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

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

[3] http://mhacoustics.com/sites/default/files/EigenUnits User Manual R01D.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:

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

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

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

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:

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

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:

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

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 variables[4]. 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 conventions[5]. Care should taken to interpret angular conventions appropriately.

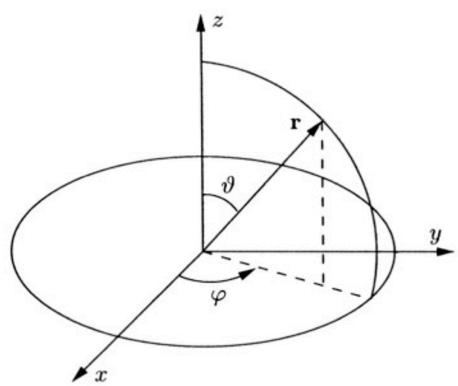


Figure 1: Spherical Coordinate Conventions

---

[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° (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:

- **ambiX**[6]: *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.

[6] http://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	
$Y_1^{\ \pm1}$	-4.8 dB	-4.8 dB	
$Y_2^{\ 0}$	-7.0 dB	-7.0 dB	
$Y_2^{\pm1}$	-7.0 dB	-5.7 dB	
$Y_2^{\pm2}$	-7.0 dB	-5.7 dB	
$Y_3^{\ 0}$	-8.5 dB	-8.5 dB	
$Y_3^{\ \pm1}$	-8.5 dB	-7.0 dB	
$Y_3^{\ \pm2}$	-8.5 dB	-5.9 dB	
$Y_3^{\ \pm3}$	-8.5 dB	-6.4 dB	
$Y_4^{\ 0}$	-9.5 dB	-9.5 dB	
$Y_4^{\ \pm1}$	-9.5 dB	-8.0 dB	
$Y_4^{\ \pm2}$	-9.5 dB	-6.7 dB	
$Y_4^{\ \pm3}$	-9.5 dB	-6.2 dB	
$Y_4^{\ \pm4}$	-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}	$\tan^{-1}\left(\sqrt{2}\right)$	$\pi/4$
Y_3^{-1}	$\cos^{-1}\left(\sqrt{11/15}\right)$	$\pi/2$
Y_3^{0}	0	0
Y_3^{1}	$\cos^{-1}\left(\sqrt{11/15}\right)$	0
Y_3^{2}	$\tan^{-1}\left(\sqrt{2}\right)$	0
Y_3^{3}	$\pi/2$	0
Y_4^{-4}	$\pi/2$	$\pi/8$
Y_4^{-3}	$\cos^{-1}\left(\sqrt{1/2}\right)$	$\pi/6$
Y_4^{-2}	$\cos^{-1}\left(2\sqrt{1/7}\right)$	$\pi/4$
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}	$\cos^{-1}\left(2\sqrt{1/7}\right)$	0
Y_4^{3}	$\cos^{-1}\left(\sqrt{1/2}\right)$	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 software[8].

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).

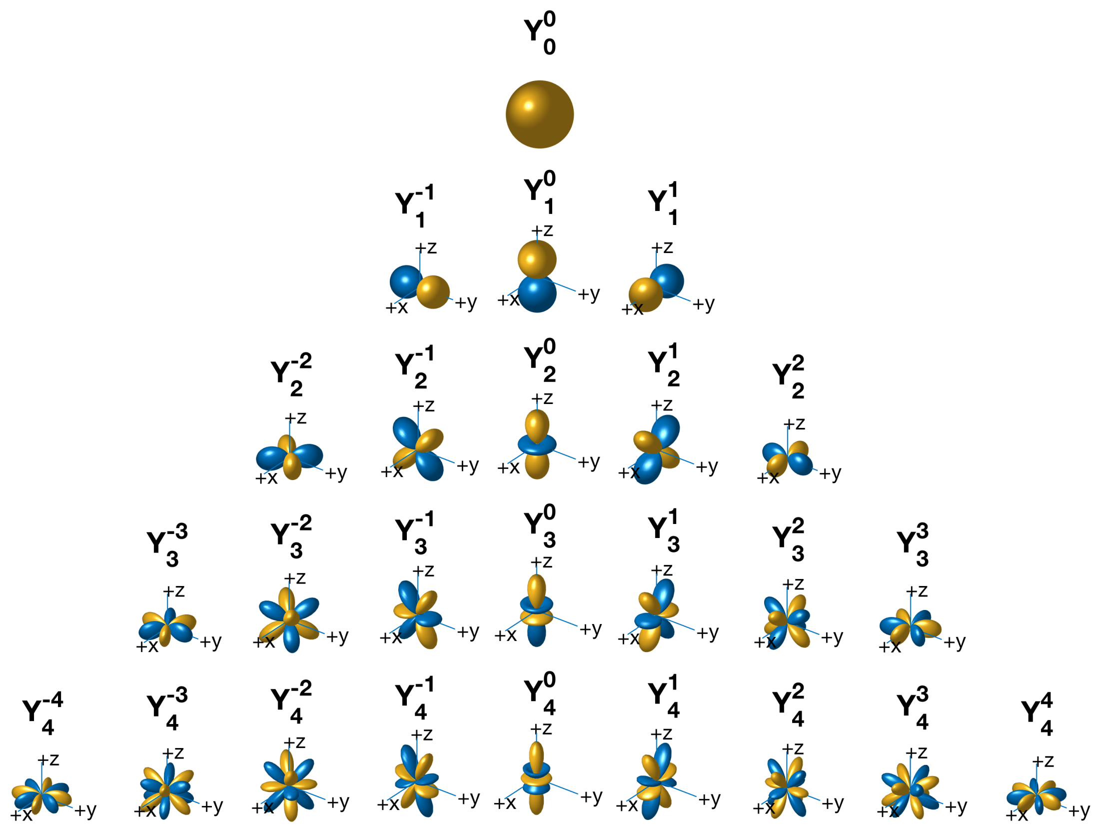


Figure 3: Eigenbeams up to 4th order

[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. 44.1kHz/2 or 48kHz/2). However, all Eigenbeams will exhibit more complex spectral and spatial behavior above the spatial Nyquist limit[9].

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.

