



DATASHEET

Eigenbeam Data

Specification for Eigenbeams

Version 1

Rev. A

mh acoustics, LLC
25A Summit Ave
Summit, NJ 07901
USA

Phone: +1 908-277-3131

FAX: +1 908-934-9233

www.mhacoustics.com

Copyright © 2016 mh acoustics, LLC. All rights reserved.

Eigenmike, EigenStudio, and EigenUnits are trademarks of mh acoustics, LLC.

Table of Contents

1. Introduction	4
2. Definitions and Conventions	5
2.1 Eigenbeams (Spherical Harmonics, Ambisonic signals)	5
2.2 Spherical Coordinate System	6
2.3 Ambisonics Conventions	7
2.3.1 Channel Ordering.....	8
2.3.2 Normalization	9
2.3.3 Polarization	9
3. Eigenbeam Specifications	11
3.1 Overview	11
3.2 Individual Eigenbeams	13

1. Introduction

The em32 Eigenmike[®] microphone array is capable of capturing a soundfield with a high degree of spatial accuracy. The 32 raw microphone signals can be encoded into Eigenbeams (a.k.a. spherical harmonics, Ambisonic signals) of up to 4th order using the stand-alone EigenStudio[®] application or the EigenUnits[®] plugins (EigenUnit-Encoder).¹ These signals are suitable for use with various decoders, beamformers, etc. either from mh acoustics (beamforming with EigenStudio or EigenUnits) or using third-party tools. See the "EigenStudio User Manual"² and "EigenUnits User Manual"³ documents for more details.

The following document details the characteristics of the Eigenbeams produced by the encoding process using mh acoustics' software. There are many variations on how these signals are defined and represented across various implementations in the Ambisonics community (especially regarding channel ordering, normalization, and polarization). This document aims to alleviate any confusion that may arise when working with the Eigenbeams from the em32 Eigenmike array by presenting definitions, conventions, and performance specifications. Furthermore, practical real-world limitations when encoding to higher-order Eigenbeams (or equivalently Ambisonics) are discussed and should be considered when designing or working with Ambisonic decoders.

¹ <http://mhacoustics.com/download>

² [http://mhacoustics.com/sites/default/files/EigenStudio User Manual R02B.pdf](http://mhacoustics.com/sites/default/files/EigenStudio%20User%20Manual%20R02B.pdf)

³ [http://mhacoustics.com/sites/default/files/EigenUnits User Manual R01D.pdf](http://mhacoustics.com/sites/default/files/EigenUnits%20User%20Manual%20R01D.pdf)

2. Definitions and Conventions

2.1 Eigenbeams (Spherical Harmonics, Ambisonic signals)

The Eigenbeam signals are characterized by their beampatterns which are represented by the spherical harmonic functions. Different conventions exist for the spherical harmonic functions regarding scaling, polarity and real or complex valued. In the beamforming community the following complex orthonormal definition is commonly used:

$$Y_n^m(\vartheta, \varphi) \equiv \sqrt{\frac{(2n+1)(n-m)!}{4\pi(n+m)!}} P_n^m(\cos\vartheta) e^{im\varphi} \quad (1.1)$$

On the other hand, the Ambisonics community typically uses the following real-valued spherical harmonic functions:

$$Y_{n,m}(\vartheta, \varphi) \equiv \aleph_{n,|m|} P_{n,|m|}(\cos\vartheta) \begin{cases} \sqrt{2} \sin(|m|\varphi) & \text{if } m < 0, \\ 1 & \text{if } m = 0, \\ \sqrt{2} \cos(|m|\varphi) & \text{if } m > 0. \end{cases} \quad (1.2)$$

where P again represents the Legendre function. Following the suggested notation by Abramovitz in the standard book "Handbook of Mathematical Functions" we distinguish the notation P_n^m and $P_{n,m}$ to indicate the use of the Condon-Shortley phase. P_n^m includes the Condon-Shortley phase while $P_{n,m}$ does not. $\aleph_{n,m}$ is a normalization factor. To achieve the orthonormal property for the real valued spherical harmonic function this norm factor becomes:

$$\aleph_{n,|m|} = \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}} \quad (1.3)$$

In the context of Ambisonics the normalization from Eq. 1.3 is commonly referred to as N3D normalization.

The real-valued spherical harmonic functions can be expressed in terms of the complex valued version as follows:

$$Y_{n,m} = \begin{cases} \sqrt{2}(-1)^m \Im(Y_n^m) & \text{if } m < 0, \\ Y_n^0 & \text{if } m = 0, \\ \sqrt{2}(-1)^m \Re(Y_n^m) & \text{if } m > 0. \end{cases} \quad (1.4)$$

We will refer to the subscript n as the order and m as the degree of the spherical harmonic. Note that the naming of n and m is reversed relative to standard naming conventions of the associated Legendre Functions P in the field of mathematics for these variables⁴. For the remainder of this document we will use the *real*-valued definition of the spherical harmonic functions. However, some tables and plots may use the Y_n^m symbol to improve readability.

2.2 Spherical Coordinate System

For the Eigenmike array, the following conventions are used when defining the spherical coordinate system (see Figures 1 and 2):

- **Vertical Angle** (ϑ): is the angle in the vertical dimension. In degrees it ranges from 0 to 180. The 0 degrees direction points away from the top of the spherical array (the opposite side from where the shaft mounts to the Eigenmike array; towards the ceiling in a typical arrangement). The 90 degrees direction is the horizontal plane, and the 180 degrees direction is in the direction of the shaft (typically towards the floor).
- **Horizontal Angle** (φ or ϕ): is the angle in the horizontal plane. It ranges from 0 to 360 degrees. The 0 degrees direction aligns with the "mh acoustics" logo on the shaft of the Eigenmike array. The angle increases in the counter-clockwise direction looking from the top of the Eigenmike array.

Note that some third-party software in the Ambisonics community may use different conventions⁵. Care should be taken to interpret angular conventions appropriately.

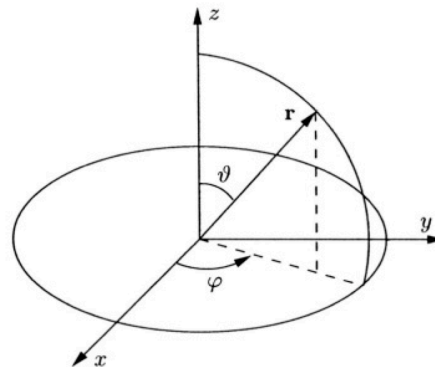


Figure 1: Spherical Coordinate Conventions

⁴ The reason we have chosen to flip this naming terminology has to do with differential beampatterns being typically defined by their "order". For the spherical harmonic representation the differential "order" is related to the subscript n , so we have termed this variable as "order" and m as the degree.

⁵ Often the vertical angle is defined from -90° (floor) to $+90^\circ$ (ceiling) with 0° degrees being the horizontal plane. Using this definition for the vertical angle requires changing the argument of the Legendre function in Eqs. 1.1 and 1.2 from \cos to \sin . With this modification both definitions yield equivalent results.



Figure 2: Coordinate Axes w.r.t. em32

2.3 Ambisonics Conventions

Many different conventions exist in the Ambisonics community for Ambisonic signal channel ordering, normalization, and polarization. The following sections describe in detail how each of these conventions relate to the Eigenbeams as implemented in the Eigenmike software.

Also, in some instances, certain terms may refer to a combination of channel ordering and normalization schemes. In particular, the following Ambisonic terms are often used to refer to the a combination of conventions:

- **ambiX⁶**: ACN channel ordering; SN3D normalization
- **FuMa**: FuMa channel ordering; FuMa normalization
- **MPEG-H**: a subset of the MPEG-H standard supports Ambisonics using ACN channel ordering and N3D normalization

An excellent (but still under-construction) resource for learning more about Ambisonic channel ordering and normalization schemes can be found in the Wikipedia page on "Ambisonic data exchange formats"⁷, as well as in the references cited by that article.

⁶http://iem.kug.ac.at/fileadmin/media/iem/projects/2011/ambisonics11_nachbar_zotter_sontacchi_deleflie.pdf

⁷ https://en.wikipedia.org/wiki/Ambisonic_data_exchange_formats

2.3.1 Channel Ordering

Two different channel orderings for Ambisonic signals are commonly found in use by Ambisonic software tools: *ACN* and *FuMa*. This document does not present the Eigenbeams in a specific ordering, but instead refers to them using the unambiguous Y_n^m notation (see section 2.1). In practice, both EigenStudio and the EigenUnits plugins are capable of outputting the Eigenbeams in either channel ordering (as a selectable parameter). The relationship between the channel number and Eigenbeam for each ordering is shown below.

Channel Number	ACN Ordering		FuMa Ordering	
	Eigenbeam	ACN No.	Eigenbeam	FuMa Name
1	Y_0^0	ACN 0	Y_0^0	W
2	Y_1^{-1}	ACN 1	Y_1^1	X
3	Y_1^0	ACN 2	Y_1^{-1}	Y
4	Y_1^1	ACN 3	Y_1^0	Z
5	Y_2^{-2}	ACN 4	Y_2^0	R
6	Y_2^{-1}	ACN 5	Y_2^1	S
7	Y_2^0	ACN 6	Y_2^{-1}	T
8	Y_2^1	ACN 7	Y_2^2	U
9	Y_2^2	ACN 8	Y_2^{-2}	V
10	Y_3^{-3}	ACN 9	Y_3^0	K
11	Y_3^{-2}	ACN 10	Y_3^1	L
12	Y_3^{-1}	ACN 11	Y_3^{-1}	M
13	Y_3^0	ACN 12	Y_3^2	N
14	Y_3^1	ACN 13	Y_3^{-2}	O
15	Y_3^2	ACN 14	Y_3^3	P
16	Y_3^3	ACN 15	Y_3^{-3}	Q
17	Y_4^{-4}	ACN 16	Y_4^0	B*
18	Y_4^{-3}	ACN 17	Y_4^1	C*
19	Y_4^{-2}	ACN 18	Y_4^{-1}	D*
20	Y_4^{-1}	ACN 19	Y_4^2	E*
21	Y_4^0	ACN 20	Y_4^{-2}	F*
22	Y_4^1	ACN 21	Y_4^3	G*
23	Y_4^2	ACN 22	Y_4^{-3}	H*
24	Y_4^3	ACN 23	Y_4^4	I*
25	Y_4^4	ACN 24	Y_4^{-4}	J*

* FuMa is not defined beyond 3rd-order. However, the extended FuMa channel ordering shown above is given as a reference for the implementation in the Eigenmike software.

2.3.2 Normalization

Multiple normalization schemes for Ambisonic signals are also found in use by Ambisonic software tools. These include, but are not limited to: *SN3D*, *N3D*, *FuMa*, and *maxN*. In practice, both EigenStudio and the EigenUnits plugins are capable of outputting the Eigenbeams with various normalization schemes (as a selectable parameter).

As described earlier the *N3D* normalized Eigenbeam is obtained by combining Eqs. 1.1 and 1.2. The table below lists the (approximate) scale factors to be applied for converting from *N3D* to other normalization schemes.

Eigenbeam	to SN3D	to FuMa	to maxN
Y_0^0	0.0 dB	-3.0 dB	0.0 dB
Y_1^0	-4.8 dB	-4.8 dB	
$Y_1^{\pm 1}$	-4.8 dB	-4.8 dB	
Y_2^0	-7.0 dB	-7.0 dB	
$Y_2^{\pm 1}$	-7.0 dB	-5.7 dB	
$Y_2^{\pm 2}$	-7.0 dB	-5.7 dB	
Y_3^0	-8.5 dB	-8.5 dB	
$Y_3^{\pm 1}$	-8.5 dB	-7.0 dB	
$Y_3^{\pm 2}$	-8.5 dB	-5.9 dB	
$Y_3^{\pm 3}$	-8.5 dB	-6.4 dB	
Y_4^0	-9.5 dB	-9.5 dB	
$Y_4^{\pm 1}$	-9.5 dB	-8.0 dB	
$Y_4^{\pm 2}$	-9.5 dB	-6.7 dB	
$Y_4^{\pm 3}$	-9.5 dB	-6.2 dB	
$Y_4^{\pm 4}$	-9.5 dB	-6.9 dB	

2.3.3 Polarization

The polarization of the Eigenbeams results from Eq. 1.3. As mentioned previously, Condon-Shortly phase is not used. All Eigenbeams are in-phase at their angles of maximum sensitivity listed below. See section 2.2 for coordinate conventions; angles are in radians.

Eigenbeam	Vertical (ϑ)	Horizontal (φ)
Y_0^0	0	0
Y_1^{-1}	$\pi/2$	$\pi/2$
Y_1^0	0	0
Y_1^1	$\pi/2$	0

Y_2^{-2}	$\pi/2$	$\pi/4$
Y_2^{-1}	$\pi/4$	$\pi/2$
Y_2^0	0	0
Y_2^1	$\pi/4$	0
Y_2^2	$\pi/2$	0
Y_3^{-3}	$\pi/2$	$\pi/6$
Y_3^{-2}	$\tan^{-1}(\sqrt{2})$	$\pi/4$
Y_3^{-1}	$\cos^{-1}\left(\sqrt{\frac{11}{15}}\right)$	$\pi/2$
Y_3^0	0	0
Y_3^1	$\cos^{-1}\left(\sqrt{\frac{11}{15}}\right)$	0
Y_3^2	$\tan^{-1}(\sqrt{2})$	0
Y_3^3	$\pi/2$	0
Y_4^{-4}	$\pi/2$	$\pi/8$
Y_4^{-3}	$\cos^{-1}\left(\sqrt{\frac{1}{2}}\right)$	$\pi/6$
Y_4^{-2}	$\cos^{-1}\left(2\sqrt{\frac{1}{7}}\right)$	$\pi/4$
Y_4^{-1}	$\cos^{-1}\left(\sqrt{\frac{27+\sqrt{393}}{56}}\right)$	$\pi/2$
Y_4^0	0	0
Y_4^1	$\cos^{-1}\left(\sqrt{\frac{27+\sqrt{393}}{56}}\right)$	0
Y_4^2	$\cos^{-1}\left(2\sqrt{\frac{1}{7}}\right)$	0
Y_4^3	$\cos^{-1}\left(\sqrt{\frac{1}{2}}\right)$	0
Y_4^4	$\pi/2$	0

3. Eigenbeam Specifications

The following sections detail performance specifications for the Eigenbeams, as implemented in the latest version of the Eigenmike software⁸.

The data presented here represents the cumulative output of the signal chain comprising the em32 microphone array and associated signal processing software (i.e. encoding of the sensor signals by either EigenStudio or EigenUnits plugins). The data has been derived from simulations using the actual software implementation, and has been proven to agree well with real-world measurement results.

3.1 Overview

A view of the complete set of the Eigenbeams, up to fourth order, are depicted below in Figure 3. The two different colors represent phase relative to Y_0^0 . For example, in the dipole Y_1^1 , the signal generated from sounds incident from the direction of the positive x-axis are in phase with Y_0^0 , while the signal generated from sounds incident from the direction of the negative x-axis are in 180° out of phase with Y_0^0 (see Section 2.3.3 for more details).

