User Assignable Parameters		
Ambisonic Order		*Order 'N'*
Array Radius		*(in millimetres)*
Array Type		2D or 3D *(3D to be added)*
Sampling Scheme		
a	Equal Angle	
b	Gaussian	*(To Be Added)*
c	T-Design	*(To Be Added)*
Capsule Directivity		*Directivity 'd'*
a	Omni	*d = 0*
b	Cardioid	*d = 1*
c	Figure of 8	*d = 2*
IR Filter Length		*(in Samples)*
Sampling Frequency		*(in kHz)*
Source Test Distance		*(in metres)*
Source Increment		*(in degrees '°')*

Table 1 – User Assignable Parameters in GUI

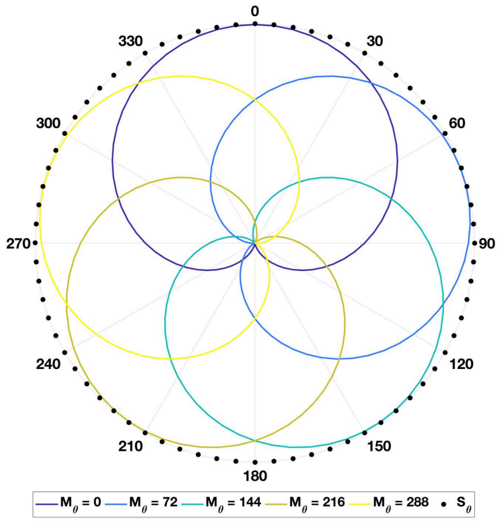

Figure 2 - Offset Directivity Patterns at M_θ positions

4.2 Generating Simulated Impulses

4.2.1 *Capsule Directivity and Offset*

Each capsule that makes up the array will have a stated 1st order polar pattern based on the directivity factor 'd' as shown below in equation 2. The direction of orientation of the microphone is set by the parameter '*M*'.

$$0.5 \cdot \big((2 - d) + d\big) \cdot \cos\left(S_\theta - M_\theta\right) \qquad (2)$$

Where

d is the microphone capsule directivity factor
S_θ is a given angular source position in radians
M_θ is a given angular capsule position in radians

This generates gains for *'M'* microphones at *'S'* source positions, shown in Figure 2.

4.2.2 *Simulating Distance Effects*

In order to simulate the distance effects, such as array radius, a fractional delay line is necessary. This can be implemented as a Sinc pulse that can be used to represent an impulse occurring in the free field at given time *'t'*.

To generate the required fractional delay line, a time delay needs to be calculated. To derive this, first we must generate both the microphone and source positions in cartesian coordinates, see Eq. 3 & 4.

$$M_x = Radius \cdot \cos(M_\theta)$$
$$M_y = Radius \cdot \sin(M_\theta) \qquad (3)$$
$$S_x = SrcDist \cdot \cos(S_\theta)$$
$$S_y = SrcDist \cdot \sin(S_\theta) \qquad (4)$$

Where the *SrcDist* is the distance between the centre of the array and the cartesian coordinates at each angular source location. The *Radius* is the radius of the array in metres.

These coordinates are then used in the numerator of Eq. 5 to derive the time delay by generating the distances from the source minus the microphone location. The resulting distance is then divided by *'c'* thus generating a time delay in seconds, with *'c'* being the speed of sound at *343m/s.*

$$\frac{\sqrt{(S_x - M_x)^2 + (S_y - M_y)^2}}{c} \qquad (5)$$

A nominal acoustic delay is calculated by multiplying the sampling frequency by the source distance (the distance from a given source position to the centre of the array) in metres, this is divided by 'c'. This is used so the simulated impulses can be centred relative to the specified filter length.