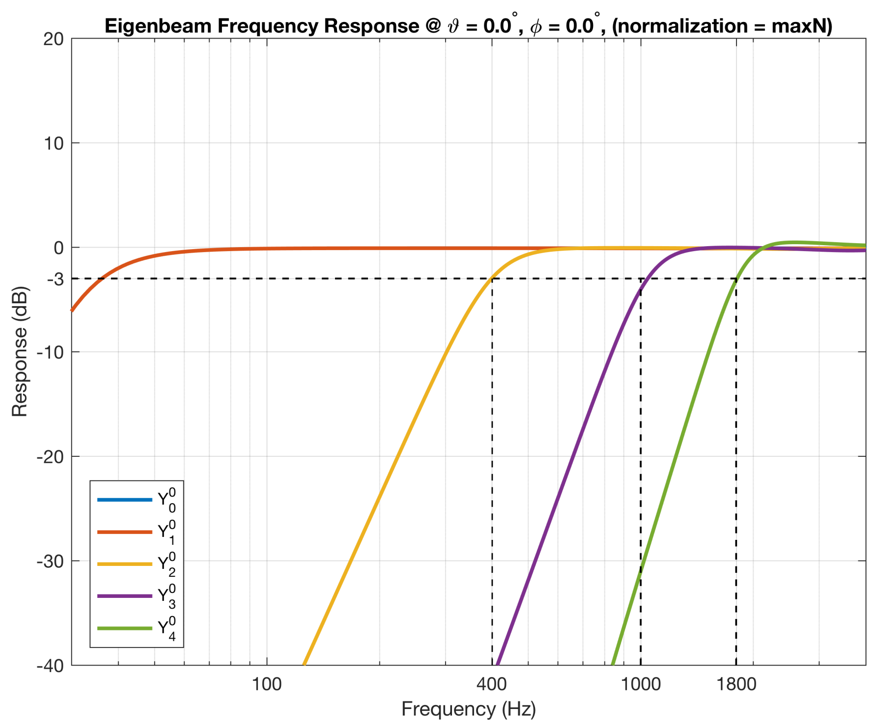


Figure 4: Eigenbeam cutoff frequencies

[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.

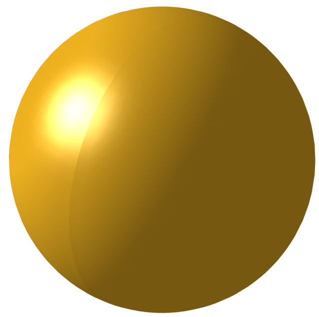
The Eigenbeam SNR plots represent the *change* in SNR for the encoding of a specific Eigenbeam *relative to the SNR of a single microphone capsule.* This is referred to as the White-Noise-Gain (WNG). The absolute SNR of the Eigenbeam can be calculated by adding (in dB) the Eigenbeam's WNG value to the individual microphone capsule SNR as specified by the hardware datasheet[10][11]. Note that these values are specific to the em32 Eigenmike hardware and software.

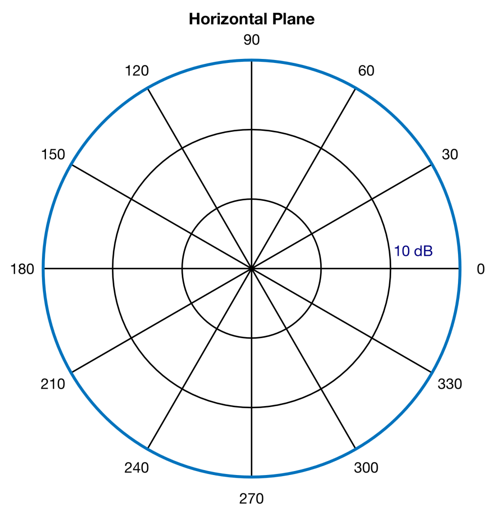
Also, for these plots, data is depicted only up to spatial Nyquist. However, it should be noted that the Eigenbeam software encoding process does generate signals extending all the way up to temporal Nyquist (i.e. 44.1kHz/2 or 48kHz/2).

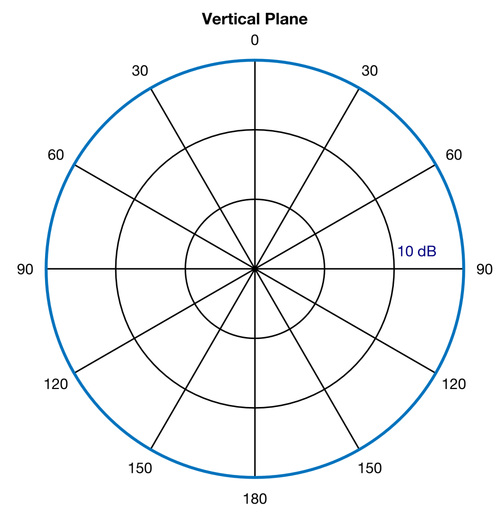
[10] Refer to the "em32_datasheet" included with the em32 hardware for more details.

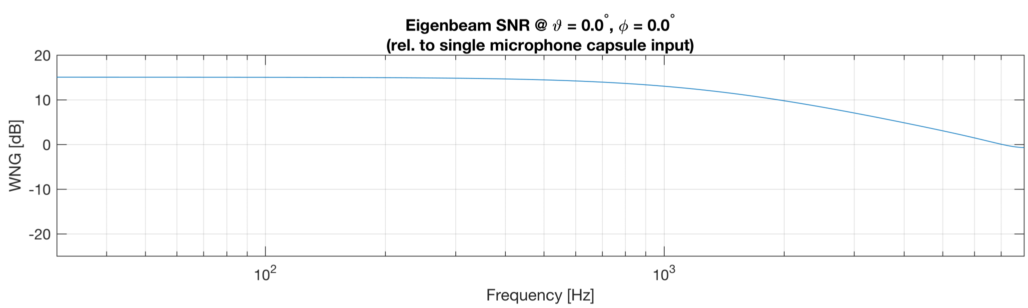
[11] For example, the encoding process for Y_1^1 yields a WNG of approximately +11dB at 1kHz. Using individual microphone capsules with an SNR of 79dBA (15dBA ENL) would then yield an absolute SNR for Y_1^1 of 79+11= 90dB at 1kHz.