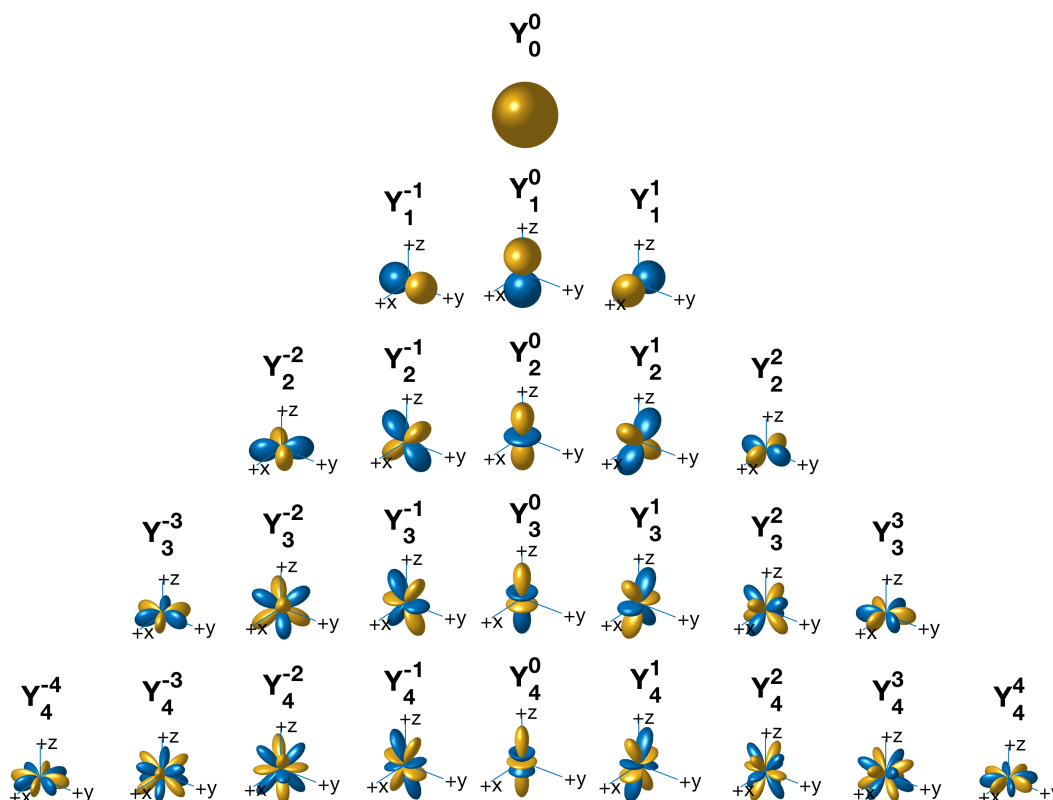


Figure 3: Eigenbeams up to 4th order

⁸ EigenStudio v2.6.2 and EigenUnits v0.8, October 2016

All Eigenbeams are computed such that they have a nominally flat magnitude frequency response, up to the spatial Nyquist frequency. It is important to note that Eigenbeams of increasingly higher order are restricted to increasingly limited operating frequency ranges, in order to limit the maximum amount of system self-noise generated during the encoding process. The table below lists the lowest operating frequency per order.

Eigenbeam Order, n	Cutoff Frequency (approx.)
0, 1	30 Hz
2	400 Hz
3	1000 Hz
4	1800 Hz

A portion of the spectrum for the $m=0$ degree Eigenbeams is shown below for illustration. Here, the Eigenbeams are shown with $maxN$ normalization applied for easy interpretation in the plot. It should be noted that the encoded signals extend all the way up to the temporal Nyquist frequency (i.e. $44.1\text{kHz}/2$ or $48\text{kHz}/2$). However, all Eigenbeams will exhibit more complex spectral and spatial behavior above the spatial Nyquist limit⁹.

In practice, mh's EigenStudio application and EigenUnits plugins automatically incorporate this operating range into the beamforming/decoding process. Special care should be taken when designing or working with third-party decoders.

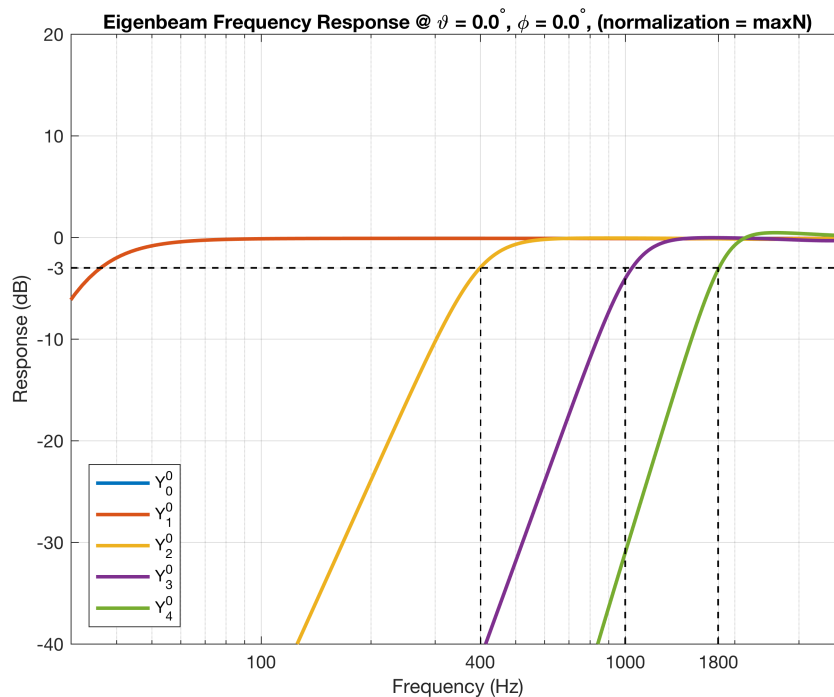


Figure 4: Eigenbeam cutoff frequencies

⁹ Approximately 8kHz for the 32-element 8.4cm diameter em32 Eigenmike

3.2 Individual Eigenbeams

This section details the performance of 25 individual Eigenbeams up to fourth order. Details regarding their spatial response, relative phase (polarity), and noise performance are given.

For compactness, the non-zero degree Eigenbeams are shown as pairs of +/- m . In these cases, the two-dimensional polar plots shown depict only the $m>0$ response. The corresponding polar plot for the $m<0$ Eigenbeam can be inferred through symmetry and a simple rotation. Also, polar plots for the vertical plane refer to the X-Z plane. Per standard practice, all polar plots are normalized to the maximum sensitivity in the displayed plane.

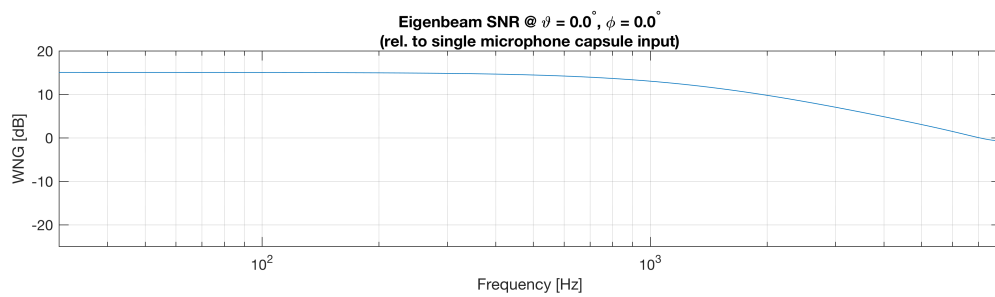
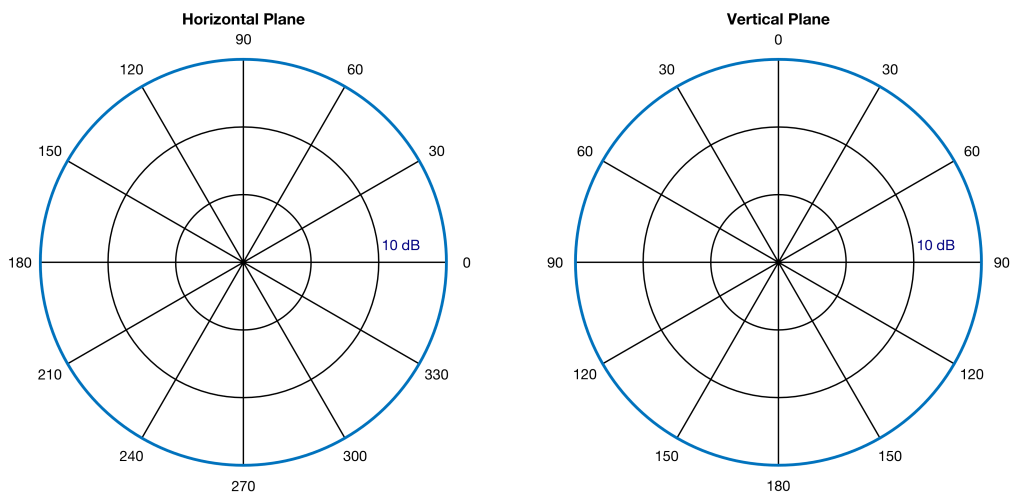
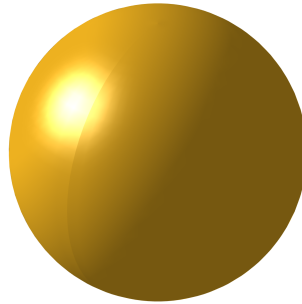
The Eigenbeam SNR plots represent the *change* in SNR for the encoding of a specific Eigenbeam *relative to the SNR of a single microphone capsule*. This is referred to as the White-Noise-Gain (WNG). The absolute SNR of the Eigenbeam can be calculated by adding (in dB) the Eigenbeam's WNG value to the individual microphone capsule SNR as specified by the hardware datasheet¹⁰¹¹. Note that these values are specific to the em32 Eigenmike hardware and software.

Also, for these plots, data is depicted only up to spatial Nyquist. However, it should be noted that the Eigenbeam software encoding process does generate signals extending all the way up to temporal Nyquist (i.e. 44.1kHz/2 or 48kHz/2).

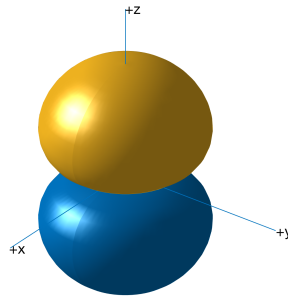
¹⁰ Refer to the "em32_datasheet" included with the em32 hardware for more details.

¹¹ For example, the encoding process for Y_1^1 yields a WNG of approximately +11dB at 1kHz. Using individual microphone capsules with an SNR of 79dBA (15dBA ENL) would then yield an absolute SNR for Y_1^1 of 79+11= 90dB at 1kHz.

Eigenbeam Y_0^0



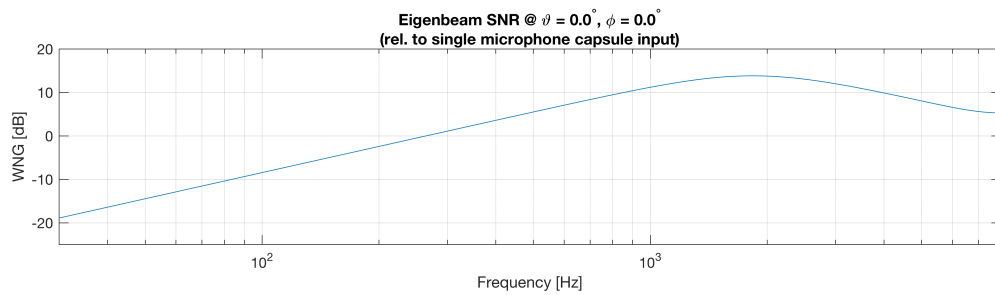
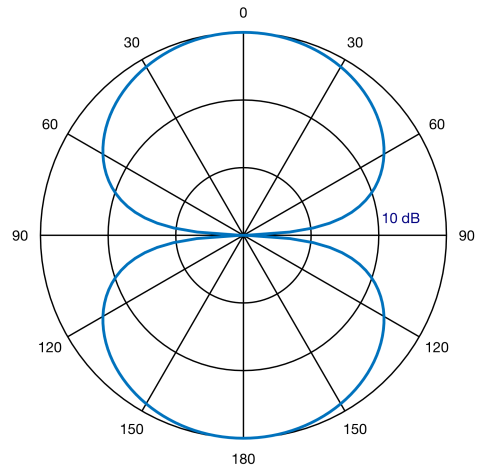
Eigenbeam Y_1^0



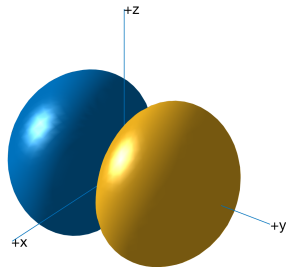
Horizontal Plane

N/A

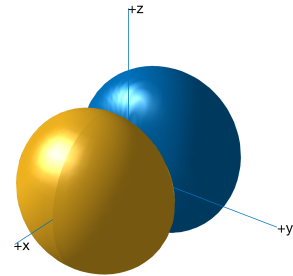
Vertical Plane



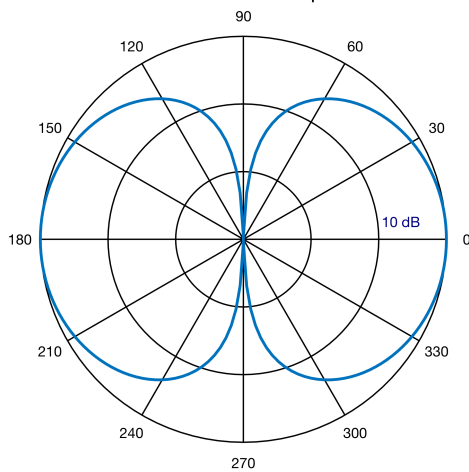
Eigenbeam Y_1^{-1}



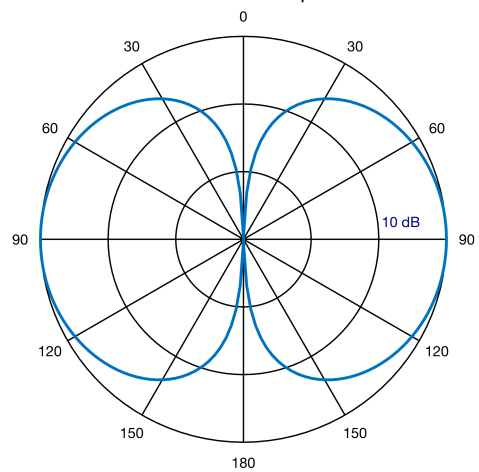
Eigenbeam Y_1^1



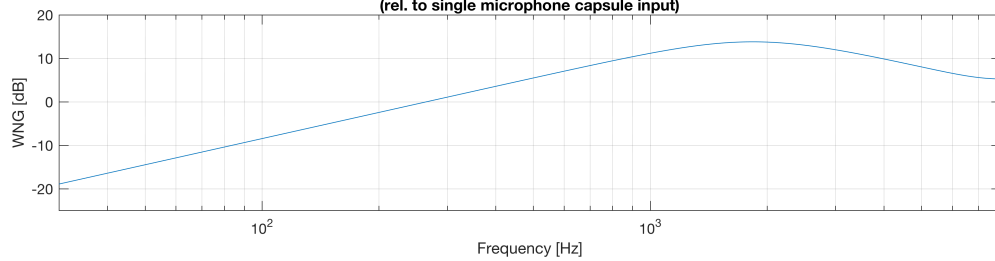
Horizontal Plane (Y_1^1)



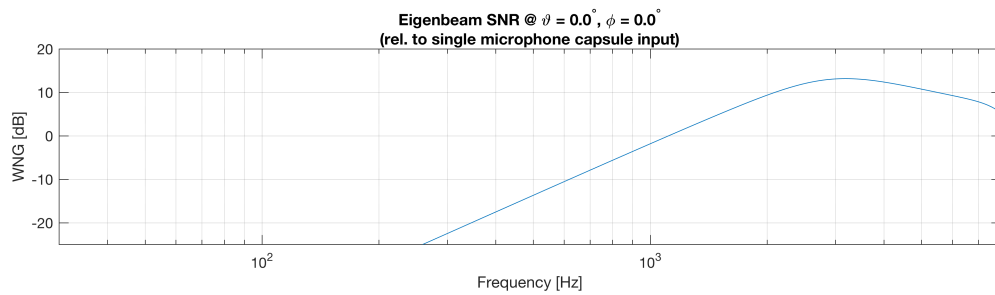
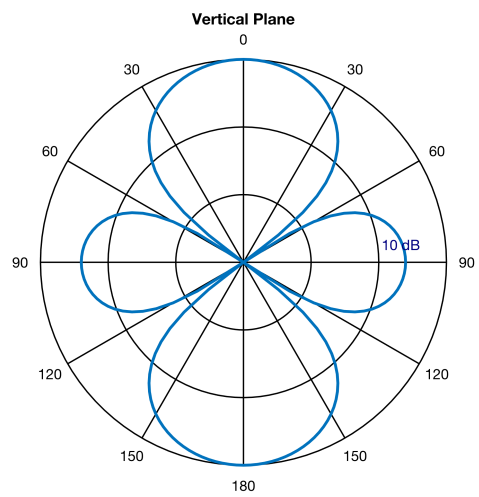
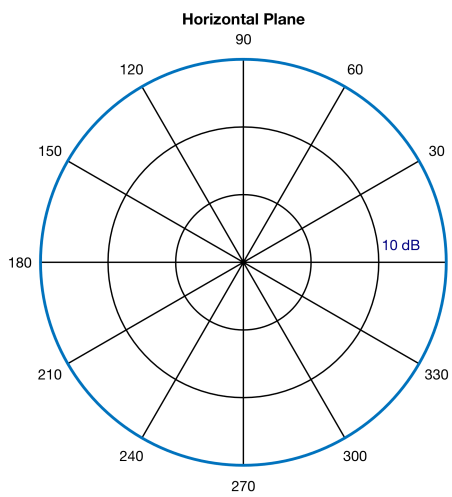
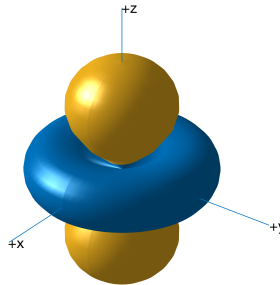
Vertical Plane (Y_1^1)



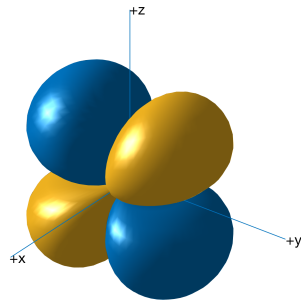
Eigenbeam SNR @ $\theta = 90.0^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)



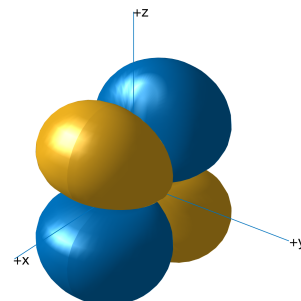
Eigenbeam Y_2^0



Eigenbeam Y_2^{-1}



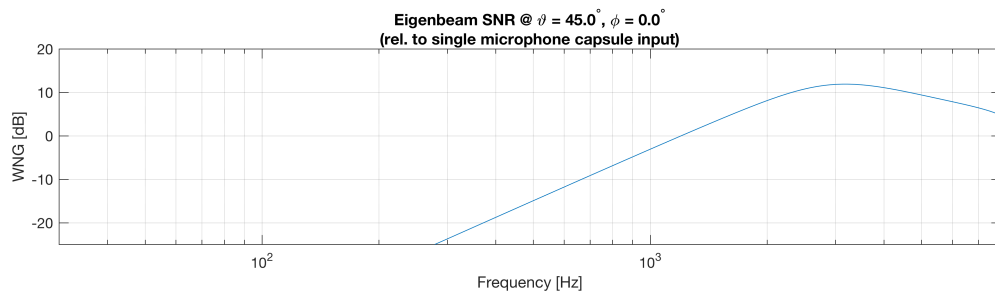
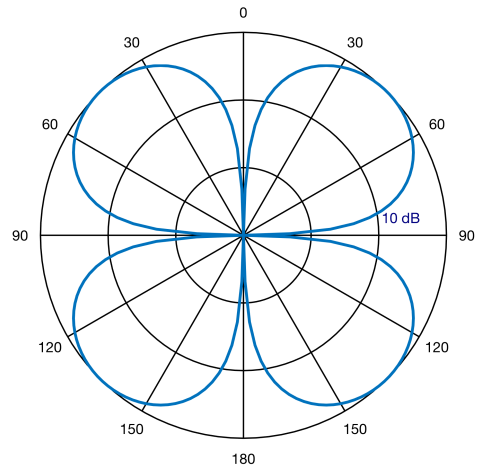
Eigenbeam Y_2^1



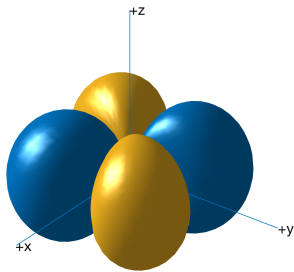
Horizontal Plane (Y_2^1)

N/A

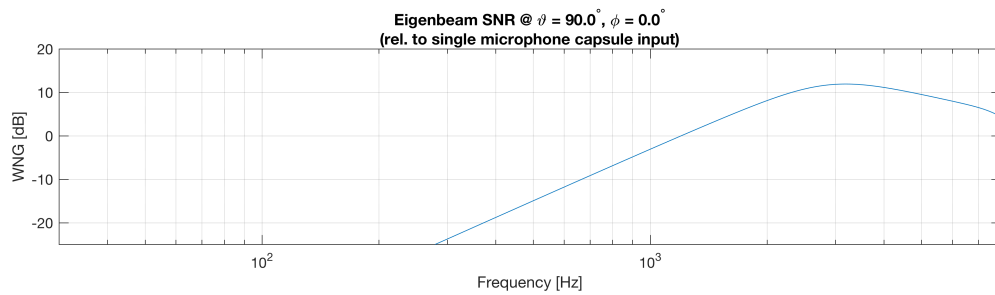
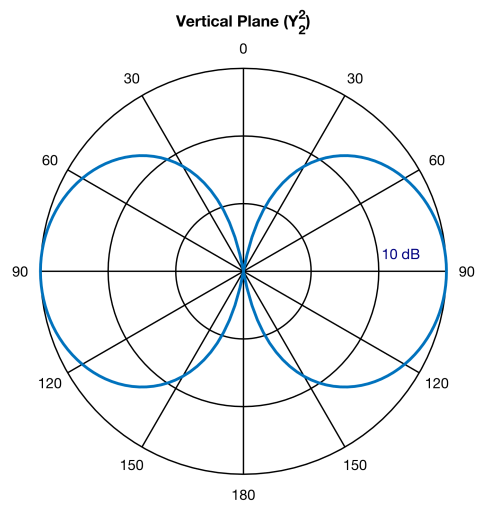
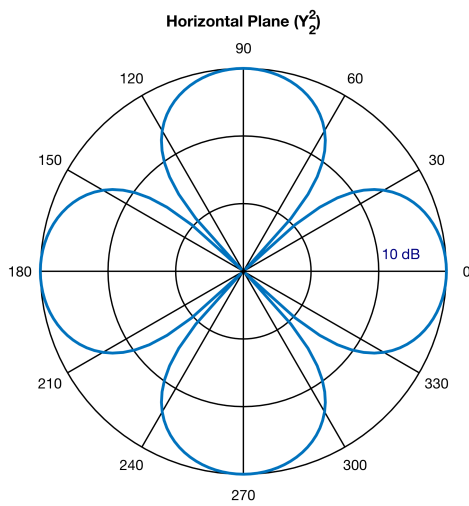
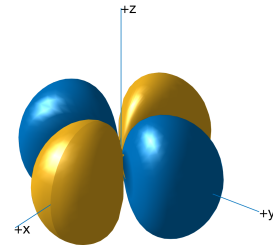
Vertical Plane (Y_2^1)



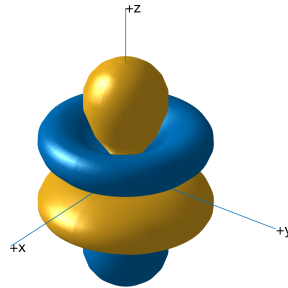
Eigenbeam Y_2^{-2}



Eigenbeam Y_2^2



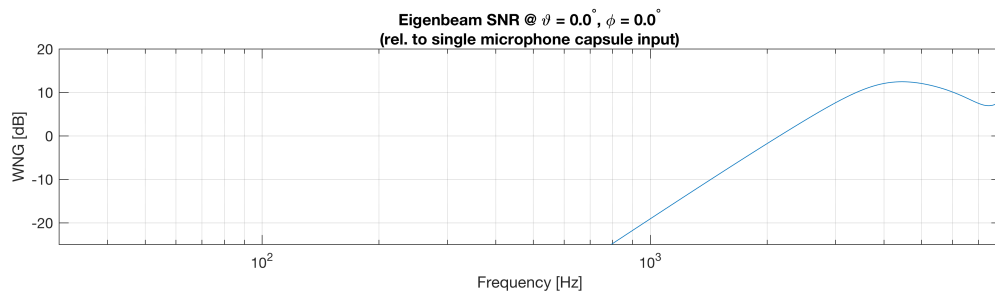
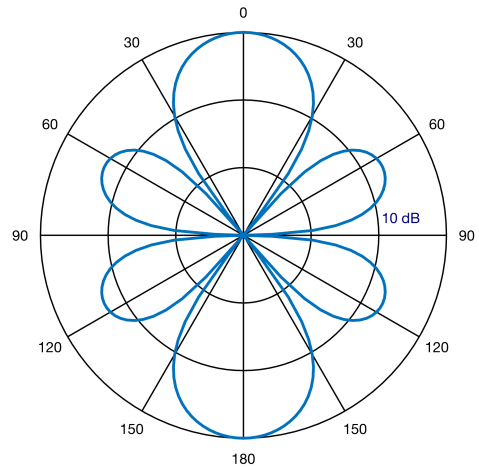
Eigenbeam Y_3^0



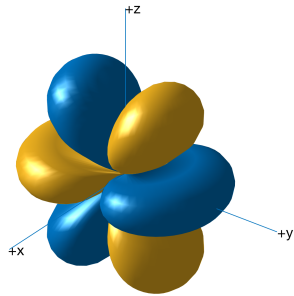
Horizontal Plane

N/A

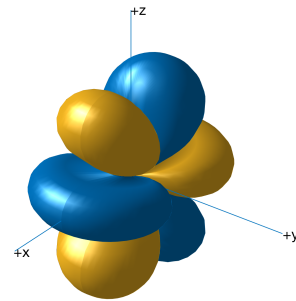
Vertical Plane



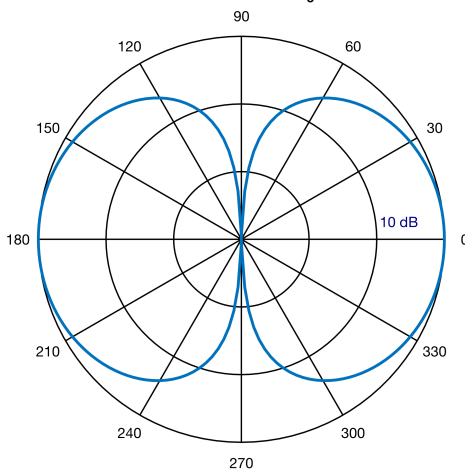
Eigenbeam Y_3^{-1}



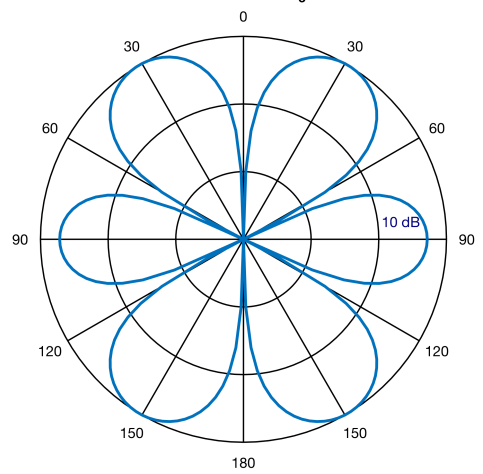
Eigenbeam Y_3^1



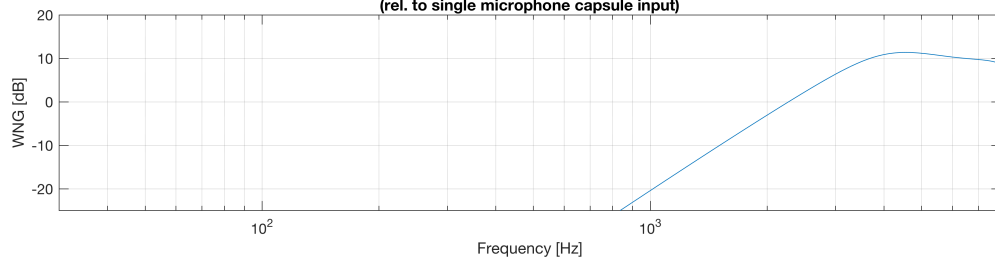
Horizontal Plane (Y_3^1)



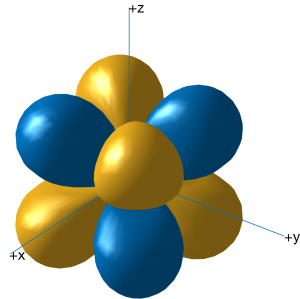
Vertical Plane (Y_3^1)



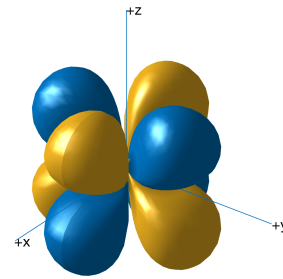
Eigenbeam SNR @ $\theta = 31.1^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)



Eigenbeam Y_3^{-2}



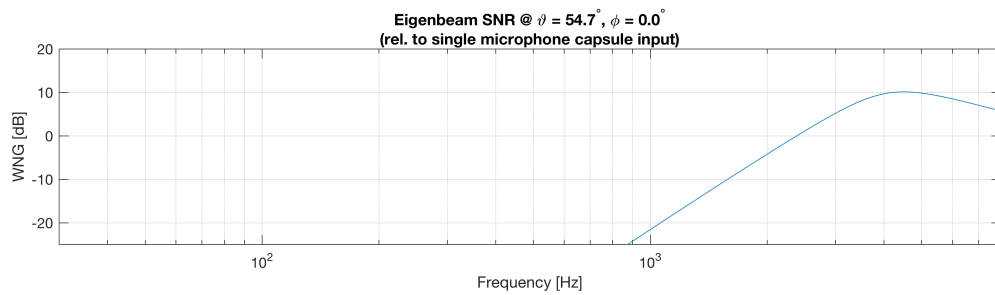
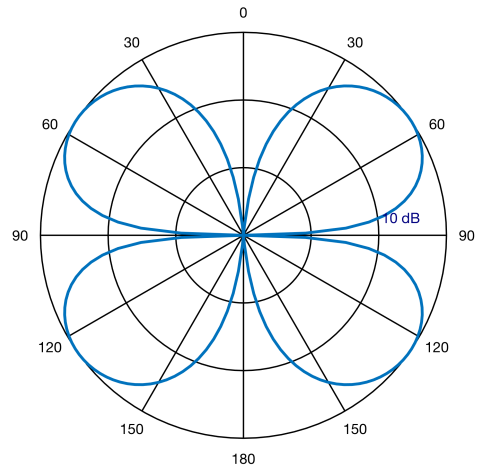
Eigenbeam Y_3^2



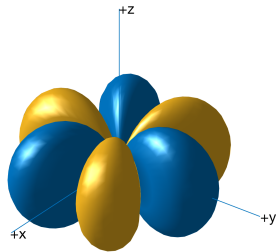
Horizontal Plane (Y_3^2)

N/A

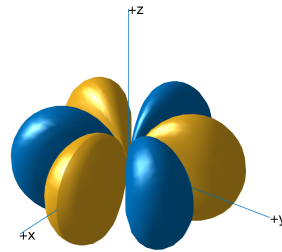
Vertical Plane (Y_3^2)