As previously mentioned once the time delay is calculated the Sinc pulse can then be utilised to implement the fractional delay line. A Sinc is defined in the following equation using the sine function.

$$sinc(x) \equiv \begin{cases} 1 \\ \dfrac{\sin(x)}{x} \end{cases} \qquad (6)$$

which has the following normalisation [6].

$$\int_{-\infty}^{\infty} sinc(x)dx = \pi \qquad (7)$$

A set of dynamically generated Sinc pulses are used (minus the calculated time delay for source to microphone) to give the fractional delay (inter-sample) needed for that source location. A hamming window is then applied to fix ripple issues.

Using Eq. 2 and applying the aforementioned Sinc pulses results in the packed matrix *'C'* containing simulated impulses per capsule / source direction.

4.2.3 Generating B-Format Signals

To generate the desired B-Format signals a set of SH coefficients must be calculated and applied to each microphone capsule. The coefficients for each SH are generated using the following equation modified from [7].

$$Y_n^m(\theta,\phi) \equiv N_n^{|m|} P_n^{|m|}(\sin(M_\phi)) \begin{Bmatrix} \cos(|m|M_\theta) & \text{if} \geq 0 \\ \sin(|m|M_\theta) & \text{if} < 0 \end{Bmatrix} \qquad (8)$$

Where

M_θ is a given angular capsule azimuth in radians
M_Φ is a given angular capsule elevation in radians
n is the order (0 to the max order N)
m is the degree ($-n$ to n)
P_n^m is the associated Legendre polynomial
N_n^m is a gain value for a given normalisation scheme

Once calculated for a given SH they must be convolved with each capsules simulated IRs. These are then summed together to create the desired SH signal.

5 Evaluation Tool - Overview

The *Ambisonic Array Evaluation Tool (AAET)* differs to its counterpart in that its purpose is to evaluate/validate a physical arrays performance, be it a prototype or a commercially available product, against the performance of a comparable simulation.

This application takes in the same user assignable parameters as the AADT and generates its simulated impulses in the same manner shown above. It needs to be supplied with a set of measured IRs, The GUI takes in a .MAT file containing the array parameters and measured IRs. These are then both plotted in the same manner as the AADT but with the simulated / measured responses being displayed side by side for direct comparison.

The main difference in terms of functionality is the ability to generate calibration filters from the measured responses utilising various approaches. This pre-filtering of the capsules can be evaluated in the same manner as the raw responses. This is useful as you can instantly view any change in response. Implementation and evaluation of the post-filtering stage will be considering in Part 2 of this work.

5.1 Prototype Circular Array

While creating the AAET application this work necessitated the development of a physical prototype. In this instance, it was decided to develop a 2nd order circular array. The decision was taken to focus solely on the azimuth plane. This was so factors such as spacing, calibration and filtering could be evaluated with greater ease and findings could be used to inform the development of a 3D spherical array. Secondary to this, with the current measurement apparatus only offering the ability to measure responses in the azimuth plane, it was decided a circular array was appropriate as an initial prototype.

The minimum number of capsules needed to capture the horizontal only 2nd order SH signals using an equal-angle sampling distribution is $2(N+1)$, where N is the order [5]. This led to a circular array prototype that consists of five 16.5mm electret capsules with a capsule spacing of 72° and an array radius of 20mm. This radius was calculated considering the physical size of the TSB165A-T capsules utilised [8]. The initial prototype was used to validate visually against the simulation data generated in the AAET.

5.1.1 Objective Measurement

Objective measurement of the prototype array took place in a hemi-anechoic chamber, a total set of 360 measured IRs were captured. Measurements were taken on the horizontal plane in 5° increments, using an automated turntable, at a 2-metre distance from the acoustic centre of the prototype array.

The measurement source material was a 15 second 20Hz to 20kHz logarithmic sine sweep, this was fed to a KRK Systems ROKIT RP8 loudspeaker. An omnidirectional measurement microphone [9] was used to capture the response of the loudspeaker used. Nelson/Kirkeby inversion was utilised to generate an inverse of this speaker response using Eq. 9 [10]. The resultant filter is convolved with the captured IRs thus compensating for the frequency response of the loudspeaker.

$$H_{inv}(\omega) = \frac{H_{target} \cdot Conj(H(\omega))}{Conj(H(\omega)) \cdot H(\omega) + e(\omega)} \qquad (9)$$

Where H is the measured response at the frequency index ω, H_{target} is the desired target response, ε *is* a regularisation parameter.