Eigenbeam Y_0^0

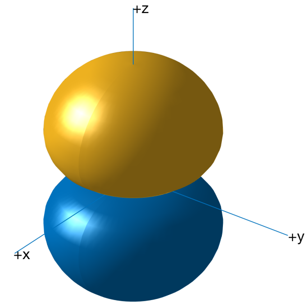








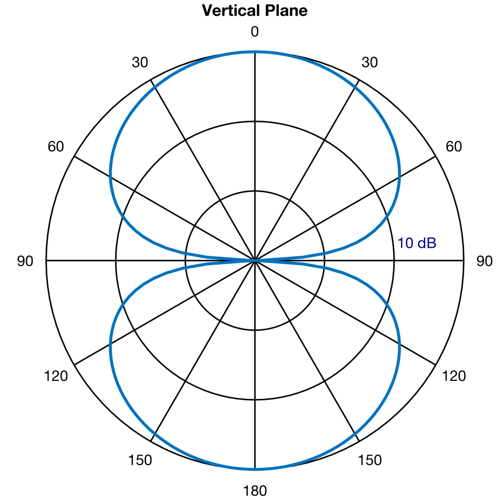
Eigenbeam Y_1^0

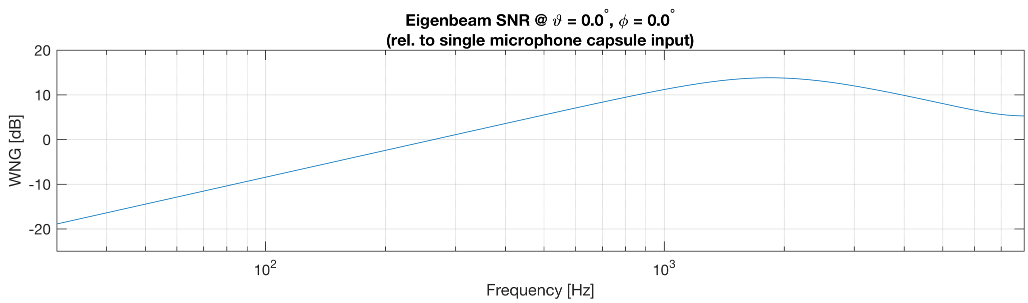


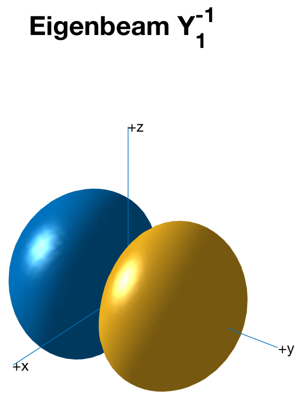
Horizontal Plane

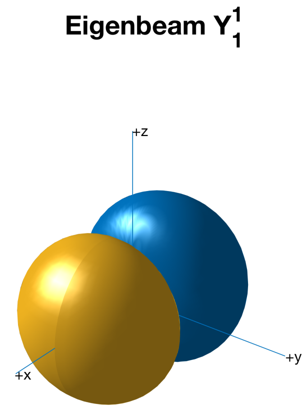
N/A

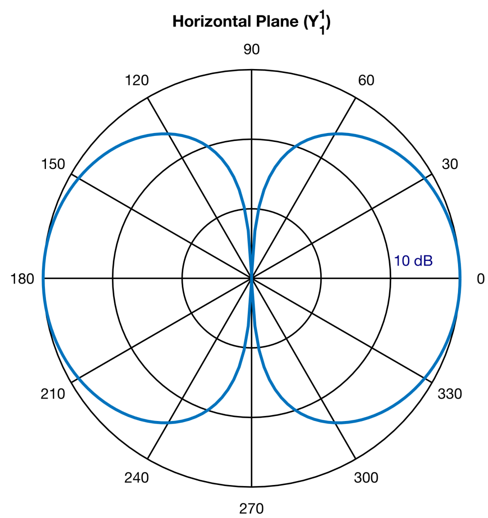
Vertical Plane

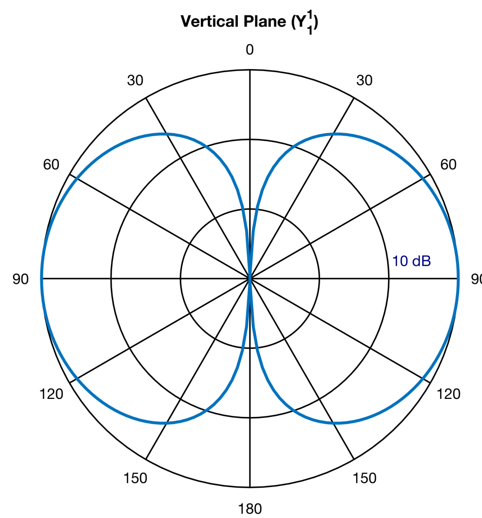


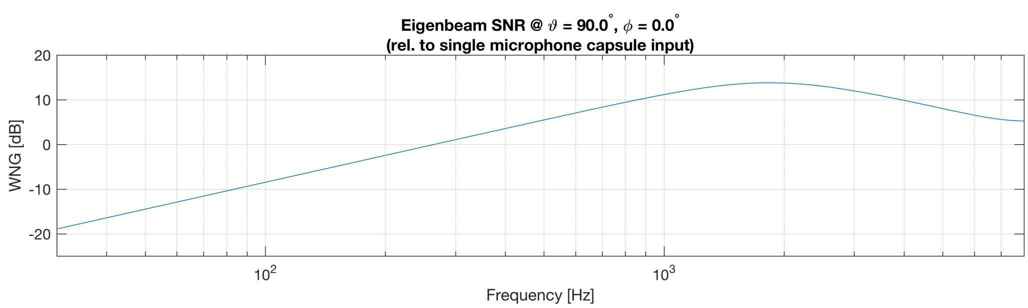



Eigenbeam Y_1^{-1}
+z
+x
+y

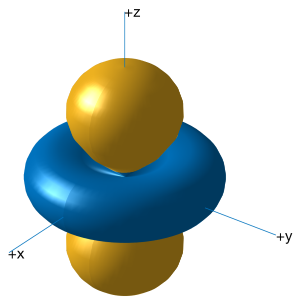

Eigenbeam Y_1^{1}
+z
+x
+y

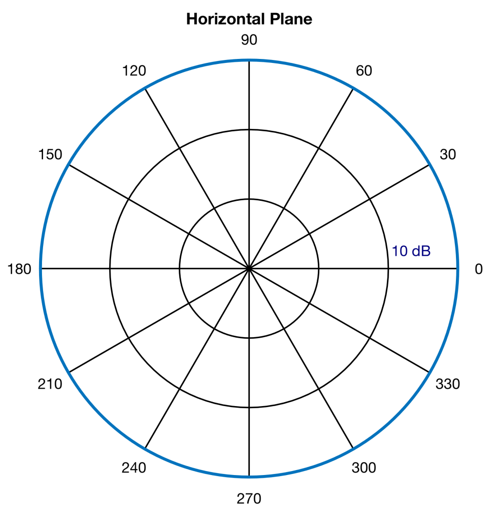

Horizontal Plane (Y_1^1)
90
120
60
150
30
180
10 dB
0
210
330
240
300
270

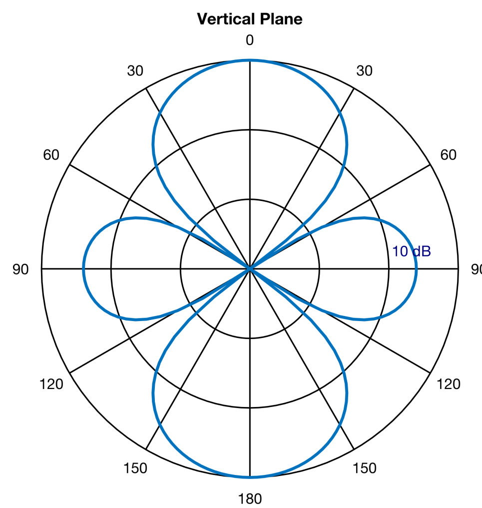

Vertical Plane (Y_1^1)
0
30
30
60
60
90
10 dB
90