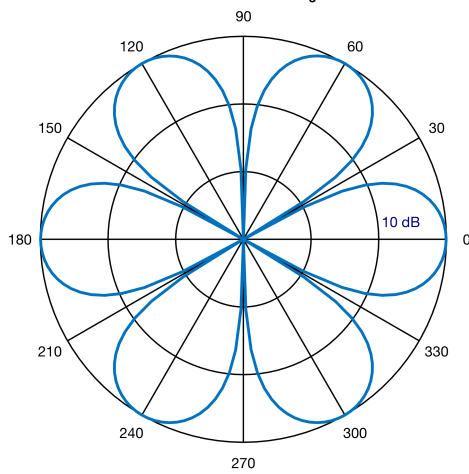
Eigenbeam Y_3^{-3}



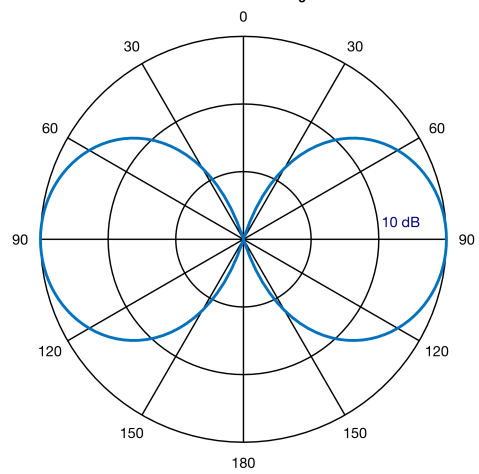
Eigenbeam Y_3^3



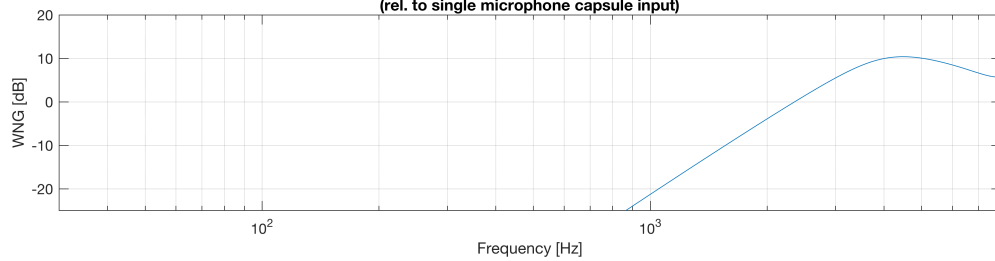
Horizontal Plane (Y_3^3)



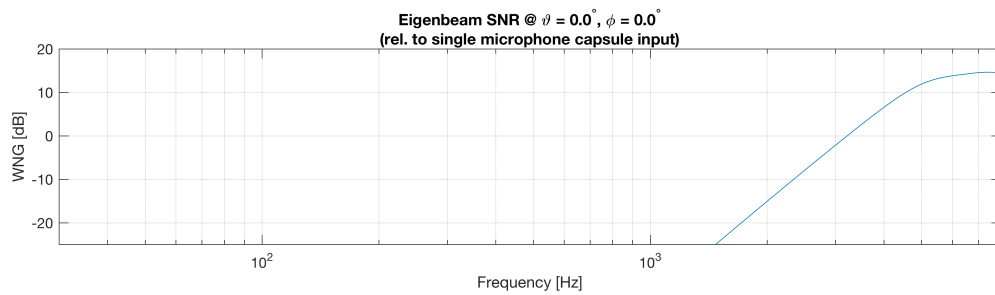
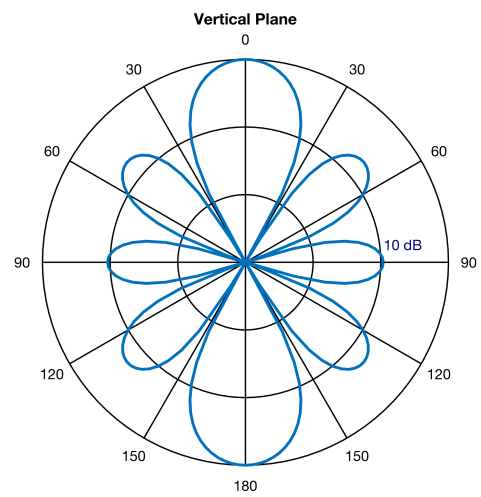
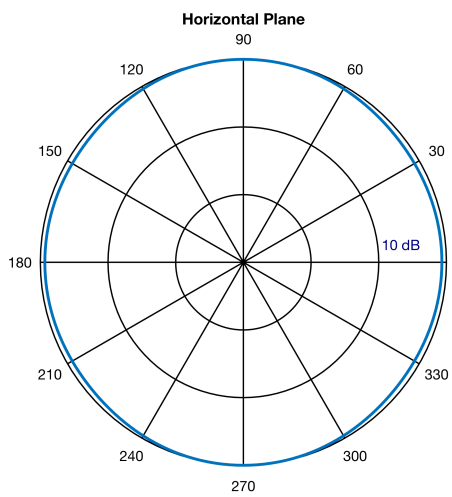
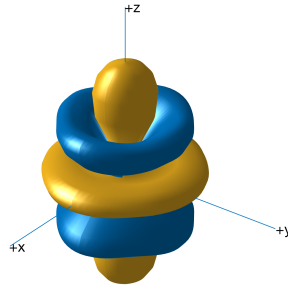
Vertical Plane (Y_3^3)



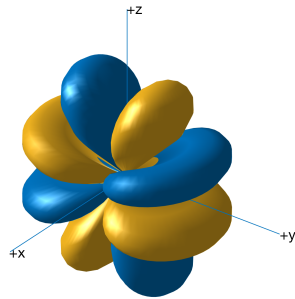
Eigenbeam SNR @ $\theta = 90.0^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)



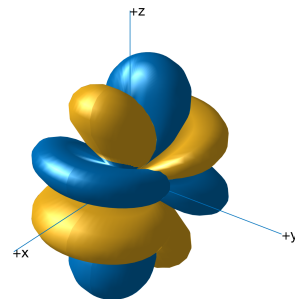
Eigenbeam Y_4^0



Eigenbeam Y_4^{-1}



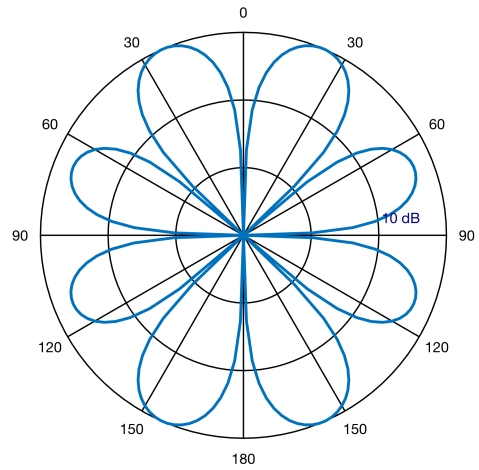
Eigenbeam Y_4^1



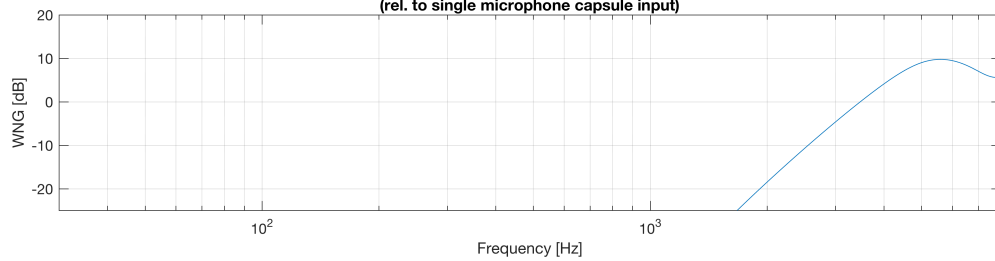
Horizontal Plane (Y_4^1)

N/A

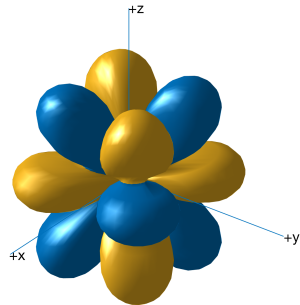
Vertical Plane (Y_4^1)



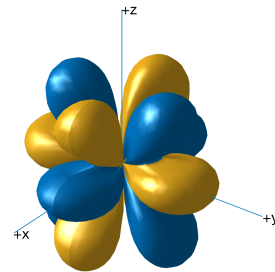
Eigenbeam SNR @ $\theta = 23.9^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)



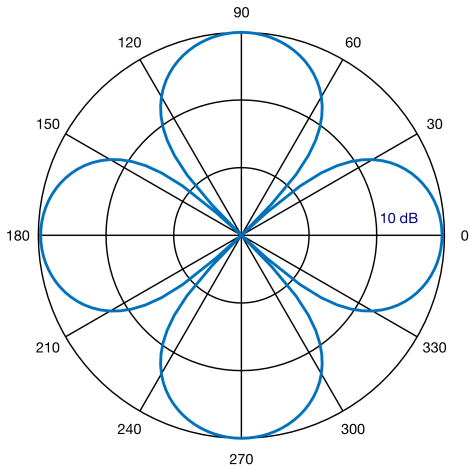
Eigenbeam Y_4^{-2}



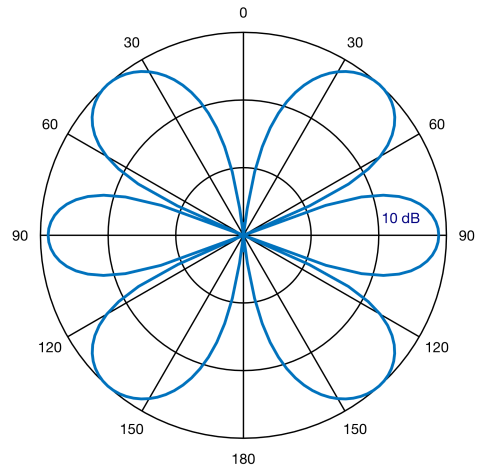
Eigenbeam Y_4^2



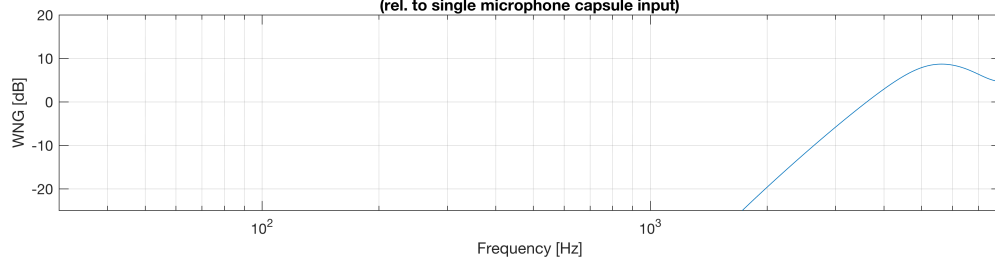
Horizontal Plane (Y_4^2)



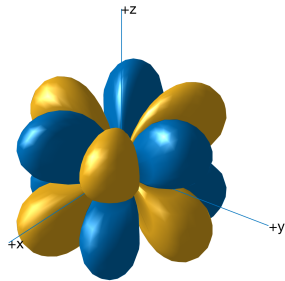
Vertical Plane (Y_4^2)



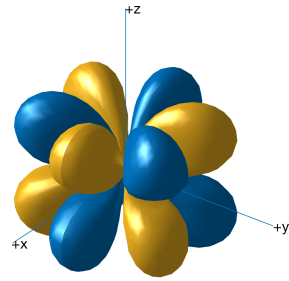
Eigenbeam SNR @ $\theta = 40.9^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)



Eigenbeam Y_4^{-3}



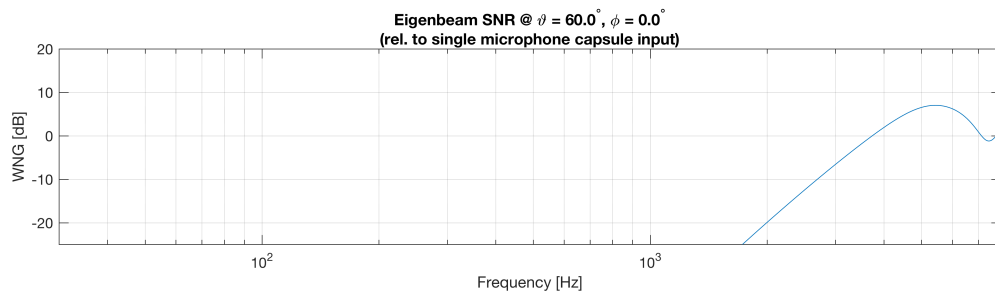
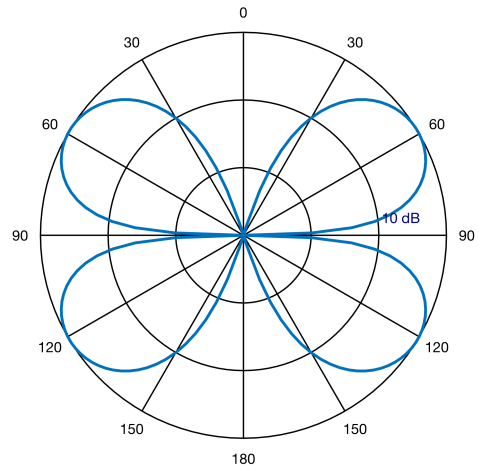
Eigenbeam Y_4^3



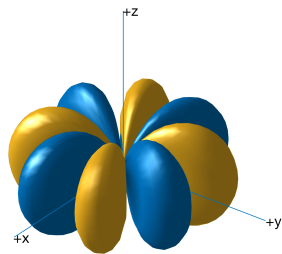
Horizontal Plane (Y_4^3)

N/A

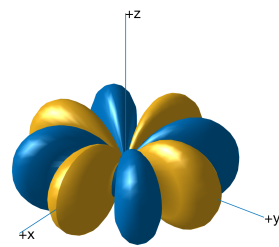
Vertical Plane (Y_4^3)



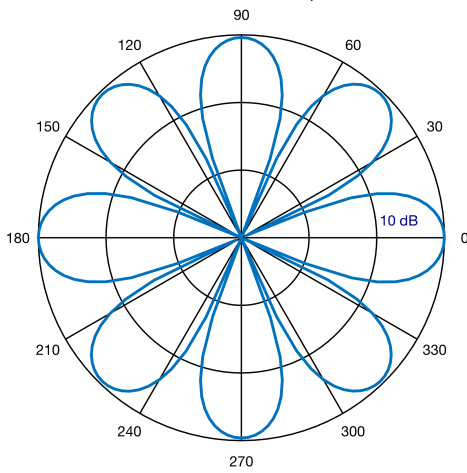
Eigenbeam Y_4^{-4}



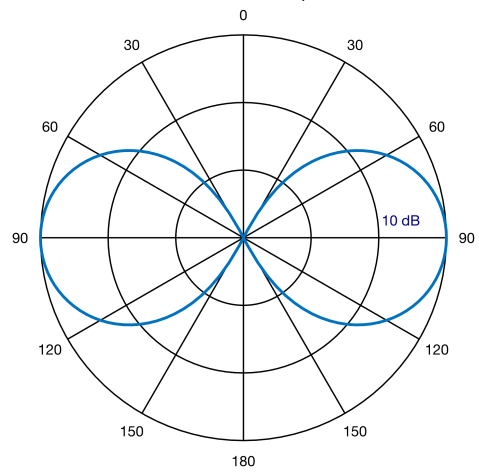
Eigenbeam Y_4^4



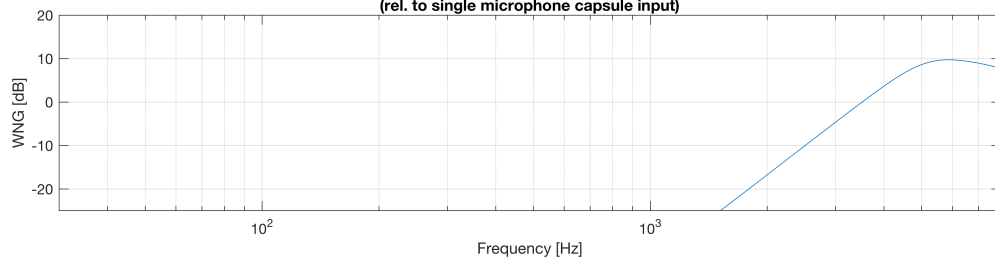
Horizontal Plane (Y_4^4)



Vertical Plane (Y_4^4)



Eigenbeam SNR @ $\theta = 90.0^\circ, \phi = 0.0^\circ$
(rel. to single microphone capsule input)





DATASHEET

Beamformer Data

Specification for Eigenmike Software Beamformer

Version 2

Rev. A

mh acoustics, LLC
25A Summit Ave
Summit, NJ 07901
USA

Phone: +1 908-277-3131

www.mhacoustics.com

Copyright © 2016-2017 mh acoustics, LLC. All rights reserved.

Eigenmike, EigenStudio, and EigenUnits are trademarks of mh acoustics, LLC.

Table of Contents

1.	Introduction.....	4
2.	Definitions and Conventions	5
2.1	Spherical Coordinate System	5
3.	Beamformer Description	7
3.1	Overview	8
3.2	Modal Beamformer	9
3.3	High-Frequency Extension	10
4.	Beamformer Specification.....	11
4.1	Metrics and Plot Types	11
4.1.1	Spatial Response.....	11
4.1.2	Directivity Index.....	11
4.1.3	Beampattern SNR	12
4.1.4	Beamwidth	12
4.2	Specification Data.....	12

1. Introduction

The em32 Eigenmike[®] microphone array is capable of capturing a soundfield with a high degree of spatial accuracy. The 32 raw microphone signals can be processed into Eigenbeams (a.k.a. spherical harmonics, Ambisonic signals) and then linearly combined to form microphone beampatterns up to 4th order. The beamforming process allows the beampatterns to be steered in any direction in 3D space and focus on specific directions in the acoustic field. Beamforming software for the em32 Eigenmike microphone array is available via the stand-alone EigenStudio[®] application or the EigenUnits[®] plugins.¹ See the "EigenStudio User Manual"² and "EigenUnits User Manual"³ documents for more details.

The following document details some of the main characteristics of the beampatterns produced by the beamforming process using mh acoustics' software. This document presents definitions, conventions, and performance specifications, as well as discussion of the overall beamforming process.

¹ <http://mhacoustics.com/download>

² [http://mhacoustics.com/sites/default/files/EigenStudio User Manual R02B.pdf](http://mhacoustics.com/sites/default/files/EigenStudio%20User%20Manual%20R02B.pdf)

³ [http://mhacoustics.com/sites/default/files/EigenUnits User Manual R01D.pdf](http://mhacoustics.com/sites/default/files/EigenUnits%20User%20Manual%20R01D.pdf)

2. Definitions and Conventions

2.1 Spherical Coordinate System

For the em32 Eigenmike array, the following conventions are used when defining the spherical coordinate system (see Figures 1 and 2):

- **Vertical Angle (ϑ):** is the angle in the vertical dimension. In degrees it ranges from 0 to 180. The 0 degrees direction points away from the top of the spherical array (the opposite side from where the shaft mounts to the Eigenmike array; towards the ceiling in a typical arrangement or equivalently the positive z-axis in spherical coordinates). The horizontal plane is in the 90 degrees direction, and the 180 degrees direction is in the direction of the shaft (typically towards the floor or also the negative z-axis in spherical coordinates).
- **Horizontal Angle (φ or ϕ):** is the angle in the horizontal plane (the x-y plane in spherical coordinates). It ranges from 0 to 360 degrees. The 0 degrees direction aligns with the "mh acoustics" logo on the shaft of the Eigenmike array. The angle increases in the counter-clockwise direction looking from the top of the Eigenmike array.

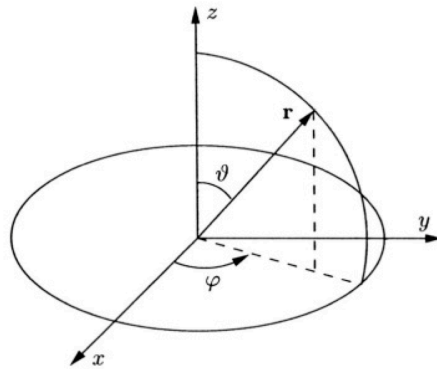


Figure 1: Spherical Coordinate Conventions



Figure 2: Coordinate Axes w.r.t. em32

3. Beamformer Description

The following sections detail processing and performance specifications for the beamformer available in the latest versions of the Eigenmike software⁴. The data presented here represents the cumulative output of the signal chain comprising the em32 microphone array and the associated signal processing software (i.e. EigenStudio or EigenUnits). The data has been derived from simulations using the actual software implementation, and has been proven to agree well with real-world measurements made of the em32 microphone array.

⁴ EigenStudio v2.8.5 and EigenUnits v0.9, May 2017

3.1 Overview

An overview of the signal chain comprising the complete beamforming process is shown in *Figure 3*. Upon streaming into the Eigenmike software, the 32 raw microphone signals are calibrated based on the factory-measured calibration values that are stored in firmware inside each em32 array. These values are used to compensate for microphone amplitude mismatch to improve subsequent beamformer performance.

Next, the 32 calibrated microphone signals are encoded into Eigenbeams, up to 4th order, by the *Eigenbeam Encoder*. For more information about the Eigenbeam encoding process and performance, please see the "Eigenbeam Datasheet"⁵.

The encoded Eigenbeams are then input to the *Modal Beamformer*, which linearly combines the Eigenbeams in order to realize the desired beampattern. This module also handles the steering of the beampattern in 3D space. See Section 3.2.

A parallel path to the encoding/beamforming signal processing implements the high-frequency extension (*HF Extension*) for signals above 9 kHz. The high-frequency extension signal path is discussed in more detail in the Section 3.3.

Finally, the main beamformer output and high-frequency extension path output signals are combined via a crossover to realize the full-band output signal.

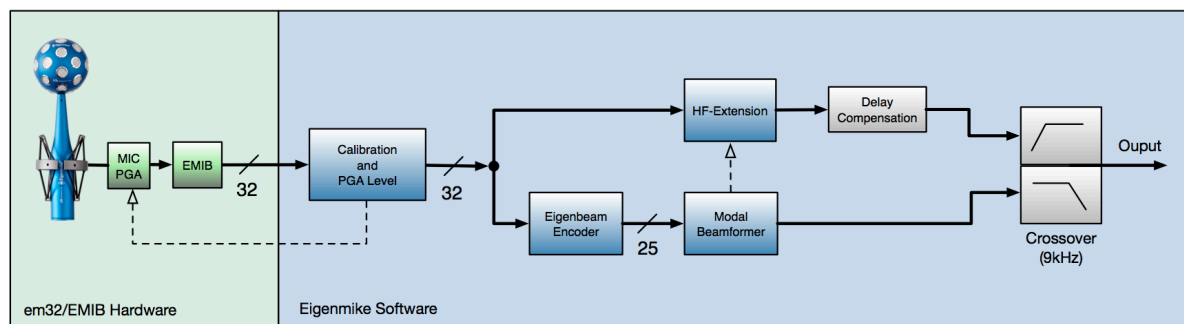


Figure 3: Beamformer Signal Flow

⁵ https://mhacoustics.com/sites/default/files/Eigenbeam%20Datasheet_R01A.pdf

3.2 Modal Beamformer

The *Modal Beamformer* linearly combines the Eigenbeams to realize the desired beampattern. This module also handles the steering of the beampattern in 3D space.

The following microphone beampatterns are available:

- **Omnidirectional** (0th order) – Equal sensitivity in all directions
- **Cardioid** (1st – 4th order) – Null(s) at 180°
- **Supercardioid** (1st – 4th order) – Maximum front-to-back power ratio
- **Hypercardioid** (1st – 4th order) – Maximum directivity
- **Dipole** (1st order) – Maximum sensitivity at 0°/180°. Nulls at 90°/270°

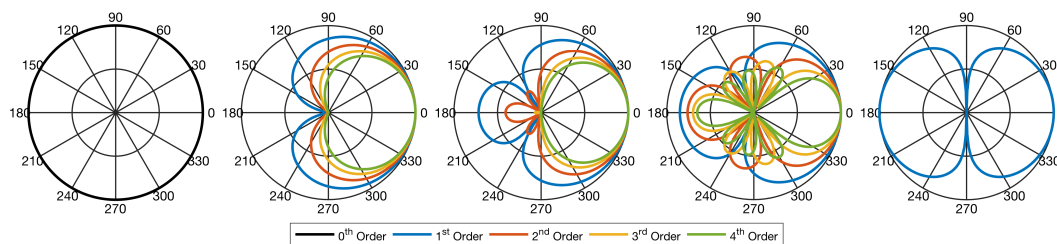


Figure 4: Comparison of Beampattern Types

(from left to right: Omnidirectional, Cardioid, Supercardioid, Hypercardioid, Dipole)

In practice, the higher-order beampatterns are restricted to limited operating frequency ranges in order to ensure a minimum beampattern SNR threshold is achieved. Thus using a white-noise-gain constrained design results in higher-order beampatterns that effectively “fall back” to progressively lower-order beampatterns at lower frequencies. This smooth transition can be seen in the Beamformer Specification plots in Section 4.

The *Modal Beamformer* is designed so that the beampatterns have a nominally flat on-axis magnitude frequency response. Note that additional application-specific equalization may be required and should be applied externally as needed. Since the beamformer is an entirely linear, time-invariant process, any EQ can be applied either to the raw input signals, or, more efficiently, directly to the beamformer output signal.

3.3 High-Frequency Extension

In parallel to the main beamforming signal path, an additional signal path implements a “high-frequency extension” as depicted in *Figure 3*. Due to the fixed geometry of the sensors on the sphere, a *spatial* Nyquist frequency exists beyond which the beampatterns become significantly *spatially* aliased, degrading the desired spatial characteristics of the beamformer.

To overcome this, the high-frequency extension is used to replace the spatially aliased portion of the main beamformer’s output signal with a more spatially defined beampattern that is steerable in 3D space. As seen in *Figure 3*, the main beamformer output and high-frequency extension path output signals are combined via a crossover to realize the complete output signal.

The *HF Extension* module is linked to the *Modal Beamformer* to allow choosing a lower or higher directivity extension beampattern appropriate for pairing with the main beamformer’s beampattern. Furthermore, steering information about the main beamformer’s steering direction is communicated to the *HF Extension* to allow steering of the high-frequency extension pattern towards the desired beam direction⁶.

Finally, additional delay and equalization is applied to the high-frequency extension signal path in order to match the main beamformer signal path and desired frequency response. The delay is realized internally in the EigenStudio software; however, when using the modular EigenUnits plugins, it is critical that the user implement this delay explicitly in their audio processing software (DAW). For more information, please see the “EigenUnits User Manual”⁷.

⁶ Note that even the omnidirectional beampattern - which by definition does not favor a specific look direction - should still provide steering information to the *HF Extension*. The *HF Extension* always has some directivity, so it is advisable that the user still specify a look direction when using the omnidirectional pattern. This is analogous to the setup and pointing of a traditional single-diaphragm omnidirectional microphone, which also typically has some directivity as wavelengths begin to “see” the microphone.

⁷ <https://mhacoustics.com/download>

4. Beamformer Specification

In this section details are given on the performance of each of the individual microphone beampatterns available in the Eigenmike software⁸. Details are given regarding their spatial and spectral response, and other performance metrics.

All beampatterns are shown steered to the nominal angle of $\vartheta=90^\circ$, $\varphi=0^\circ$. For a typical orientation of the em32 microphone array, this corresponds to a beam in the horizontal plane pointed in the forward direction, along the positive x-axis (see Section 2.1). The *Frequency Response*, *Directivity Index*, and *Beampattern SNR* plots all show the value of these metrics for this look direction.

4.1 Metrics and Plot Types

4.1.1 Spatial Response

The microphone spatial responses are depicted in multiple ways: a *3D Balloon* plot (for basic reference only), a *Waterfall of Directivity* plot, and standard *Polar Diagram* plots. The waterfall plot shows the directivity variation across both angle and frequency, and is one of the more comprehensive representations of the spatial response. However, it is more difficult to interpret with accuracy since the response amplitude is represented by color.

The *Polar Diagrams* allow for a more accurate interpretation of spatial response at a specific angle of incidence; these are displayed for a select group of frequencies. Per standard practice, all polar diagram plots are normalized to the maximum sensitivity in the displayed plane. Polar patterns for frequency regions that lie within the high-frequency extension are shown with dashed lines.

Both the *Waterfall* and *Polar Diagram* plots show the directivity of the beampattern in a 2D plane (e.g. the horizontal plane when the beampattern is steered to $\vartheta = 90^\circ$ and $\varphi = 0^\circ$). Since the available microphone beampatterns are axisymmetric around the axis pointing in the look direction, the spatial response remains the same in all other planes by this rotation.

4.1.2 Directivity Index

The *Directivity Index* plots show the 3D directivity index. The directivity index (DI) is a useful measure of the amount of directional gain of the array in a spherically diffuse sound field. It gives the amount of SNR increase due to the beamformer in a spherically isotropic noise field.

Given the beamformer response, $E(\omega, \vartheta, \varphi)$, the directivity index at angular frequency ω , and look direction $[\vartheta_0, \varphi_0]$ is given by:

$$DI(\omega, \vartheta, \varphi) = 10 \log \left[\frac{4\pi |E(\omega, \vartheta_0, \varphi_0)|^2}{\int_0^{2\pi} \int_0^\pi |E(\omega, \vartheta, \varphi)|^2 \sin \vartheta \, d\vartheta d\varphi} \right]$$

⁸ All plots depict the performance in EigenStudio when using the 0dB WNG setting.

4.1.3 Beampattern SNR

The *Beampattern SNR* plots show the *change* in SNR for the realization of a specific beampattern *relative to the SNR of a single microphone capsule*. This is referred to as the white-noise-gain. The white-noise-gain (WNG) is an important metric in beamformer performance. The white-noise-gain can be used to quantify the output SNR of the beamformer from knowledge of the noise of the microphone and electronics used in the beamformer.

A positive white-noise-gain indicates that the beamformer output SNR (where the noise here is uncorrelated self-noise between the channels and not spatial noise) will be higher than the SNR of single microphone element. Similarly, a negative white-noise-gain indicates that the output SNR of the beamformer will be lower than that of a single microphone element. A white-noise-gain of 0 dB means that the beamformer output will have the same SNR as a single microphone element (where there is an implicit assumption that each microphone element has the same SNR).

The absolute SNR of the beampattern can be calculated by adding (in dB) the beampattern's WNG value to the individual microphone capsule SNR as specified by the hardware datasheet^{9,10}.

4.1.4 Beamwidth

Finally, the *Beamwidth* plots depict the 3dB beamwidth. The beampattern's beamwidth is defined as the angle enclosed between the -3dB points (relative to the main beam peak direction) in the beampattern's directional response.