6 UI Design

The GUI was designed using Graphical User Interface Design Environment (GUIDE) a drag-and-drop environment for laying out user interfaces (UIs). The interactive behavior of the applications is then coded separately in MATLAB [11]. See Figure 3 for a view of the AADT layout

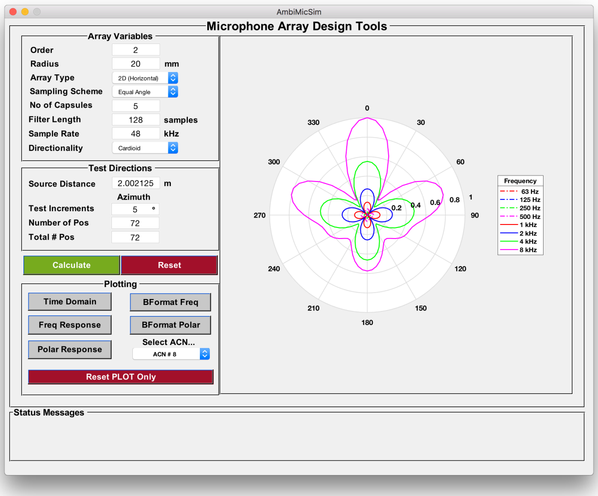

Figure 3 – AADT GUI Layout

7 Future Work

A set of additional features are to be implemented, such as full 3D simulation capability, additional array construction type (rigid/open, dual sphere), frequency dependant capsule directivity, propagation loss and additional sampling schemes. Further to this the AAET will be updated to consider both the generation and evaluation of pre/post filters in future revisions.

8 Conclusions

This e-brief has described the development of a set of ambisonic array design and evaluation tools that can be used to rapidly simulate a range of array designs. Array performance data can be generated to evaluate an array at various stages such as the pre/post processing. This can be used to aid the development of future array prototypes.

References

[1] M. A. Gerzon, P. Craven, "*Coincident Microphone Simulation Covering Three-Dimensional Space and Yielding Various Directional Outputs*," *US4042779A* (1977).

[2] M. Williams, G. Le Du, "*The Quick Reference Guide to Multichannel Microphone Arrays Design Part I : using Cardioid Microphones*". *In Audio Engineering Society Convention 110,* 5335, (2001)

[3] M. Williams, G. Le Du, "*The Quick Reference Guide to Multichannel Microphone Arrays Design Part II : using Supercardioid and Hypocardioid Microphones*". *In Audio Engineering Society Convention 116,* 6059, (2004)

[4] H, Lee. D. Johnson, M. Mironovs, "*An Interactive and Intelligent Tool for Microphone Array Design*". *In Audio Engineering Society Convention 143,* e-Brief 390, (2017)

[5] B, Rafaely. *"Fundamentals of Spherical Array Processing" (Springer Topics in Signal Processing)* Springer (2015).

[6] Weisstein, E W. "*Sinc Function*." From MathWorld--A Wolfram Web Resource. *http://mathworld.wolfram.com/SincFunction.html*

[7] J, Daniel. "*Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimedia"* PhD Thesis *(2000)*

[8] JLI Electronics LLC. "*TSB165A-T Capsule*." *http://www.jlielectronics.com/content/JLI-165A-T.pdf*

[9] Earthworks Inc. "*M30BX Omni Microphone*." *https://earthworksaudio.com/wp-content/uploads/2018/07/M30BX-Data-Sheet-2018.pdf*

[10] *H. Tokuno, O. Kirkeby, P. Nelson, "Inverse filter of sound reproduction systems using regularization" IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. 80, No. 5. 809-820. (1997).*

[11] Mathworks Inc, *"MATLAB® App Building - R2018a"* https://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf