

<https://www.lesonbinaural.fr>

**GRATUIT !**



# IEM Plug-in Suite

---

La suite de plug-ins IEM est une suite de plug-ins audio gratuite et à code source ouvert, comprenant des plug-ins Ambisonic jusqu'au 7ème ordre créés par le personnel et les étudiants de l' Institut de musique électronique et d'acoustique.

**v1.8.1 (11/11/2018) VST2**

<https://plugins.iem.at/>

**Bernard Lagnel  
Décembre 2018**

# AES Milan 2018

## Atelier W31

Samedi 26 mai de 11h15 à 12h15 (Lobby)

### W31 - Présentation de la suite de plug-ins IEM

Présentateur :

*Daniel Rudrich*, Institut de musique électronique et d'acoustique Graz - Graz, Autriche

La "suite plug-in IEM" est une suite de plug-ins audio libre et à code source ouvert créée par le personnel et les étudiants d'IEM. Il comporte des plug-ins Ambisonic 3D d'ordre supérieur jusqu'au 7ème ordre, comprenant des encodeurs, des décodeurs, un visualiseur surround, des processeurs dynamiques et des effets de retard et de réverbération, qui permettent également d'auraliser les sources directionnelles. De plus, la suite de plug-ins inclut un outil permettant de concevoir des décodeurs Ambisonic pour des dispositions arbitraires de haut-parleurs utilisant l'approche AllRAD et l'imaginaire downmix. Toute la mise en œuvre est axée sur la recherche, et la suite est conçue pour se développer continuellement en fonction des développements récents, par exemple des améliorations du rendu binaural Ambisonic. L'atelier présente les plug-ins et présente de manière pratique les applications de base et le flux de travail dans Reaper.

Dernière version: **v1.8.1** , 11 novembre 2018 , **VST2**



macOS 10.9+

Windows 64 bits

Windows 32 bits

Linux amd64

#### NOTE: Briser les changements!

La version v1.3.0 a introduit des modifications radicales! Les projets utilisant l'une des versions précédentes risquent de ne pas fonctionner comme prévu.

Pour plus d'informations, voir l' [article Breaking changes](#) et / ou [ChangeLog](#) .

Traduction GOOGLE

<https://plugins.iem.at/>

# Descriptions de plug-ins

---

Vous trouverez ici une liste des plug-ins disponibles par ordre alphabétique et leur description.

01 **AllRADecoder**

02 **BinauralDecoder**

03 **Convertisseur de coordonnées**

04 **DirectionalCompressor**

05 **DirectivitéShaper**

06 **Compensateur de distance**

07 **DualDelay**

08 **EnergyVisualizer**

09 **FdnReverb**

10 **MatrixMultiplier**

11 **MultiEncoder**

12 **MultiEQ**

13 **OmniCompressor**

14 **ProbeDecoder**

15 **RoomEncoder**

16 **SceneRotator**

17 **SimpleDecoder**

18 **StereoEncoder**

19 **Boîte à outils**

# AllRA Decoder

1

Auteur Daniel Rudrich et Franz Zotter Type Décodeur Contribution Signaux ambisoniques Sortie Signaux audio

Avec le décodeur AllRA, vous pouvez concevoir un décodeur Ambisonic pour une disposition arbitraire des haut-parleurs à l'aide de l'approche AllRAD. Plus d'informations peuvent être trouvées dans le [Guide AllRA Decoder](#).

**IN** SN3D Auto (0th)

**OUT** 12 (bus too small)

**Loudspeaker Layout**

ID	Azimuth	Elevation	Radius	Channel	Imaginary	Gain	Noise	Remove
1	0°	0°	1	1	■	1	Noise	Remove
2	45°	0°	1	2	■	1	Noise	Remove
3	90°	0°	1	3	■	1	Noise	Remove
4	135°	0°	1	4	■	1	Noise	Remove
5	180°	0°	1	5	■	1	Noise	Remove
6	-135°	0°	1	6	■	1	Noise	Remove
7	-90°	0°	1	7	■	1	Noise	Remove
8	-45°	0°	1	8	■	1	Noise	Remove
9	22.5°	40°	1	9	■	1	Noise	Remove
10	142.5°	40°	1	10	■	1	Noise	Remove
11	-97.5°	40°	1	11	■	1	Noise	Remove

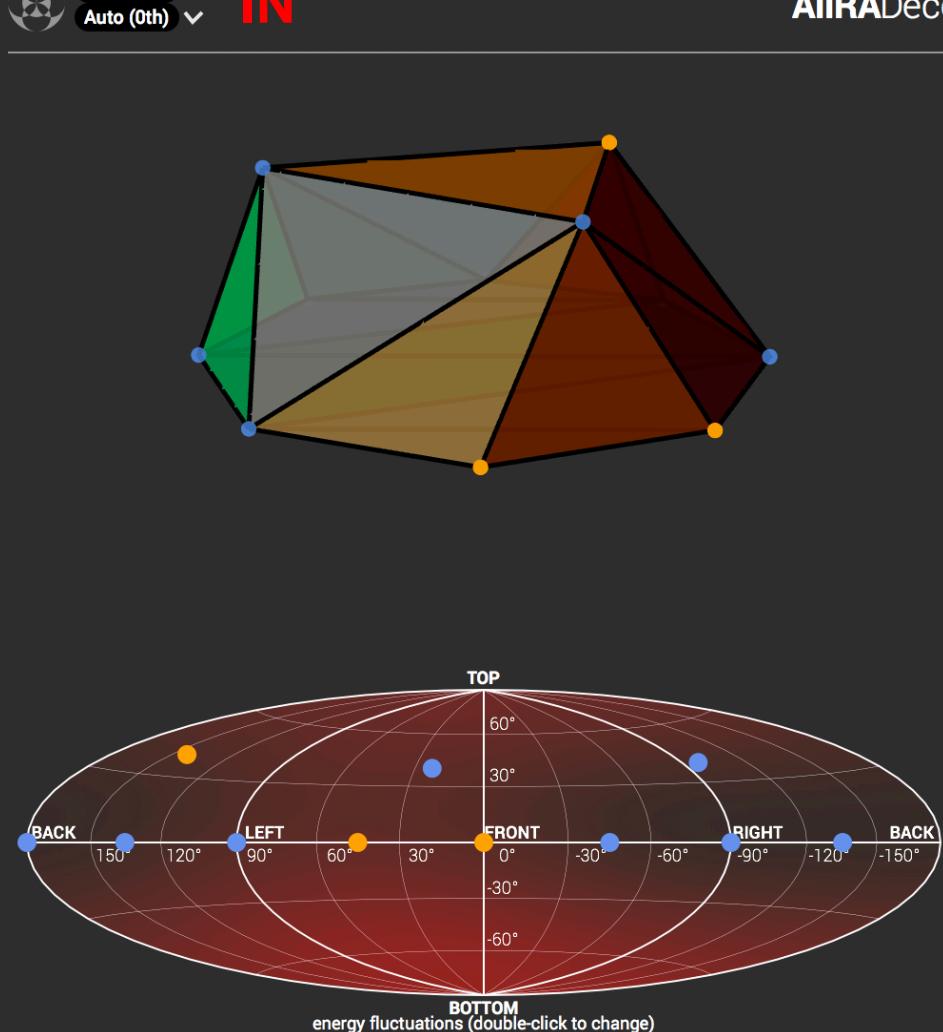
**ADD LOUDSPEAKER** **ROTATE** **UNDO** **REDO** **IMPORT**

**Improper layout**  
ERROR 4: Point of origin is not within the convex hull. Try adding imaginary loudspeakers.

**Calculate Decoder**  
Decoder Order 3rd  
**CALCULATE DECODER**

**Export Decoder/Layout**  
Export Decoder  
Export Layout  
**EXPORT**

iem OSC v0.7.3



# Guide AllRADecoder

Ce guide explique comment créer un décodeur pour n'importe quelle disposition de haut-parleur souhaitée à l'aide de l'approche AllRAD.

- 01 Informations générales
- 02 Comment commencer
- 03 Ajout de haut-parleurs imaginaires
- 04 Choisir une commande Ambisonic appropriée
- 05 Carte de distribution d'énergie et de largeur de source d'énergie renouvelable
- 06 Exporter les mises en page et les décodeurs
- 07 Caractéristiques futures

## Informations générales

Ce plug-in vous permet de créer un décodeur Ambisonic pour n'importe quelle disposition de haut-parleur et ordres Ambisonic souhaités jusqu'à la 7ème. Le décodeur résultant est automatiquement appliqué au flux audio Ambisonic entrant afin qu'il puisse être directement inséré dans une piste de décodeur de votre projet audio. Vous pouvez également l'utiliser comme application autonome pour concevoir un décodeur et l'écrire dans un fichier de configuration JSON afin de l'utiliser dans d'autres plug-ins (par exemple, **Simple Decoder**).

Une description théorique de l'approche AllRAD est disponible [ici](#).

## Comment commencer

La première chose à faire est d'insérer les coordonnées du haut-parleur (azimut et élévation) et les numéros de canal en cliquant sur "ajouter un haut-parleur". Les champs 'rayon' et 'gain' n'affecteront que les haut-parleurs imaginaires (voir ci-dessous). Le haut-parleur situé au même niveau que la tête de l'auditeur doit être spécifié par un angle d'élévation de 0 °.

Vous pouvez cliquer sur le `Noise` bouton qui enverra un court signal de bruit au canal correspondant. De cette façon, vous pouvez vérifier si votre routage est correct.

Les haut-parleurs apparaîtront à la fois dans la visualisation 3D (en haut à gauche) et dans la carte de volume (en bas à gauche). Le haut-parleur sélectionné est visualisé en vert et les imaginaires en orange, les autres en bleu. Vous pouvez faire pivoter la vue 3D en le faisant glisser et effectuer un zoom avant / arrière à l'aide de la molette de la souris. Contrairement aux haut-parleurs imaginaires, les directions physiques des haut-parleurs sont toujours visualisées avec un rayon de un.