120
120
150
150
180

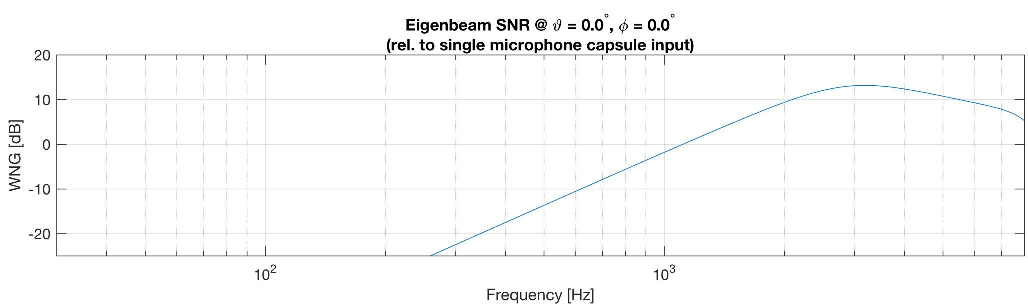

Eigenbeam SNR @ ϑ = 90.0°, ϕ = 0.0°
(rel. to single microphone capsule input)
WNG [dB]
20
10
0
-10
-20
10^2
10^3
Frequency [Hz]

Eigenbeam Y_2^0

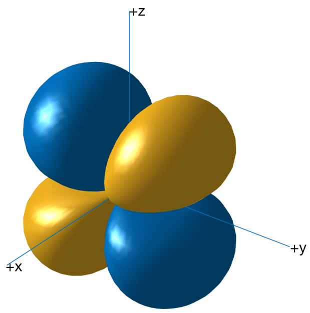




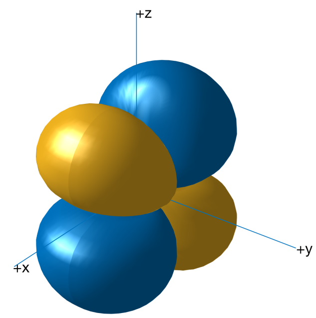




Eigenbeam Y_2^{-1}



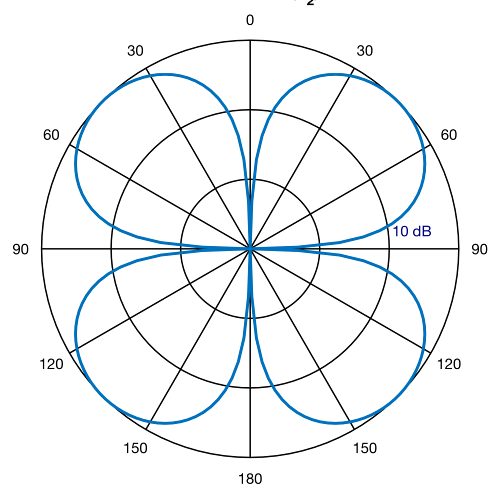
Eigenbeam Y_2^{1}



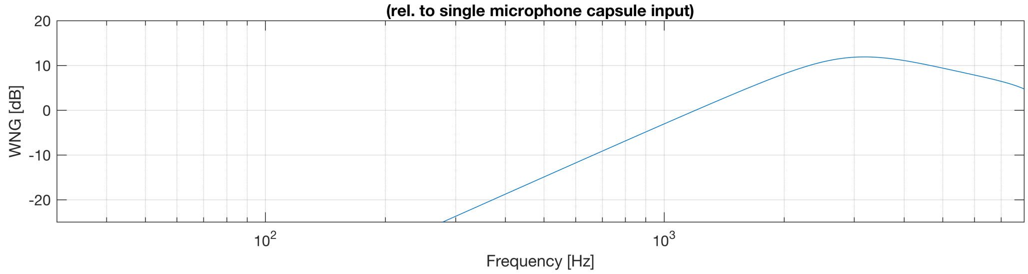
Horizontal Plane (Y_2^1)

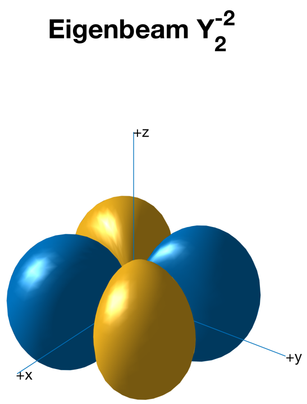
N/A

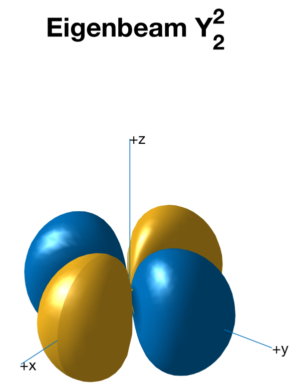
Vertical Plane (Y_2^1)

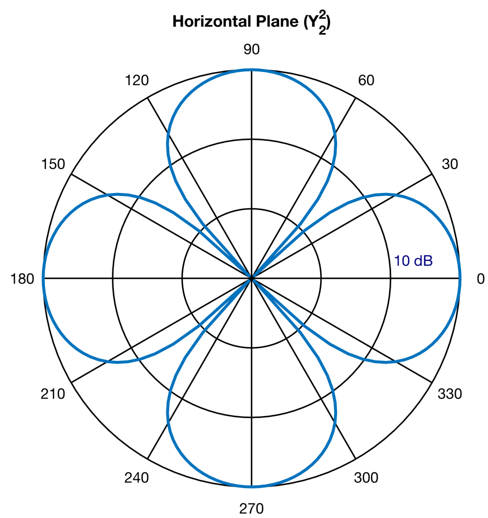


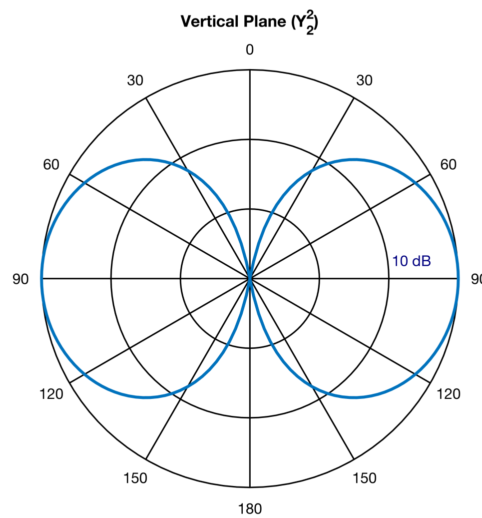
Eigenbeam SNR @ ϑ = 45.0°, ϕ = 0.0°
(rel. to single microphone capsule input)

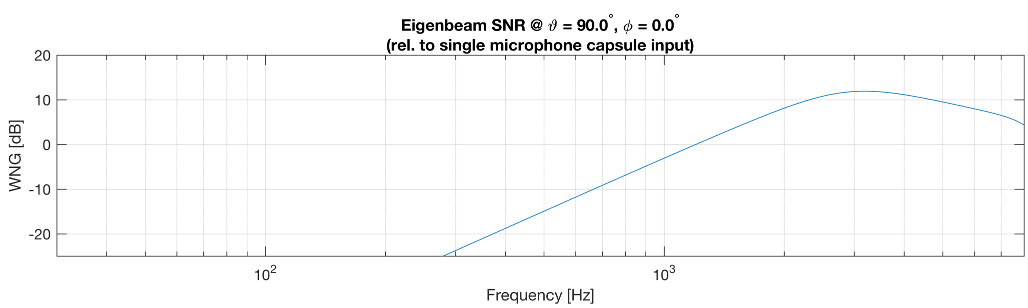



Eigenbeam Y_2^{-2}
+z
+x
+y

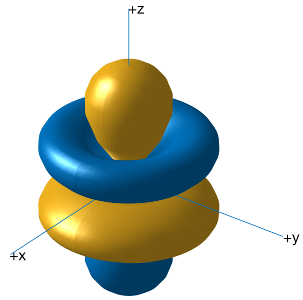

Eigenbeam Y_2^{2}
+z
+x
+y


Horizontal Plane (Y_2^2)
90
120
60
150
30
180
0
10 dB
210
330
240
300
270


Vertical Plane (Y_2^2)
0
30
30
60
60
90
90
10 dB
120
120
150
150
180


Eigenbeam SNR @ ϑ = 90.0°, ϕ = 0.0°
(rel. to single microphone capsule input)
WNG [dB]
20
10
0
-10
-20
10^2
10^3
Frequency [Hz]

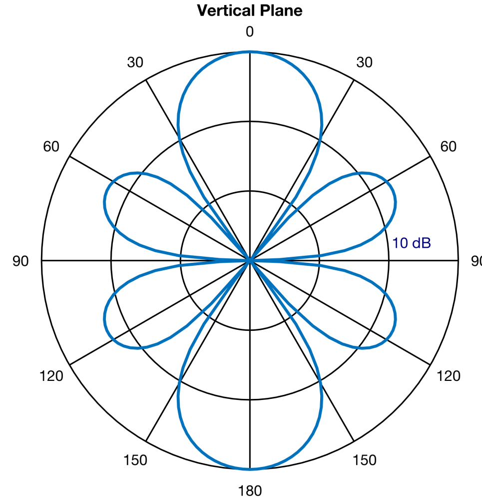
Eigenbeam Y_3^0

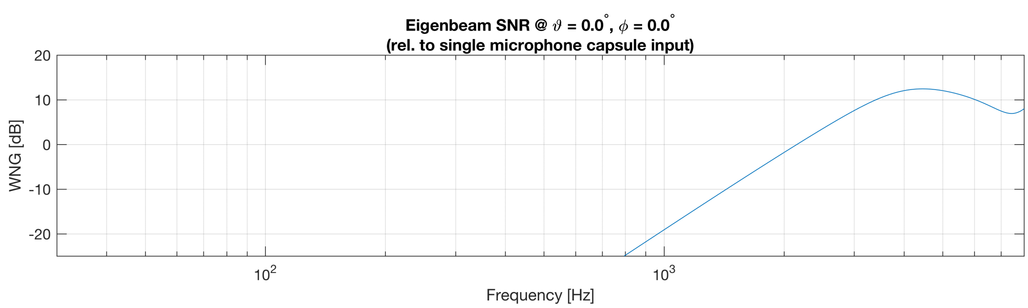


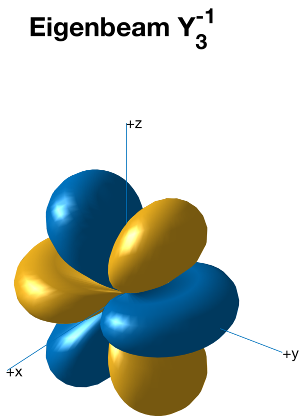
Horizontal Plane

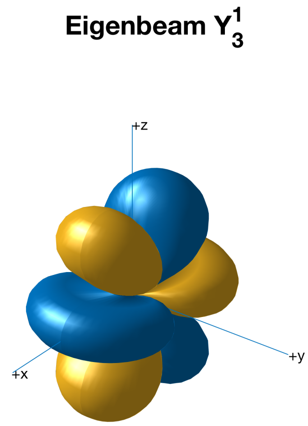
N/A

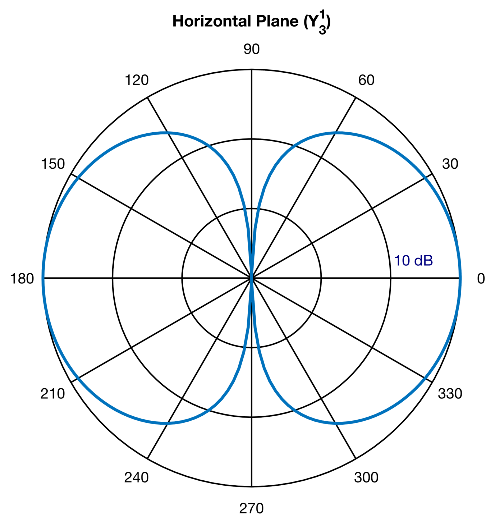
Vertical Plane

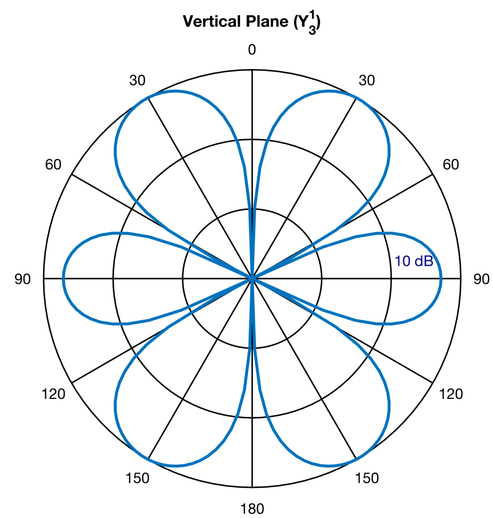


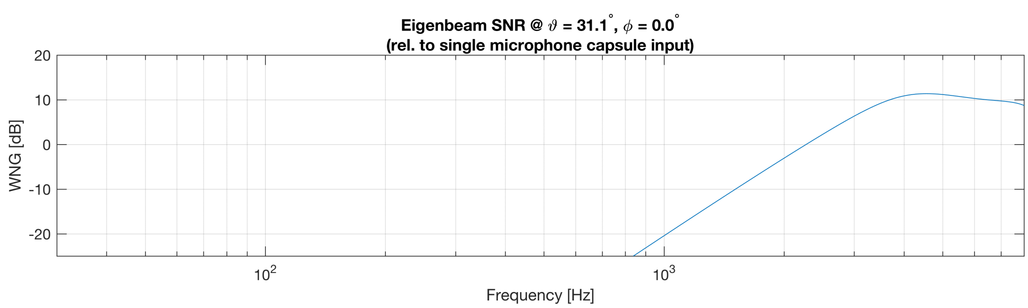



Eigenbeam Y_3^{-1}
+z
+x
+y

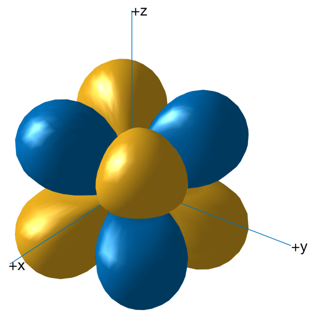

Eigenbeam Y_3^{1}
+z
+x
+y


Horizontal Plane (Y_3^1)
90
120
60
150
30
180
0
10 dB
210
330
240
300
270

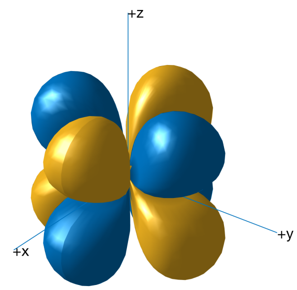

Vertical Plane (Y_3^1)
0
30
30
60
60
90
90
10 dB
120
120
150
150
180


Eigenbeam SNR @ ϑ = 31.1°, ϕ = 0.0°
(rel. to single microphone capsule input)
WNG [dB]
20
10
0
-10
-20
10^2
10^3
Frequency [Hz]

Eigenbeam Y_3^{-2}



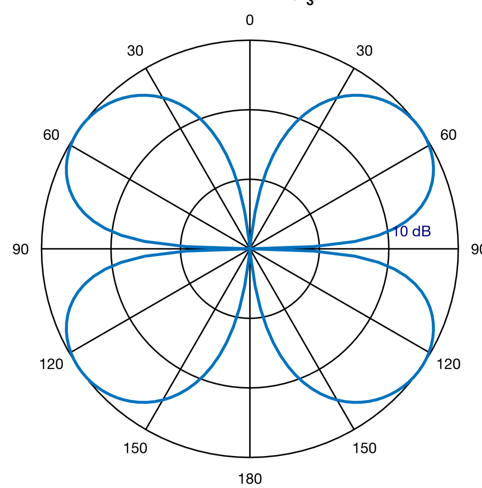
Eigenbeam Y_3^{2}

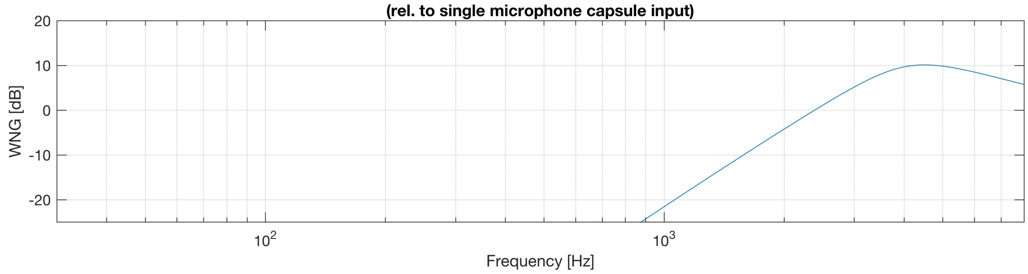


Horizontal Plane (Y_3^2)

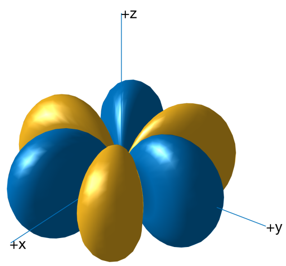
N/A

Vertical Plane (Y_3^2)