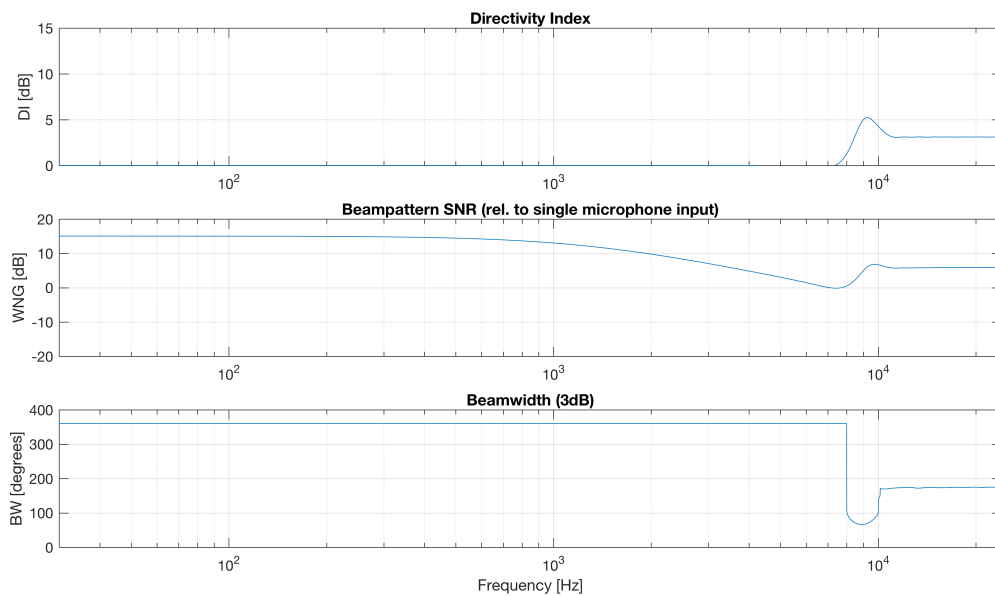
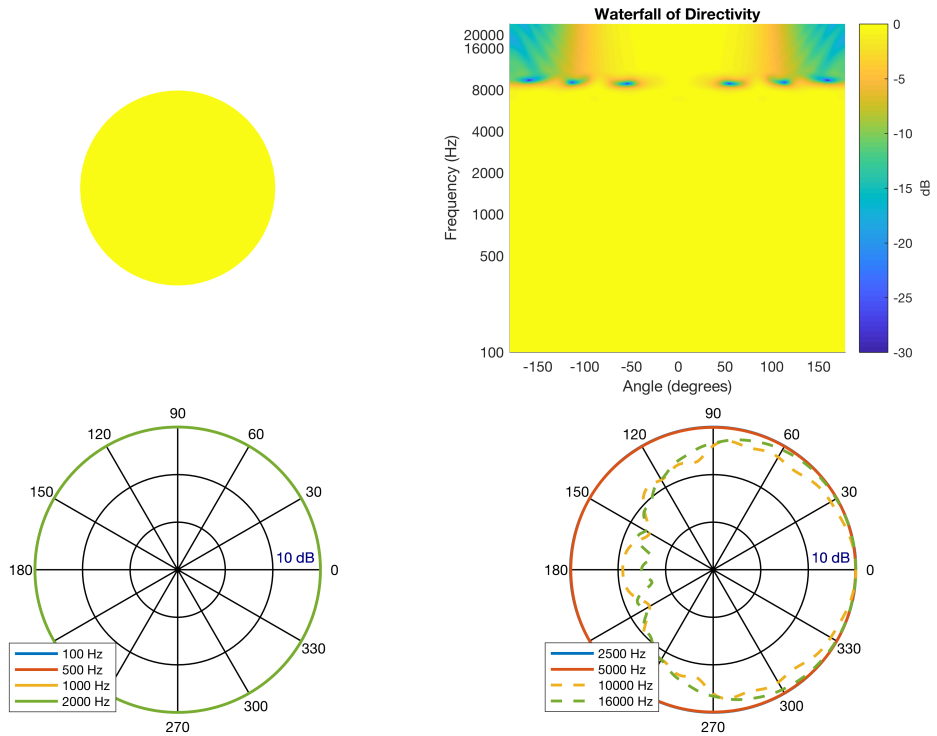
4.2 Specification Data

The beamformer specification data for each beampattern is shown on the following pages.

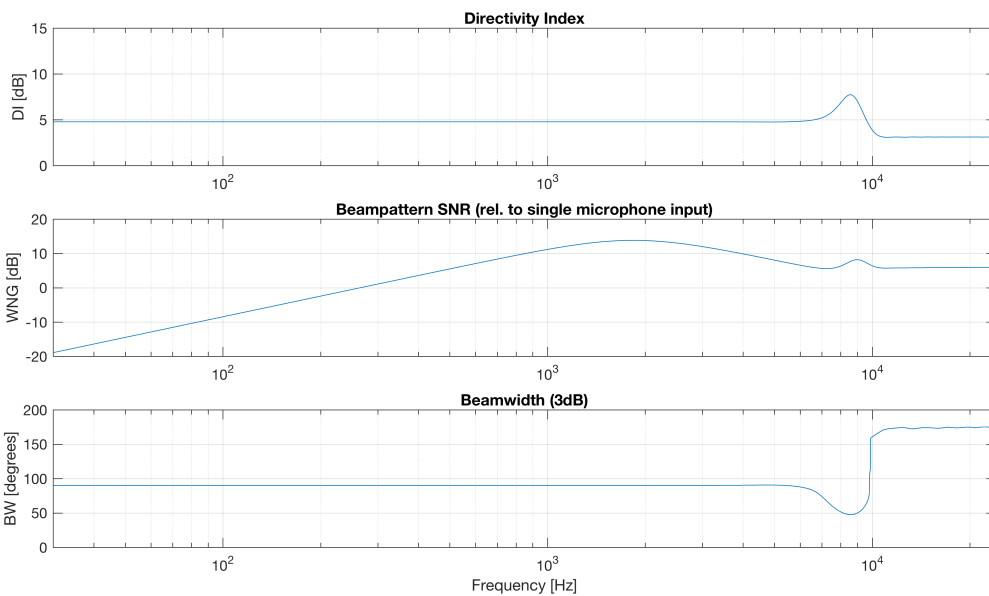
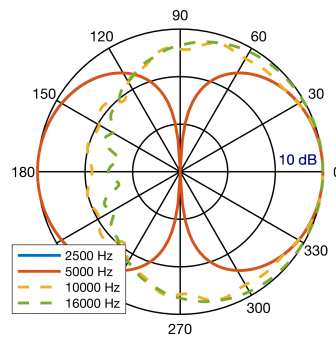
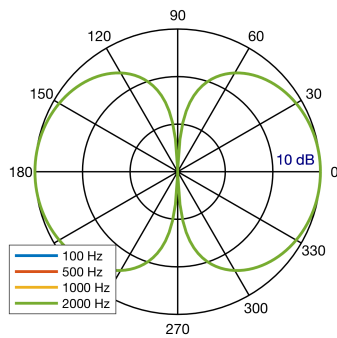
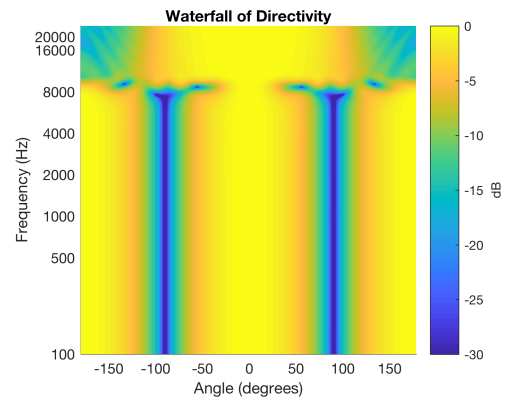
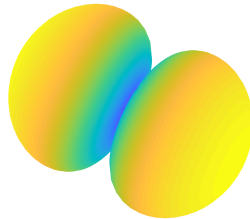
⁹ Refer to the "em32_datasheet" included with the em32 hardware for more details.

¹⁰ For example, the beamformer for the 1st-order cardioid has a WNG of approximately +15dB at 1kHz. Using individual microphone capsules with an SNR of 79dBA (15dBA ENL) would then yield an absolute SNR for this beampattern of 79+15= 94dB at 1kHz.

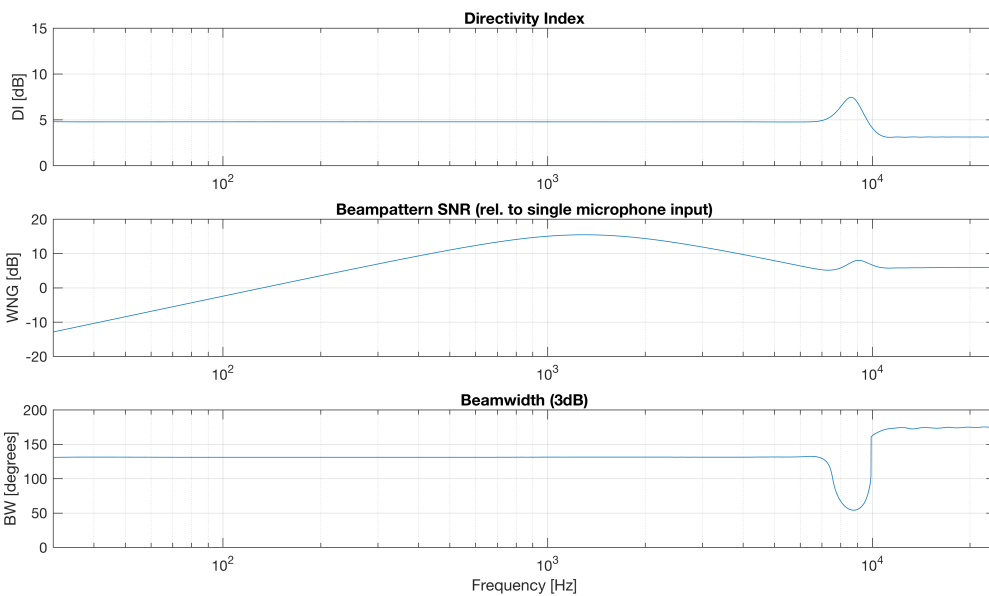
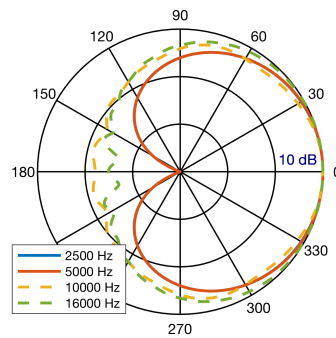
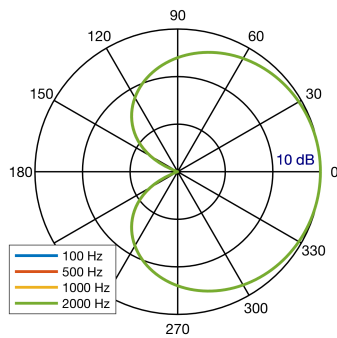
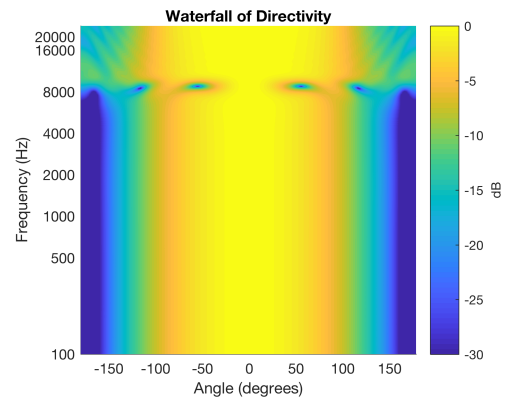
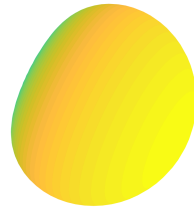
0th-order Omnidirectional



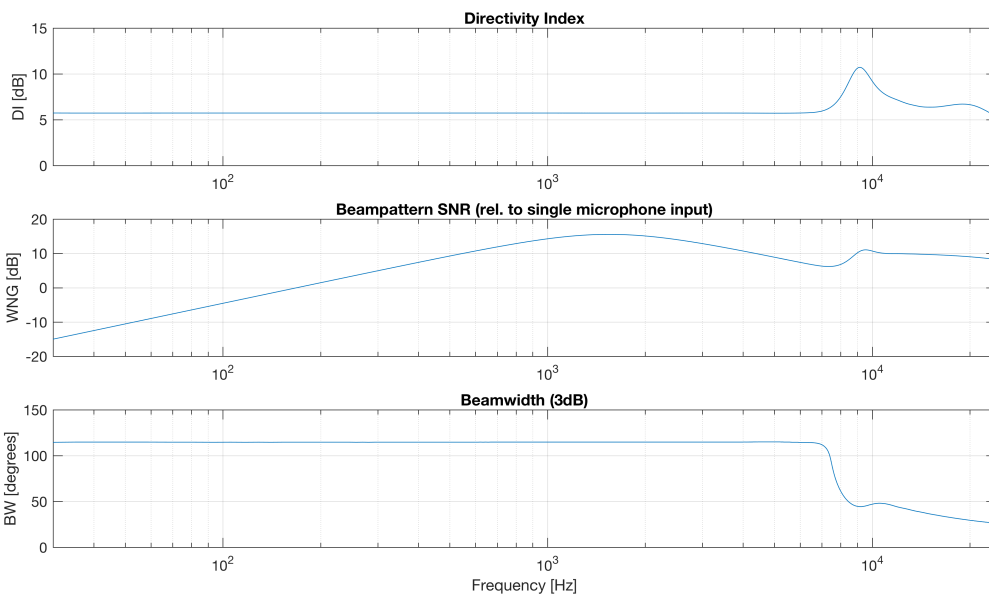
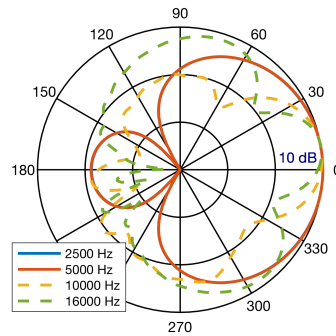
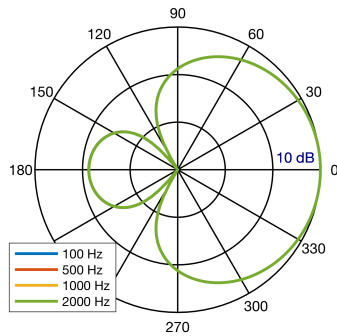
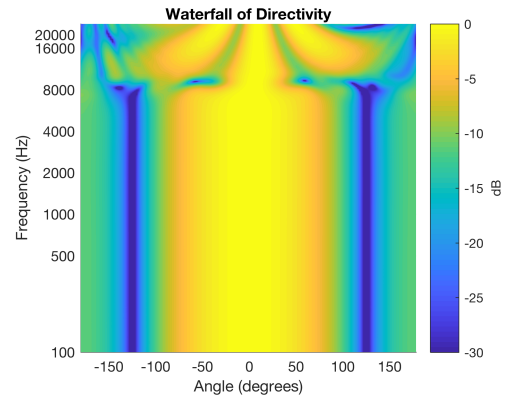
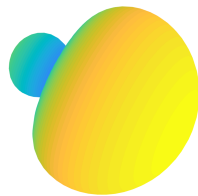
1st-order Dipole



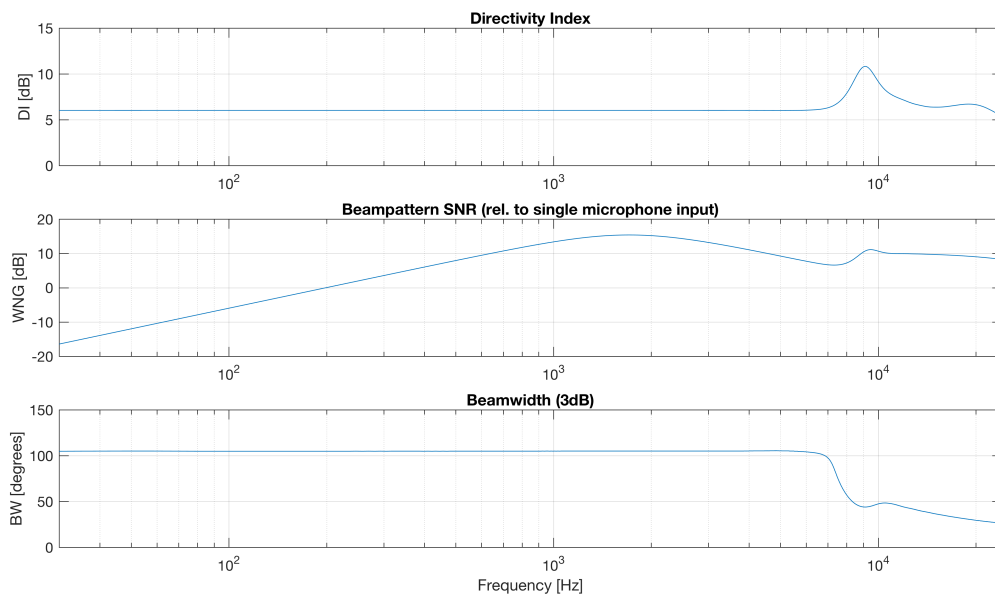
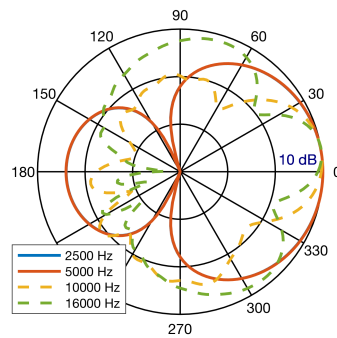
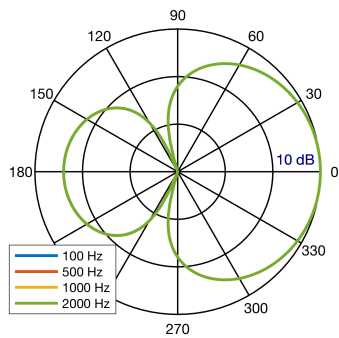
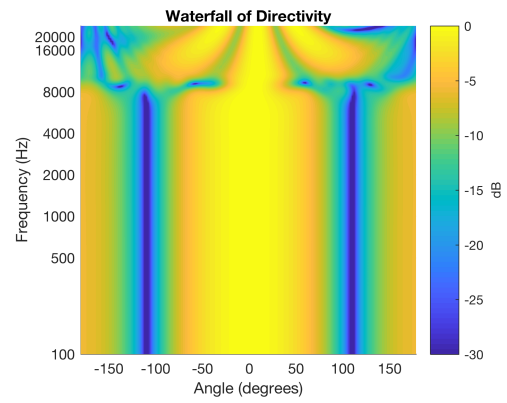
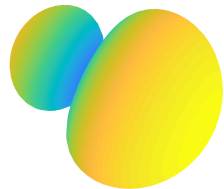
1st-order Cardioid



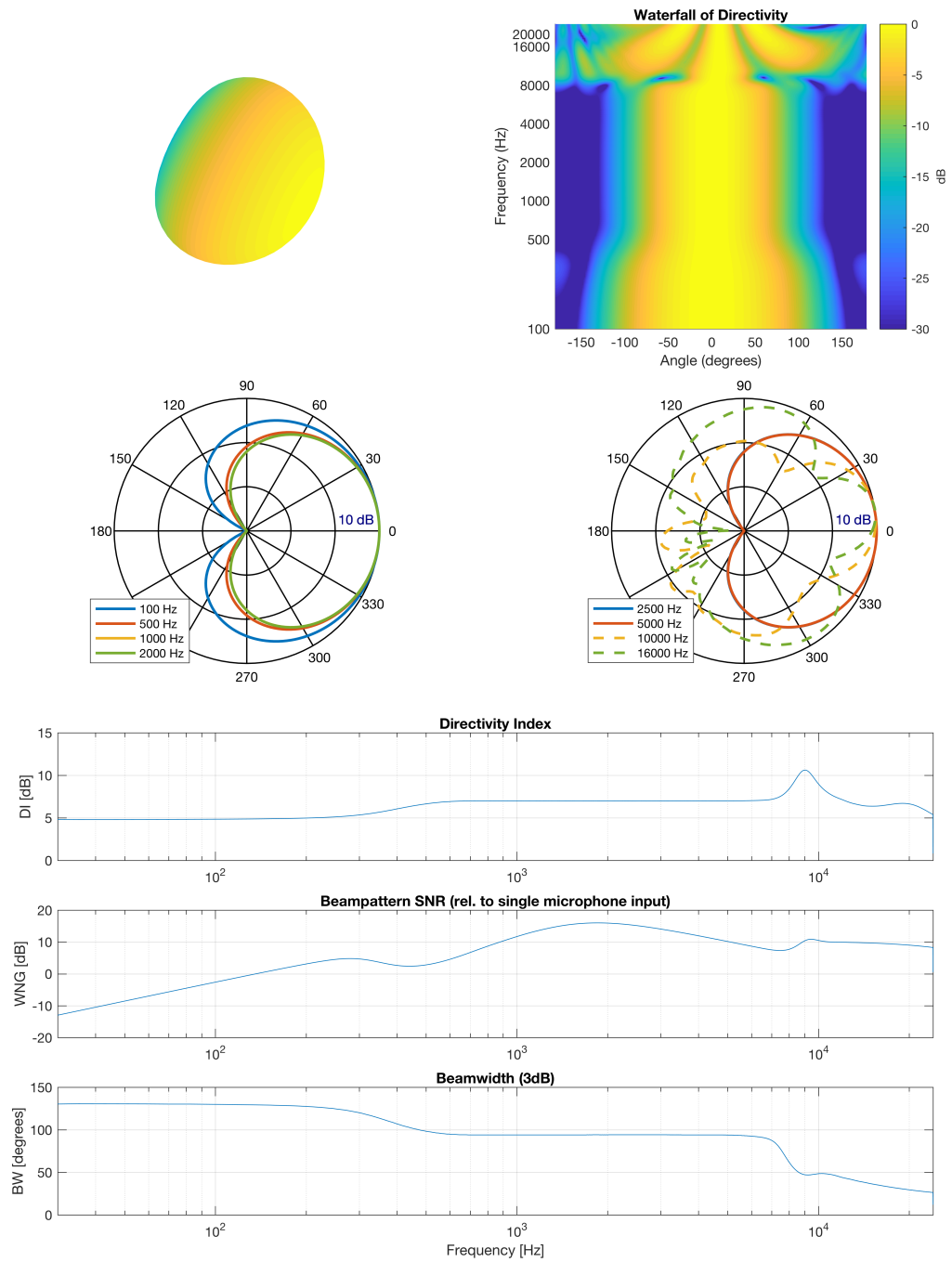
1st-order Supercardioid



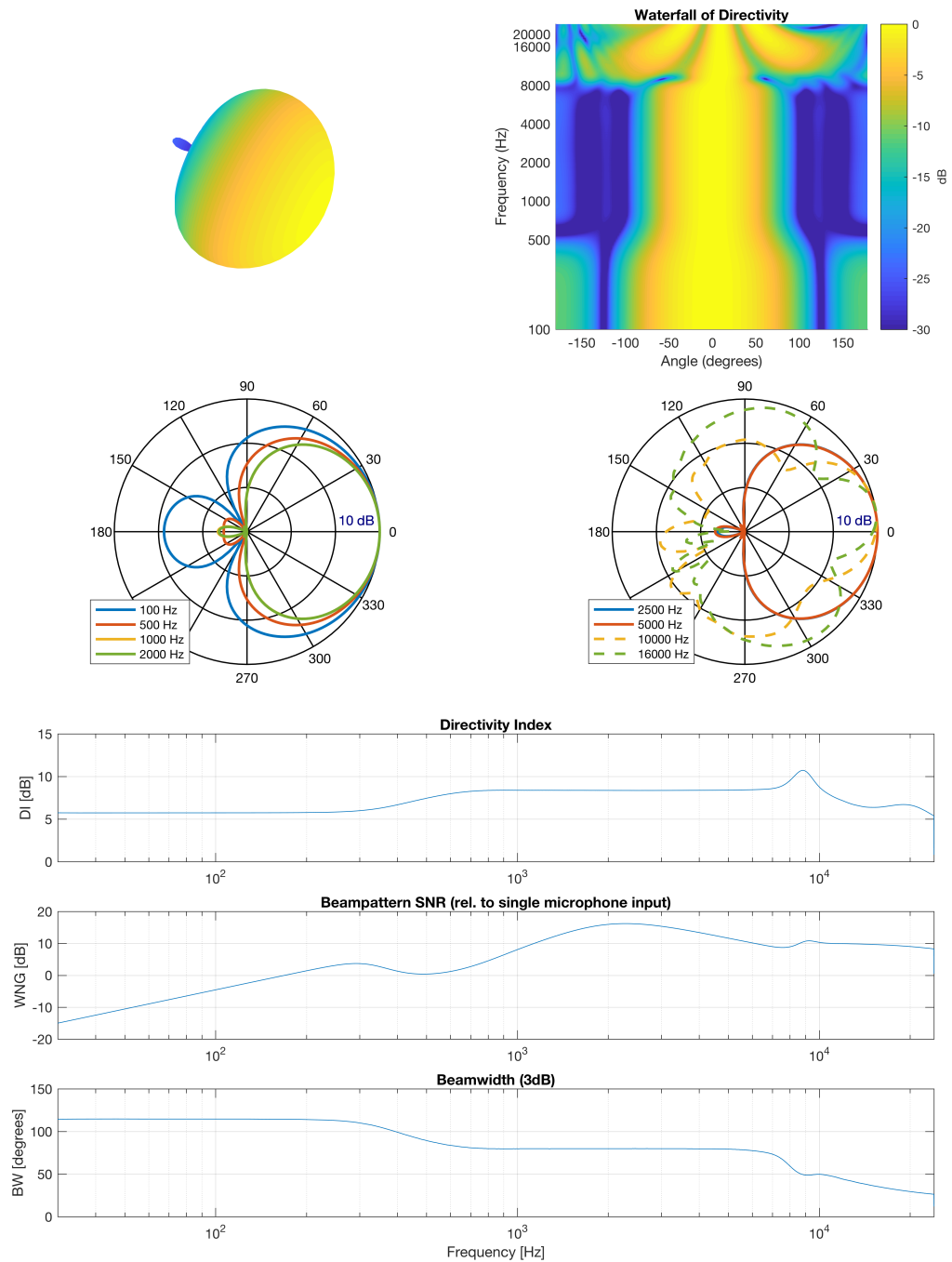
1st-order Hypercardioid



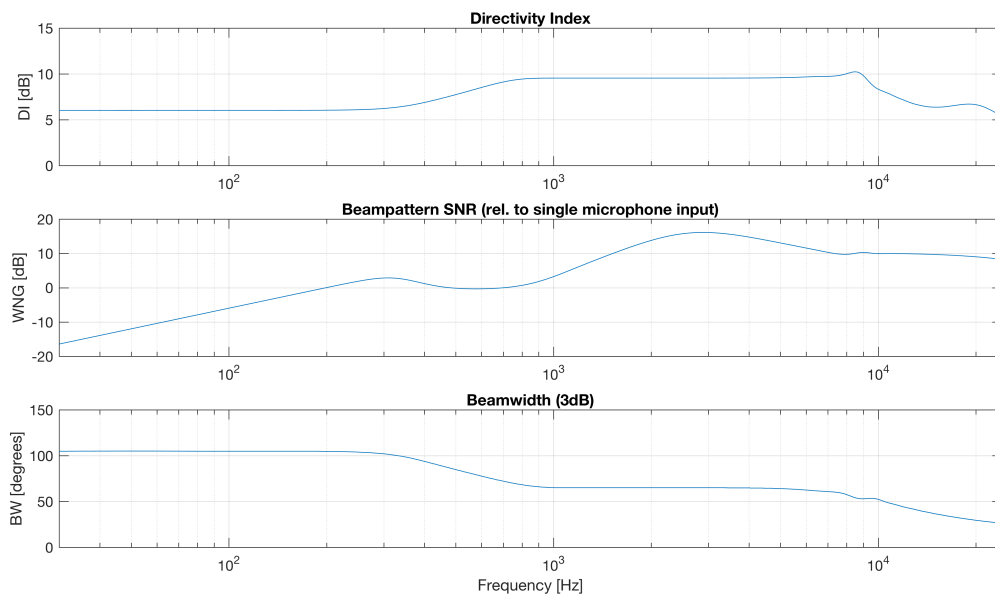
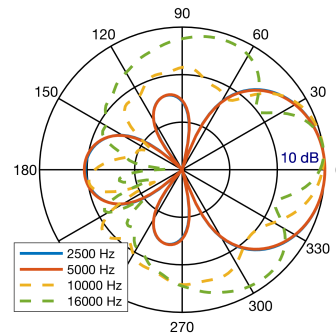
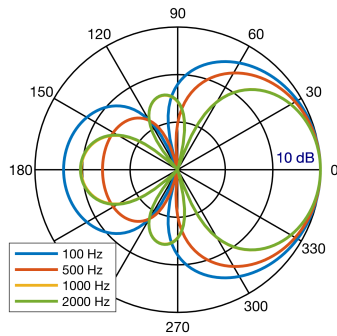
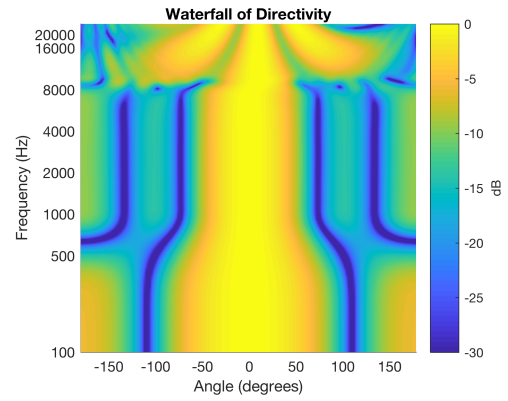
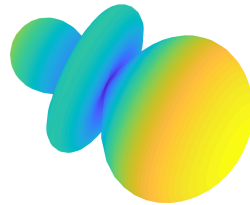
2nd-order Cardioid



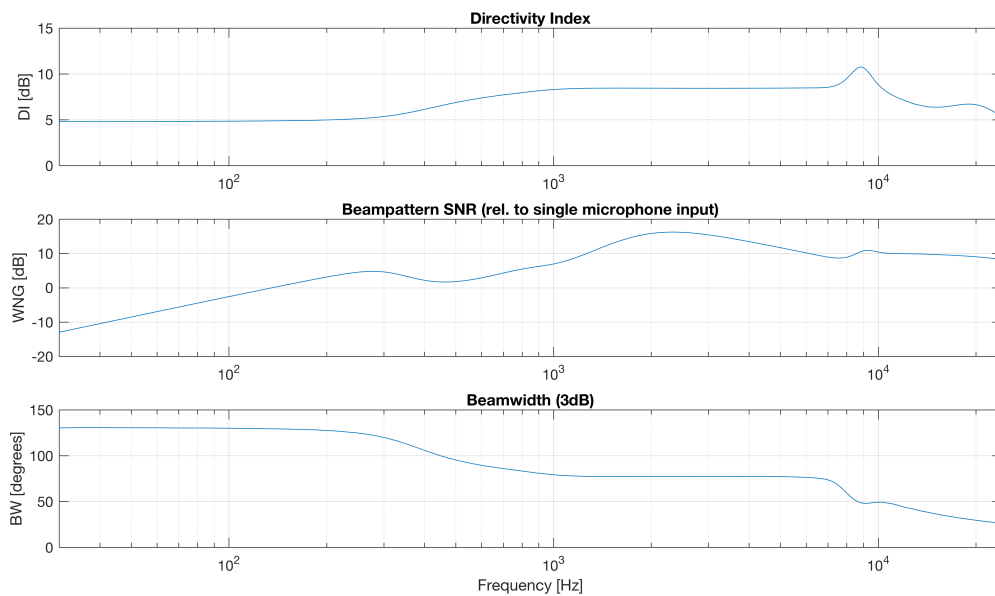
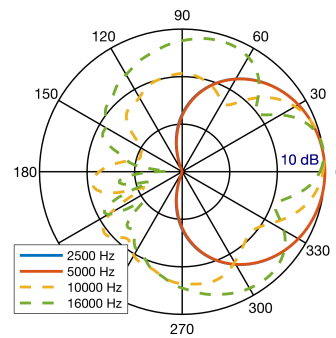
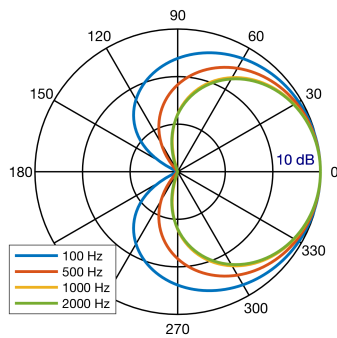
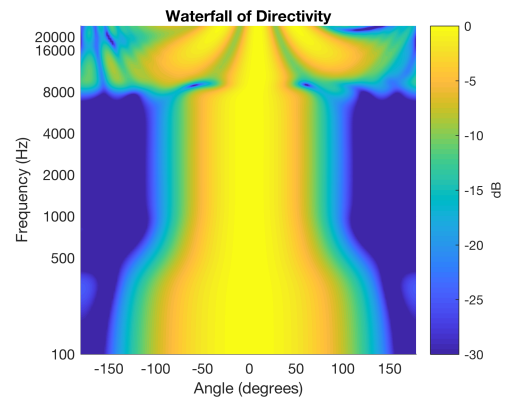
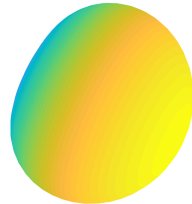
2nd-order Supercardioid



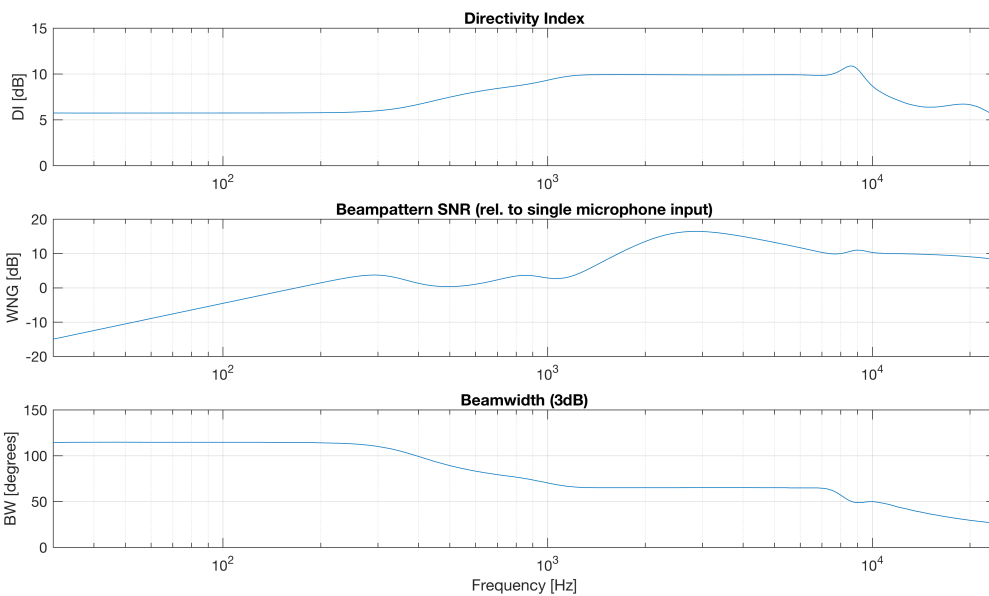
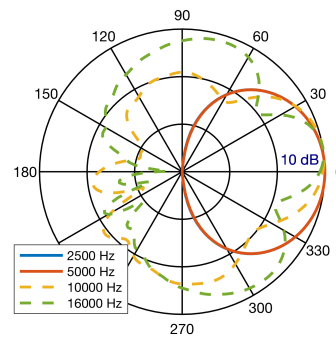
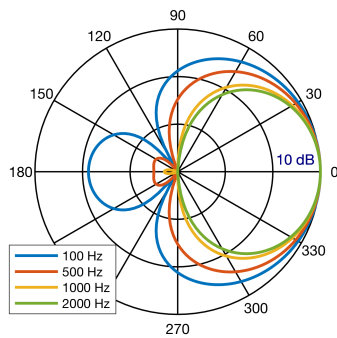
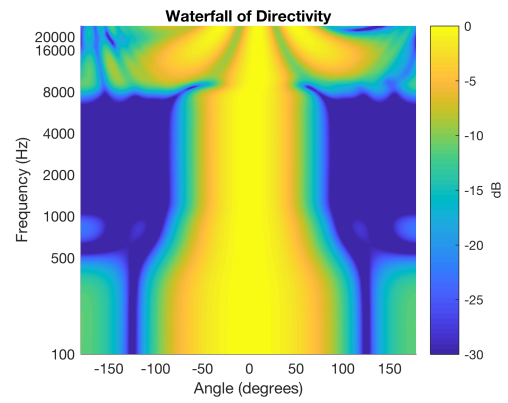
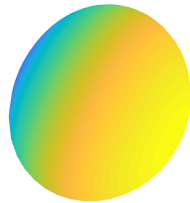
2nd-order Hypercardioid



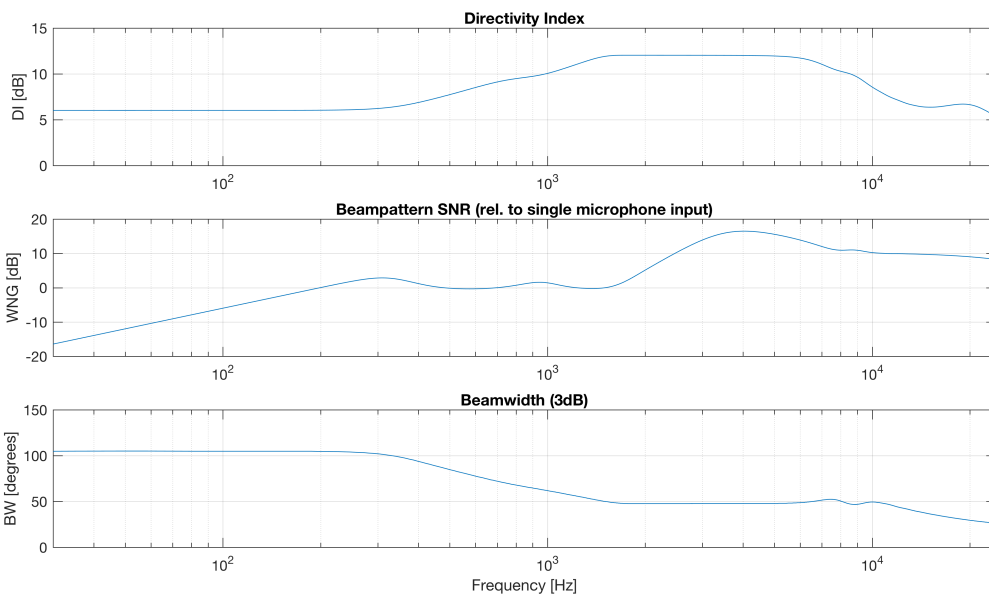
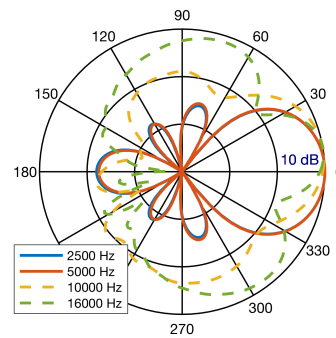
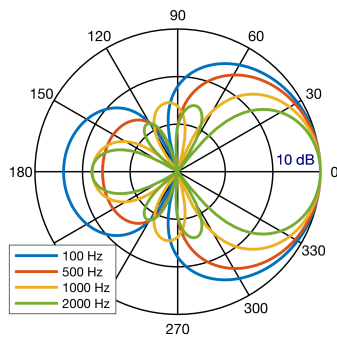
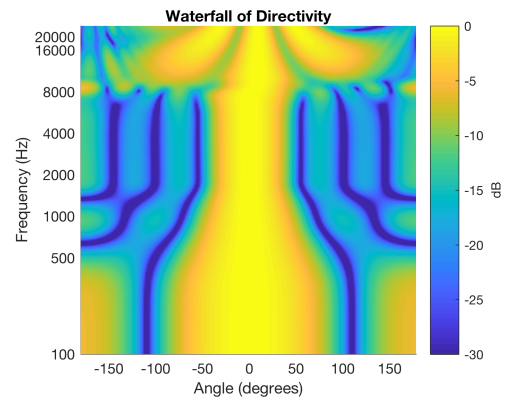
3rd-order Cardioid



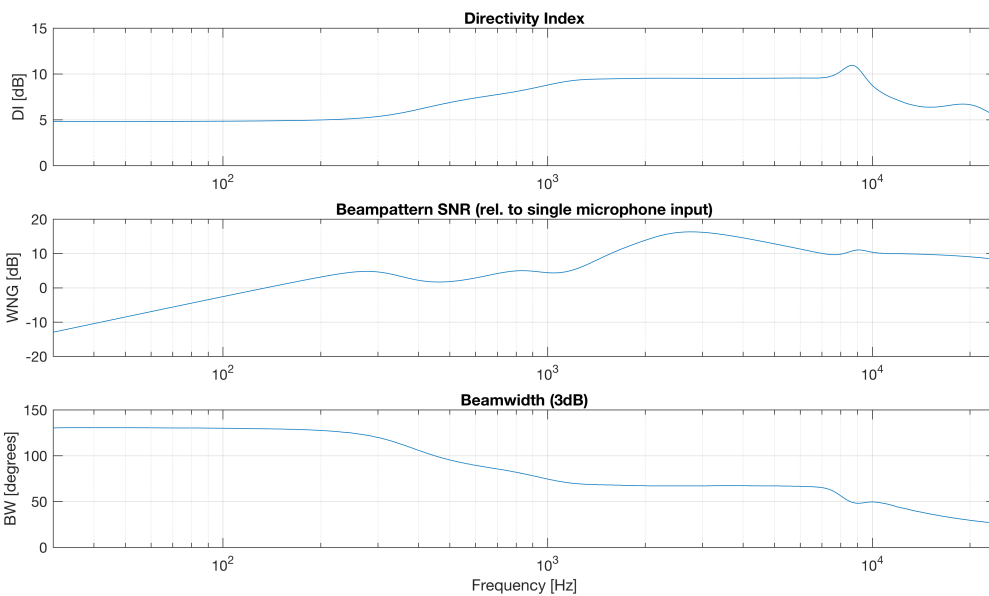
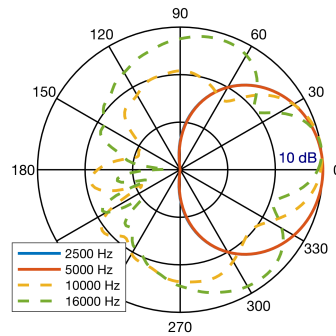
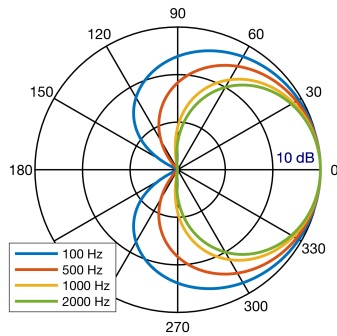
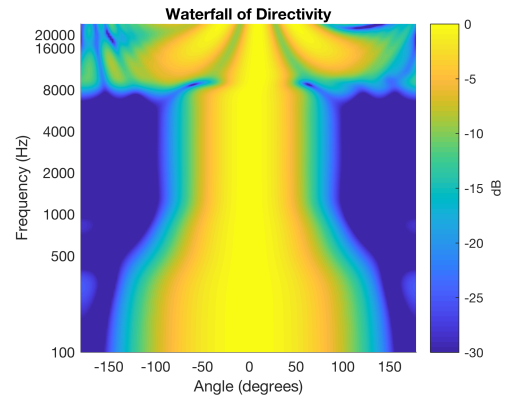
3rd-order Supercardioid



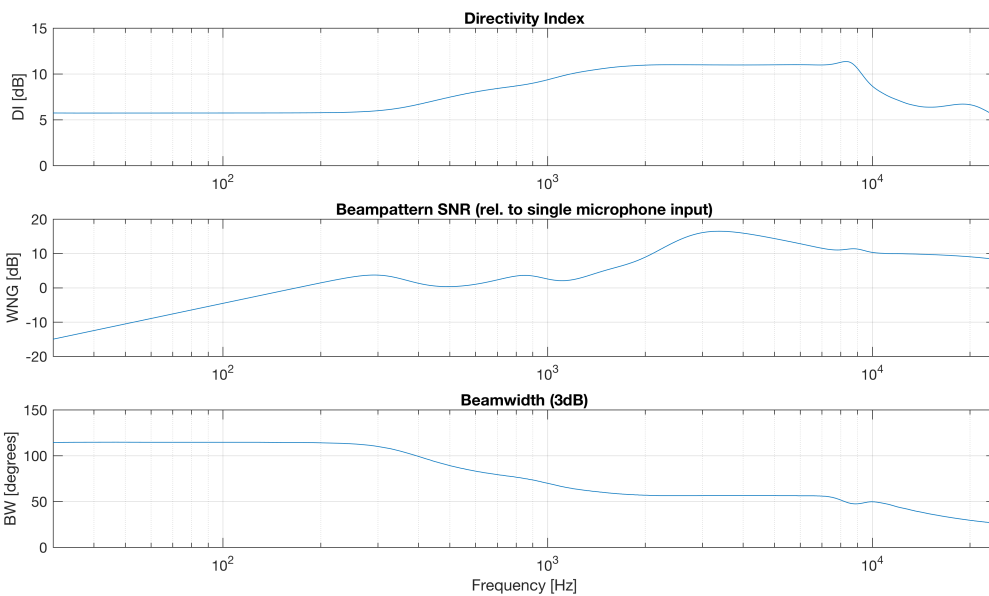
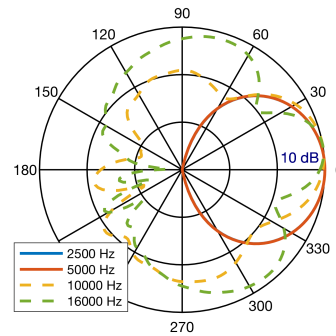
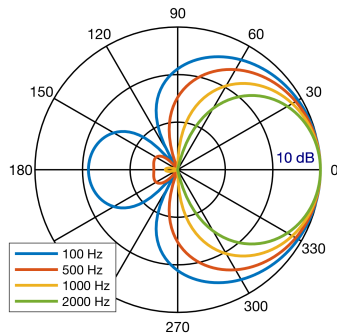
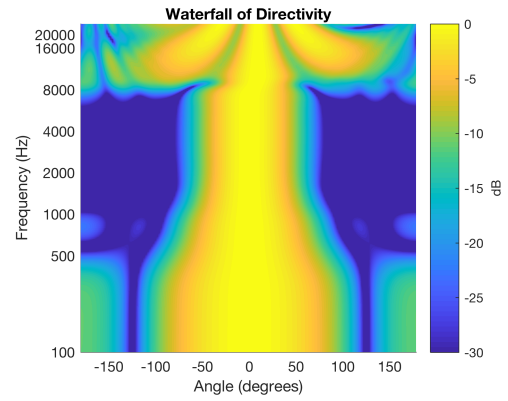
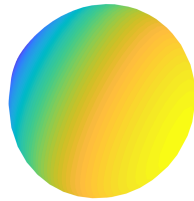
3rd-order Hypercardioid



4th-order Cardioid



4th-order Supercardioid



4th-order Hypercardioid