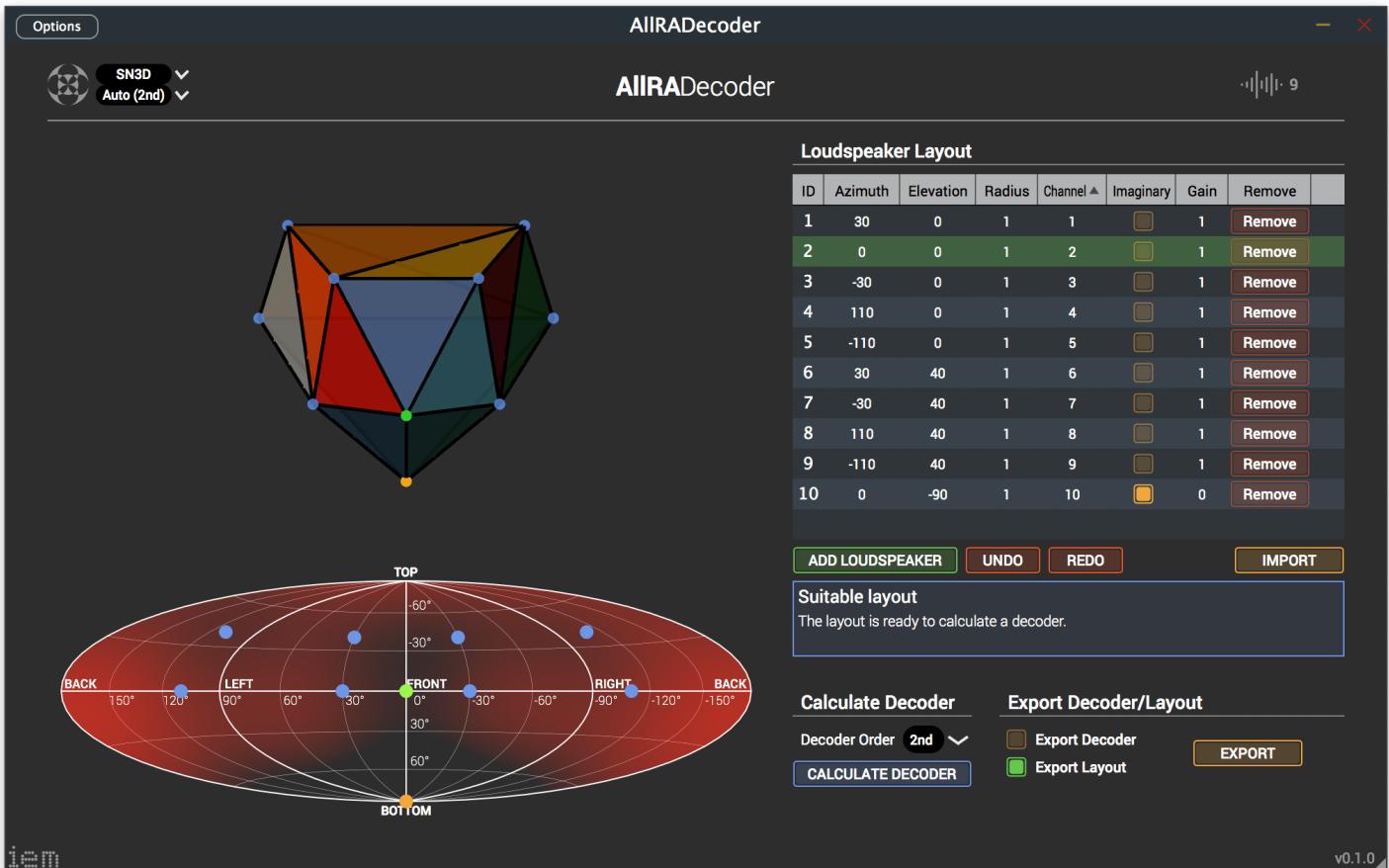
Vous pouvez faire pivoter toute la mise en page autour de l'axe z en cliquant sur le bouton de rotation. Cela ajoutera une valeur constante à tous les angles d'azimut.

Après avoir inséré suffisamment de haut-parleurs, la disposition sera immédiatement triangulée et affichée dans la visualisation 3D. Tout problème lié à votre disposition apparaît dans la zone de message située sous la liste des haut-parleurs.