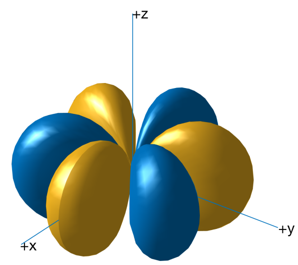


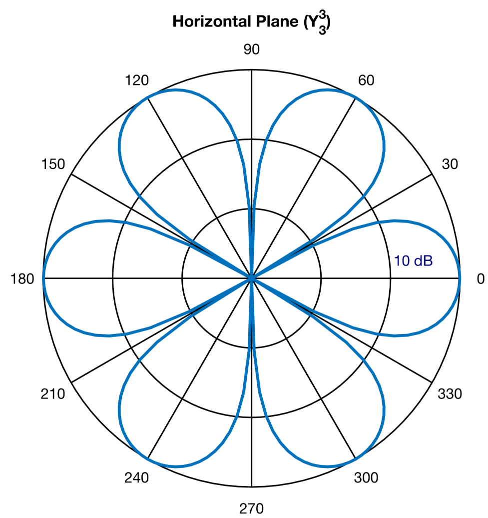


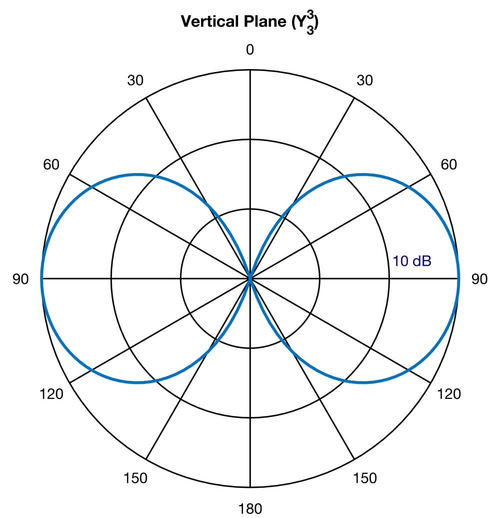
Eigenbeam Y_3^{-3}

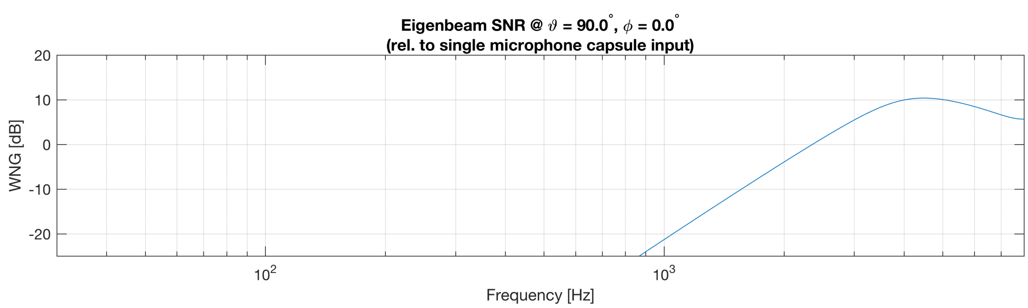


Eigenbeam Y_3^3

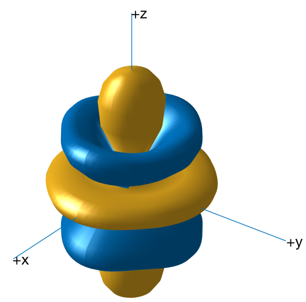


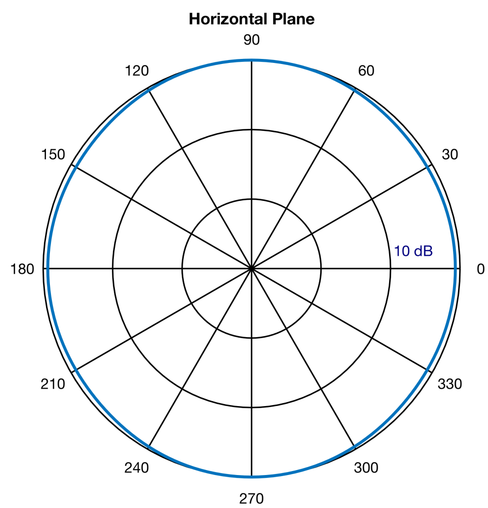


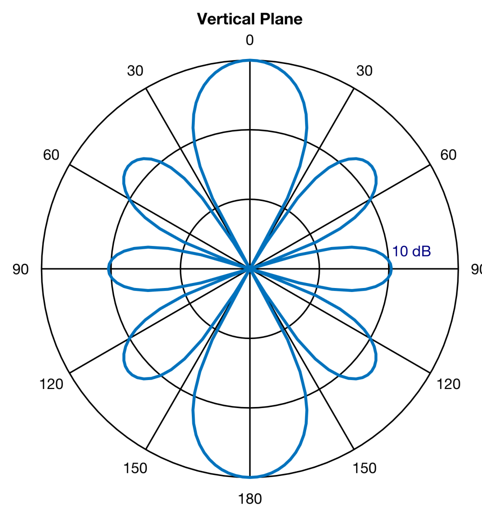


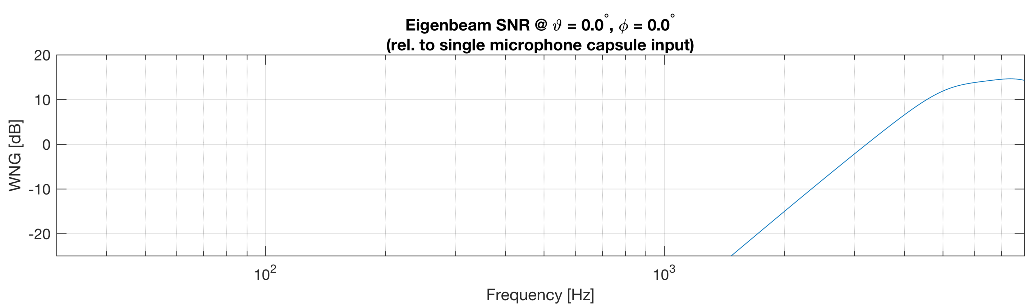


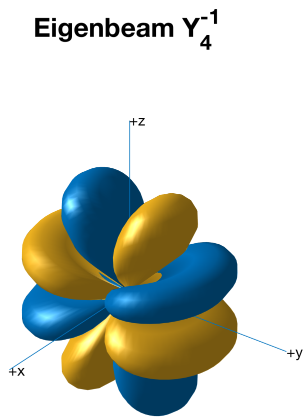
Eigenbeam Y_4^0

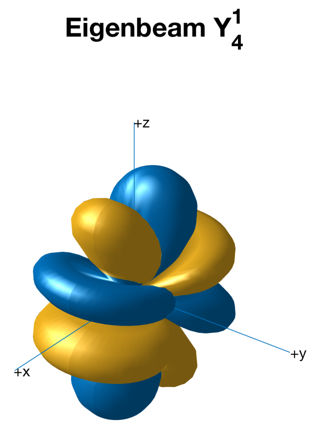






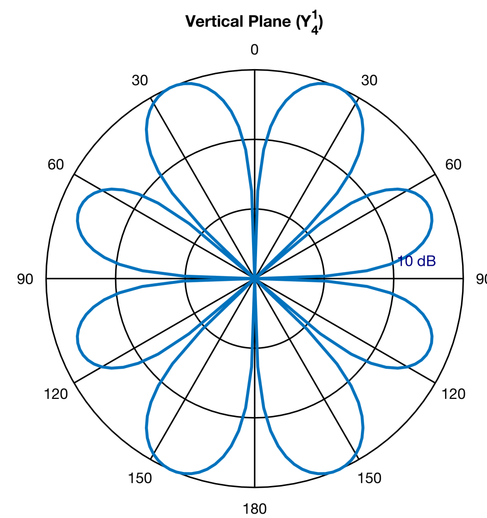


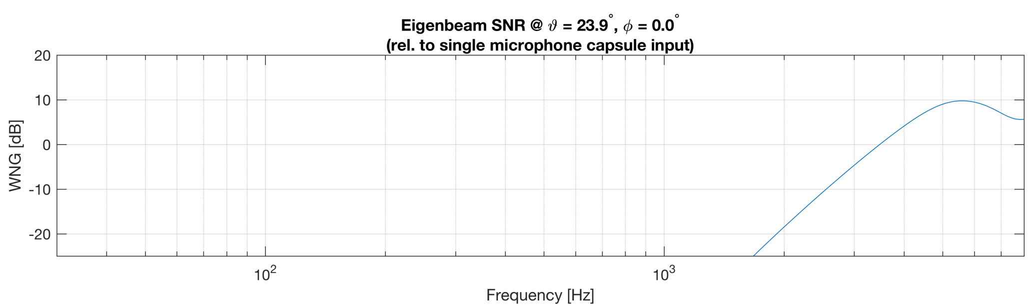


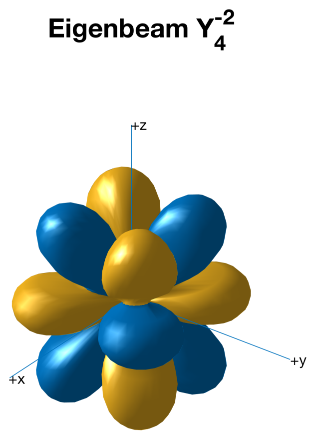


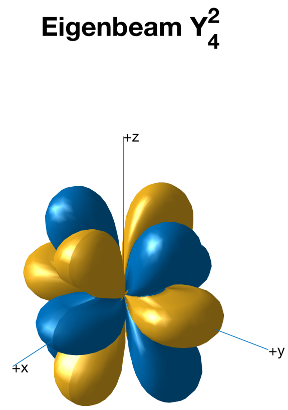
Horizontal Plane (Y_4^1)