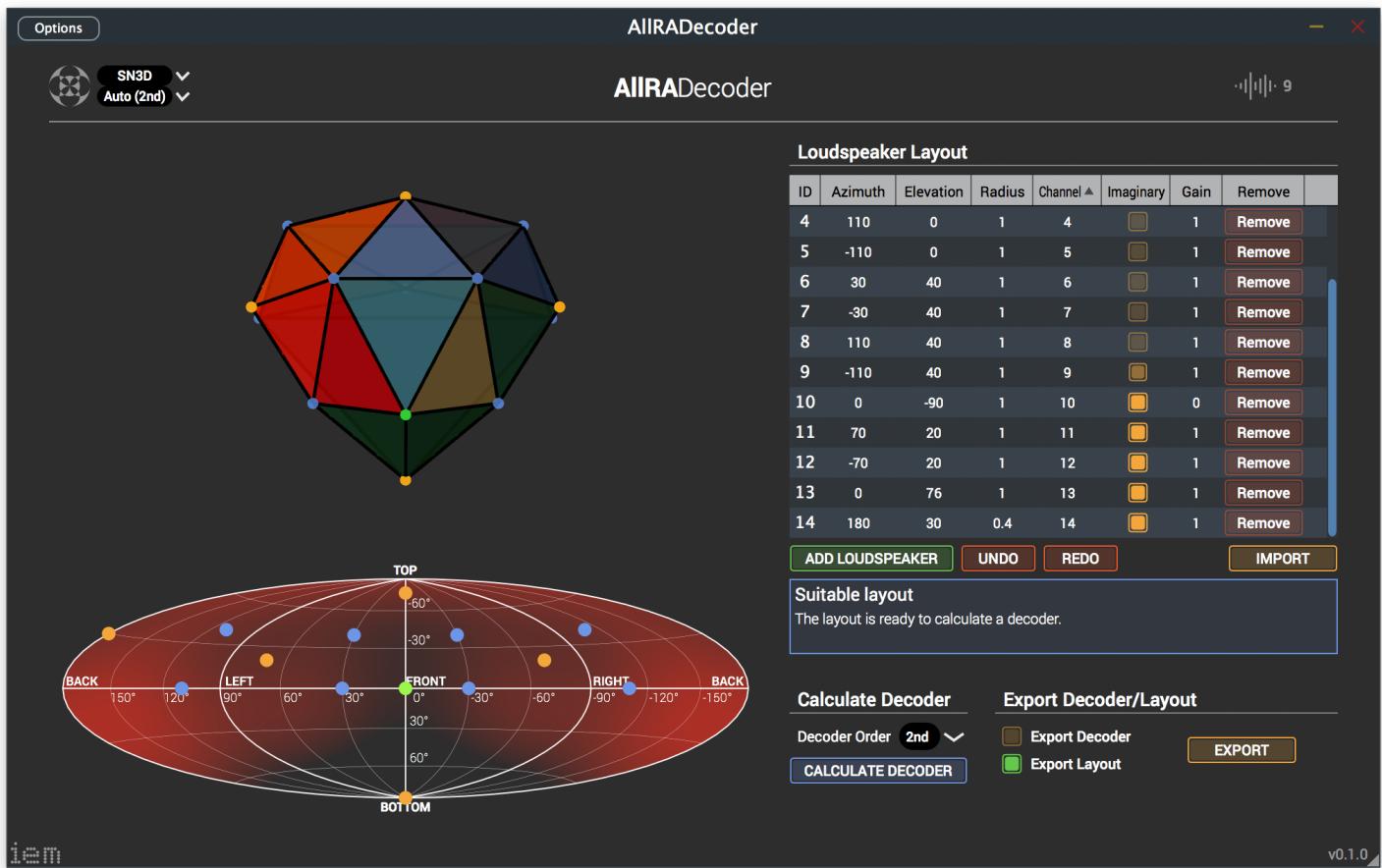
## Ajout de haut-parleurs imaginaires

Parfois, une disposition de haut-parleur ne peut pas être triangulée de manière non ambiguë. Avec l'approche AllRAD, des haut-parleurs imaginaires peuvent être ajoutés pour contourner ces cas.

Par exemple, presque toutes les configurations sans haut-parleurs dans l'hémisphère inférieur ont un plan de sol ambigu. Insérer un haut-parleur imaginaire juste en dessous de l'auditeur empêchera ce cas. La disposition suivante décrit cet exemple. Vous pouvez également vouloir régler le gain de ce haut-parleur imaginaire spécifique sur zéro, sinon les sources de dessous seront reproduites par toutes les haut-parleurs au niveau du sol. Téléchargez le fichier de configuration pour cette mise en page ici: [`layoutWithOneImaginaryLoudspeaker.json`](#)



Cependant, dans cette configuration, il existe plusieurs autres plans, qui pourraient être divisés en triangles d'une manière différente: le plan supérieur, les côtés gauche et droit, ainsi que le plan situé juste derrière l'auditeur. De plus, le plan supérieur ne peut en aucun cas être triangulé symétriquement. Il est conseillé d'ajouter ici des haut-parleurs imaginaires. Dans la présentation suivante, quatre haut-parleurs imaginaires supplémentaires ont été ajoutés. La configuration peut être téléchargée ici: [layoutWithAdditionalImaginaryLoudspeakers.json](#)



Le gain reste inchangé, car on ne veut pas de spots silencieux dans ces zones. Cependant, pour le haut-parleur n ° 14, le rayon devait être modifié. Sinon, il y aurait eu une connexion entre deux haut-parleurs imaginaires, ce qui n'est pas pris en charge par l'approche AllRAD.

## Choisir une commande Ambisonic appropriée

Le choix de l'ordre Ambisonic est un compromis entre une largeur de source constante et un panoramique lisse (ordre inférieur), ainsi qu'une localisation précise et un panoramique plus irrégulier se rapprochant davantage du VBAP (ordre élevé). Il est typique de choisir le 2ème ordre pour les dispositions de haut-parleur espacées de 60 °, le 3ème pour 45 ° et le 5ème ordre pour 30 °, respectivement. Le 7ème ordre est le plus élevé que vous puissiez choisir et convient aux dispositions espacées de 15 °.

## Carte de distribution d'énergie et de largeur de source d'énergie renouvelable

Quand un décodeur a été calculé avec succès, sa distribution d'énergie est indiquée dans la carte de sonie en bas à gauche (rouge). Les valeurs maximales sont visualisées en rouge (jusqu'à +3 dB) tandis que les zones de faible énergie (jusqu'à -3 dB) sont transparentes. Le choix de l'ordre Ambisonic influera sur la distribution de l'énergie.

Lorsque vous double-cliquez sur la carte, la largeur de la source acos-rE sera affichée (verte). Plus la couleur est brillante, plus l'image d'une source qui est orientée dans cette direction est large. En général, les sources situées à la position du haut-parleur seront plus petites (moins vertes) que les sources situées entre les haut-parleurs, car plusieurs haut-parleurs essaieront de reproduire le signal.

## Exporter les mises en page et les décodeurs

Vous pouvez exporter votre mise en page vers un fichier JSON en cliquant sur le bouton "Exporter". En outre, vous pouvez cocher la case "Exporter le décodeur" pour écrire également la matrice de décodeur calculée (n'oubliez pas de la calculer en premier) dans le fichier de configuration, qui peut par exemple être utilisé avec le plug-in **Simple Decoder**.

## Caractéristiques futures

Dans les mises en œuvre futures, il y aura plus d'analyse que la carte de volume (énergie), avec des cartes de l'analyse basée sur le vecteur rE de la cartographie directionnelle et de la largeur perçue. De plus, l' approche AllRAD 2 sera bientôt mise en œuvre. Il apporte des améliorations à une sonie invariante en panoramique sur une disposition généralement irrégulière, comme la disposition 4 + 5 + 0 de l'UIT BS.2051 avec un espacement dense des haut-parleurs uniquement à l'avant.

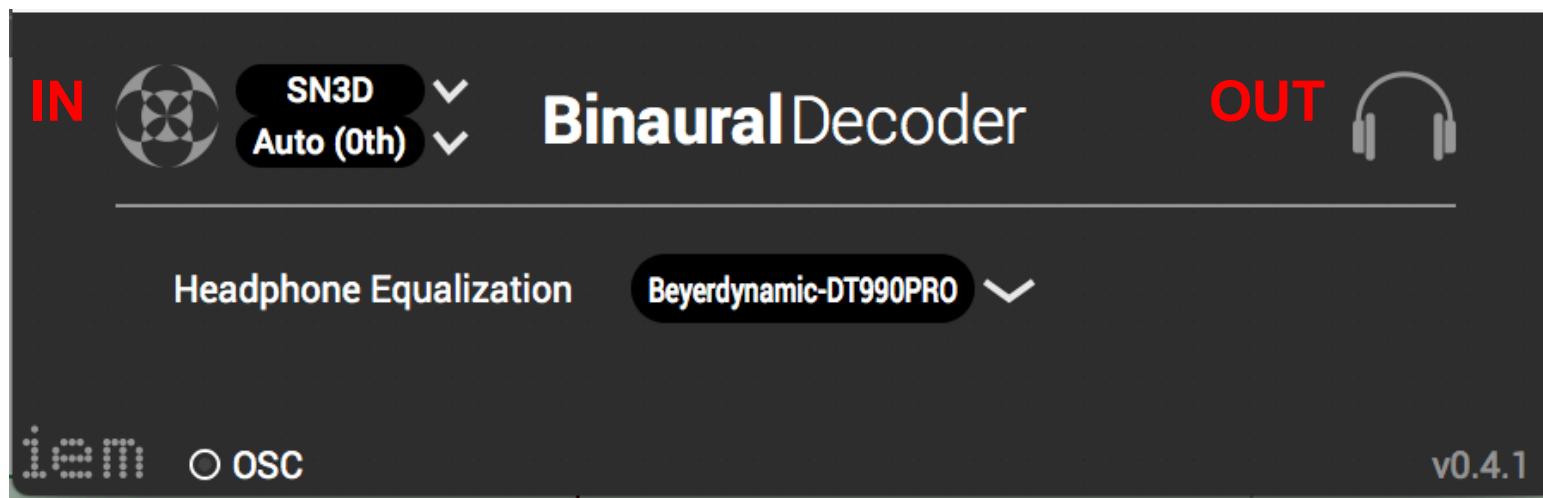
---

Plug-in Suite de IEM - Institut de musique électronique et d'acoustique

[IEM.at](#)

[Docs](#)

Site créé avec [Hugo](#) sur le thème du [kube](#).

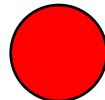


OFF
AKG-K141MK2
AKG-K240DF
AKG-K240MK2
AKG-K271MK2
AKG-K271STUDIO
AKG-K601
AKG-K701
AKG-K702
AKG-K1000-Closed
AKG-K1000-Open
AudioTechnica-ATH-M50
Beyerdynamic-DT250
Beyerdynamic-DT770PRO-250Ohms
Beyerdynamic-DT880
X Beyerdynamic-DT990PRO
Presonus-HD7
Sennheiser-HD430
Sennheiser-HD480
Sennheiser-HD560ovationII
Sennheiser-HD565ovation
Sennheiser-HD600
Sennheiser-HD650
SHURE-SRH940

## PLUG-IN KU 100 TH Köln

## HRTF NEUMANN KU 100

BinauralDecoder



Auteur Daniel Rudrich Type Décodeur binaural Contribution Signaux ambisoniques Sortie Signaux casque binaural

Le Binaural décodeur rend le signal d'entrée Ambisonic à un signal de casque binaural utilisant le *rendu Binaural des signaux Ambisonic par alignement temporel HRIR et une contrainte prolixité* approché par Markus Zaunschirm, Christian Schörkhuber et Robert Höldrich. Les HRTF utilisés proviennent de la tête factice Neumann KU 100. De plus, vous pouvez appliquer des égalisations de casque à partir des mesures de [Benjamin Bernschütz et. Al.](#).

Contrairement aux décodeurs binauraux classiques, le Binaural décodeur n'utilise pas des haut - parleurs virtuels, mais convertit les signaux Ambisonic directement aux signaux de casque binaural, avec l'aide de HRTFs pré-traitées. Ils sont traités de manière à ce que la réponse en fréquence initiale des HRTF soit maintenue le mieux possible, lorsqu'elle est utilisée avec Ambisonics.

▼		<a href="#">BinauralDecoder / Source / IRs / README.txt</a>			Voir le fichier @ f1f72516
1	1	Ces réponses impulsionales font partie de la suite de plug-ins IEM: <a href="https://plugins-iem.at">https://plugins-iem.at</a>			
2	2				
3		- Ils ont été créés à l'aide des mesures KU100 HRIR de Benjamin Bernschuetz [1] et de l'approche de rendu de Markus Zaunschirm, Christian Schoerkhuber et Robert Hoeldrich [2].			
	3	+ Ils ont été créés à l'aide des mesures KU100 HRIR présentées dans [1] et de l'approche de rendu proposée dans [2].			
4	4				
	5	+ [1] Bernschütz, Benjamin. "Une compilation sphérique HRIR / HRTF de champ éloigné de la Neumann KU 100", Actes de la 40e conférence annuelle italienne sur l'acoustique (AIA) et de la 39e conférence annuelle allemande sur l'acoustique (DAGA). 2013. <a href="http://audiogroup.web.th-koeln.de/ku100hrir.html">http://audiogroup.web.th-koeln.de/ku100hrir.html</a>			
5	6				
	7	+ [2] Schoerkhuber, Christian; Zaunschirm, Markus; Hoeldrich, Robert. "Rendu binaural des signaux ambisoniques via les moindres carrés de magnitude", Fortschritte der Akustik, DAGA, 2018			
6	8				
7		- [1] Bernschuetz, B. (2013. Compilation HRIR / HRTF de champ lointain sphérique du Neumann KU 100. AIA-DAGA 2013 Merano. <a href="http://audiogroup.web.th-koeln.de/ku100hrir.html">Http://audiogroup.web.th-koeln.de/ku100hrir.html</a>			
8		-			
9		- [2] Zaunschirm, M., Schoerkhuber, C., Hoeldrich, R. (à paraître). Rendu binaural des signaux Ambisonic par alignement temporel HRIR et contrainte de diffusion. (JASA)			

# Champ lointain sphérique | HRTF | compilation du Neumann KU100

## Licence

| CC BY-SA 3.0 |

## La description



Installation de la mesure dans la chambre anéchoïque de FH Köln.

Les jeux de données HRIR en trois dimensions ont été mesurés avec la tête fictive Neumann KU100. Un louspeaker à 3 voies actif (Genelec 8260A) a été utilisé comme source sonore avec une distance constante d'environ 3,25 m. Différentes positions de source apparentes ont été réalisées en faisant pivoter la tête fictive autour de deux axes à l'aide du système de mesure [VariSpear \[Bernschuetz2010\]](#). Les réponses impulsionales ont été capturées pour différentes configurations d'échantillonnage de la position de la source:

- plan horizontal avec une résolution de 1 °
- deux grilles de Lebedev sphériques différentes et équidistantes avec 2354 et 2702 points d'échantillonnage
- grille de Gauss 2 ° équiangulaire pleine sphère avec 16020 points d'échantillonnage

Pour plus de détails, voir le [site Web de FH Köln](#) ou le document correspondant [\[Bernschuetz2013\]](#).

## Des dossiers

```
impulse_responses / fhk_ku100_anechoic / HRIR_CIRC360RM.sofa  
impulse_responses / fhk_ku100_anechoic / HRIR_CIRC360.sofa  
impulse_responses / fhk_ku100_anechoic / HRIR_FULL2DEG.sofa
```

[\[Bernschuetz2013\]](#)

Bernschütz, B. (2013) "Compilation HRIR / HRTF de champ éloigné sphérique du Neumann KU 100", conférence annuelle allemande sur l'acoustique (DAGA)

[\[Bernschuetz2010\]](#)

Bernschütz, B., Pörschmann, C., Spors, S. et Weinzierl, S. (2010).



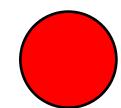
Compilation de HRIR sphérique en champ éloigné du Neumann KU100

### Audio Group Téléchargements

Administration et contact: [Johannes Mathias Arend](#)

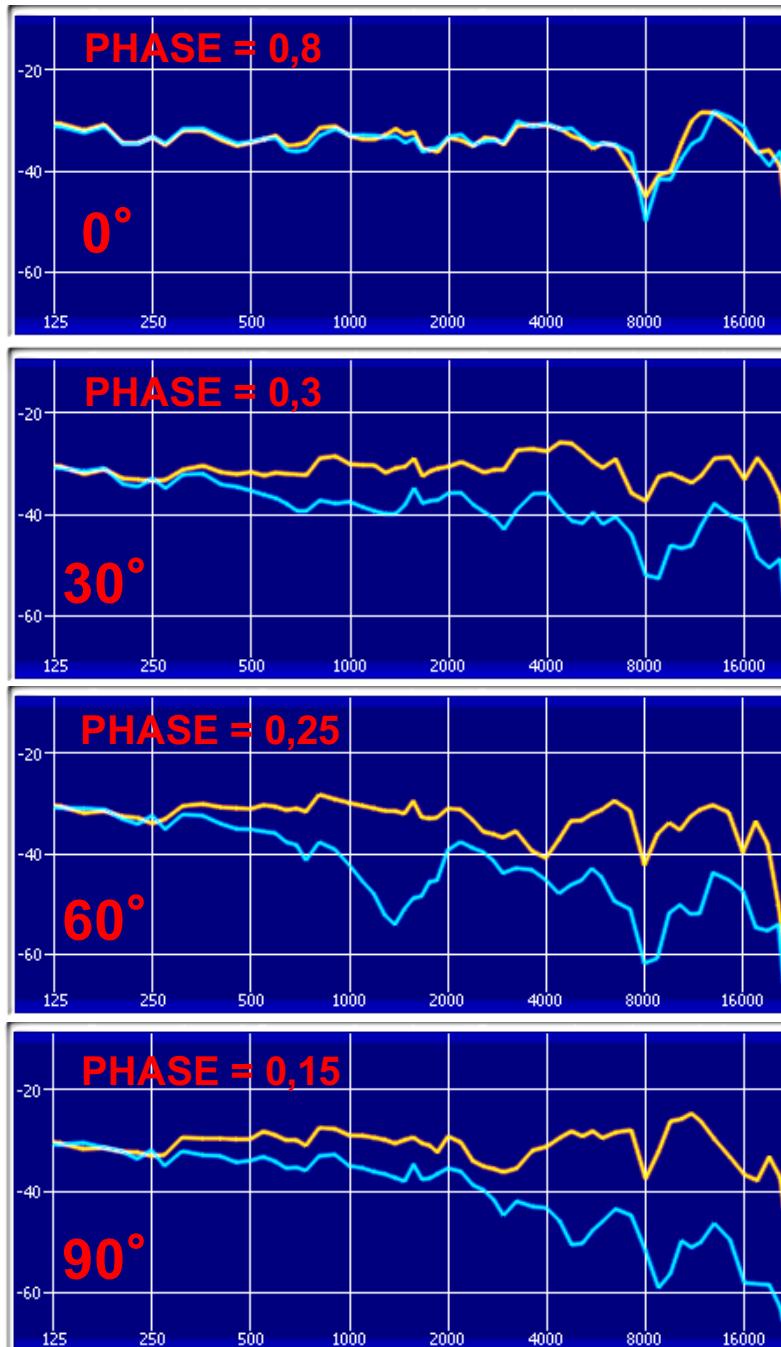
La description	Version	Date de dépôt	Type	Taille	Télécharger
<b>Document de conférence AIA / DAGA 2013, Meran (Italie)</b>	-	23.03.2013		1,6 Mo	
Description du type de <b>miro</b> / Manuel	V1.06, REV 1.3	20.09.2017		337 Ko	
Tutoriels / Exemples de types de données <b>miro</b>	V1.06	20.09.2017		10 Mo	
Fichier de définition de classe <b>miro</b> ( <i>obligatoire pour tous les jeux de données</i> )	V1.06	20.09.2017		37 Ko	
Filtres de compensation de casque <b>HPCF</b>	R15-1020	20.10.2015		1 Mo	
Filtre de compensation de champ diffus <b>DFC</b> (peut être utilisé de la même manière que le HPCF)	R13-0219	19.02.2013		6 Ko	
<b>KU100 HRIR Jeu de données Circulaire 360 ° ( Pied de microphone pivoté )</b>	R13-0225	25.02.2013		350 Ko	
<b>KU100 HRIR Dataset Circular 360 ° ( Montage à rotation complète)</b>	R13-0225	25.02.2013		350 Ko	
<b>KU100 HRIR Jeu de Données Sphérique - Lebedev 2354 Nodes</b>	R13-0225	25.02.2013		2,3 Mo	
<b>KU100 HRIR Jeu de Données Sphérique - Lebedev 2702 Nodes</b>	R13-0225	25.02.2013		2,6 Mo	
<b>KU100 HRIR Jeu de données sphérique - Gauss 2 ° Step size</b>	R14-0526	26.05.2014		15,2 Mo	
<b>KU100 HRIR Jeux de données SOFA</b> <b>SOFA - Format spatial pour l'acoustique</b> , grâce à Piotr Majdak de l'Institut autrichien de recherche sur l'acoustique ARI	-	18.11.2013		-	<a href="#">SOFA LINK</a>
<b>KU100 HRIR Jeu de données (sphérique - Gauss 2 °) en DAFF</b> <b>DAFF - Format de fichier audio directionnel</b> , merci à Jonas Stienen de l'Institut d'acoustique technique (ITA), RWTH Aachen	-	21.09.2016		-	<a href="#">DAFF LINK</a>
<b>Photos</b> (prises par Philipp Stade)	R12-1206	11.01.2013		7.8 MB	