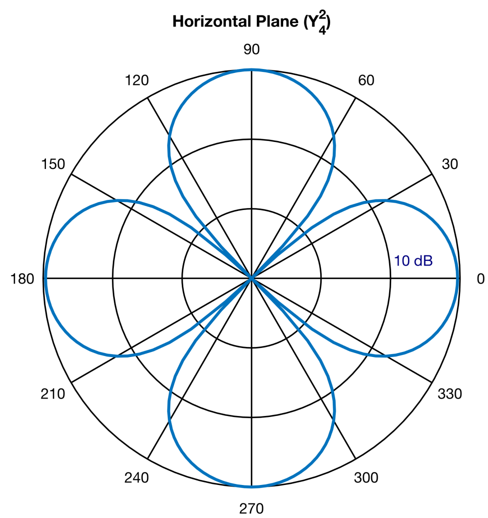
N/A

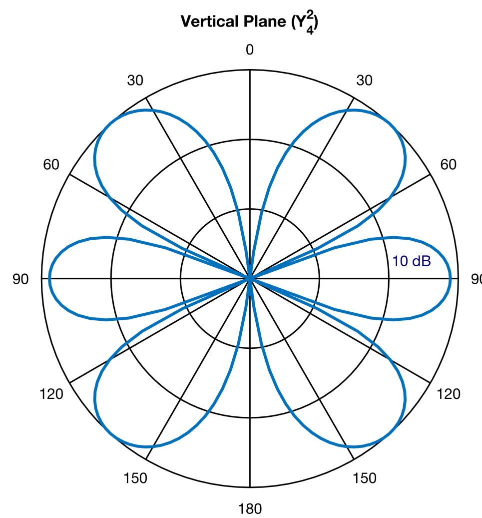


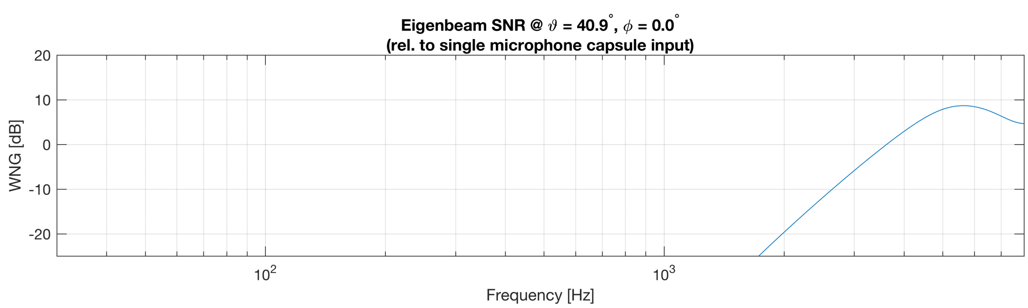



Eigenbeam Y_4^{-2}
+z
+x
+y

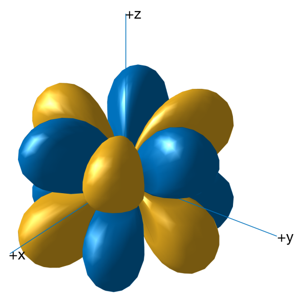

Eigenbeam Y_4^{2}
+z
+x
+y


Horizontal Plane (Y_4^2)
90
120
60
150
30
180
0
10 dB
210
330
240
300
270

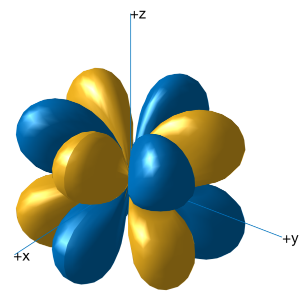

Vertical Plane (Y_4^2)
0
30
30
60
60
90
90
10 dB
120
120
150
150
180


Eigenbeam SNR @ ϑ = 40.9°, ϕ = 0.0°
(rel. to single microphone capsule input)
WNG [dB]
20
10
0
-10
-20
10^2
10^3
Frequency [Hz]

Eigenbeam Y_4^{-3}

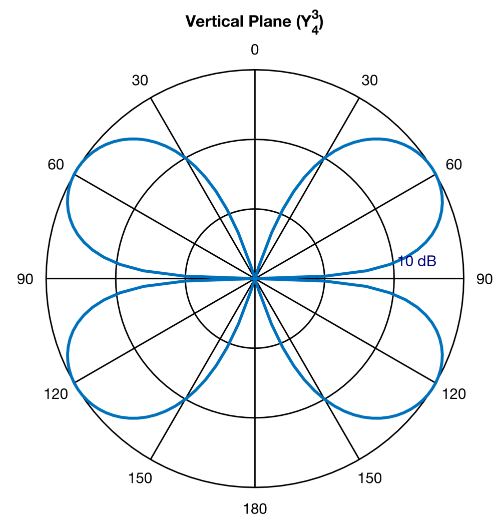


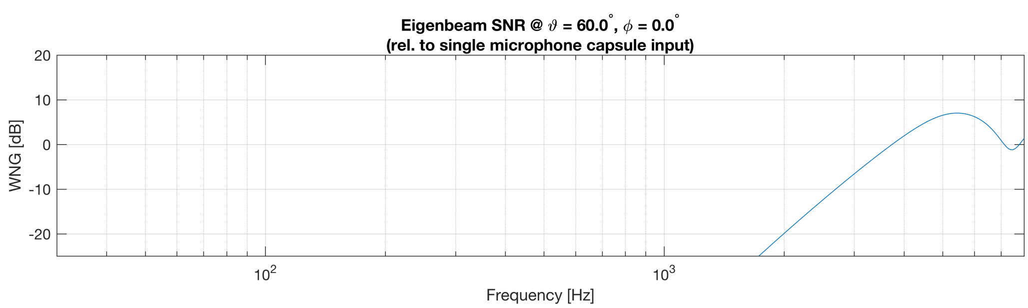
Eigenbeam Y_4^{3}



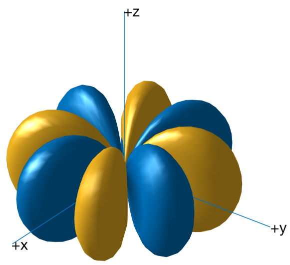
Horizontal Plane (Y_4^3)

N/A





Eigenbeam Y_4^{-4}



Eigenbeam Y_4^{4}