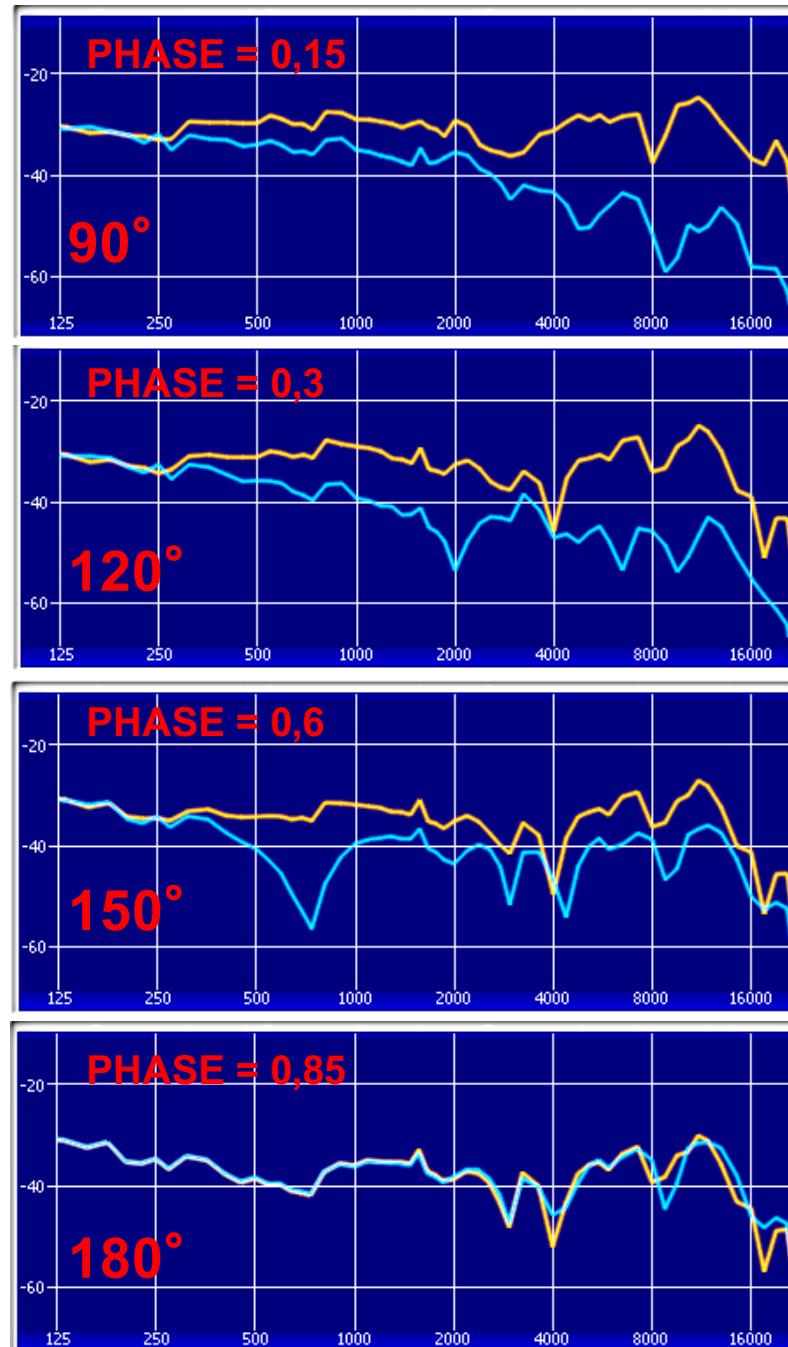
# PLUG-IN KU 100 TH Köln

— Oreille Ipsilatéral  
— Oreille Contralatéral



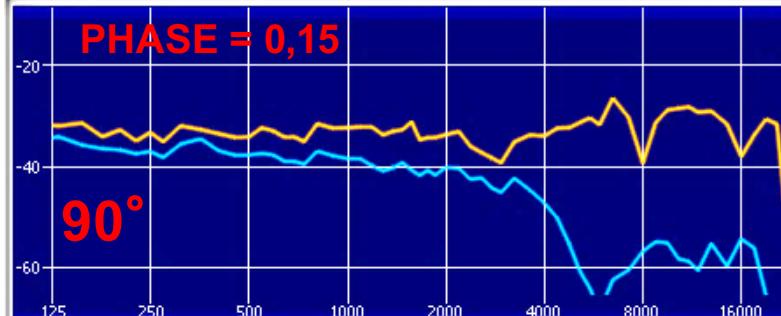
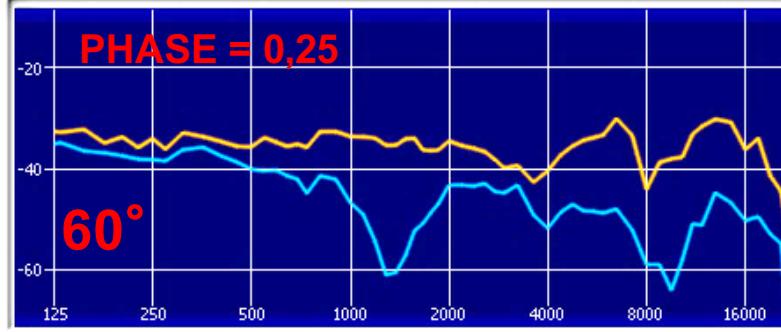
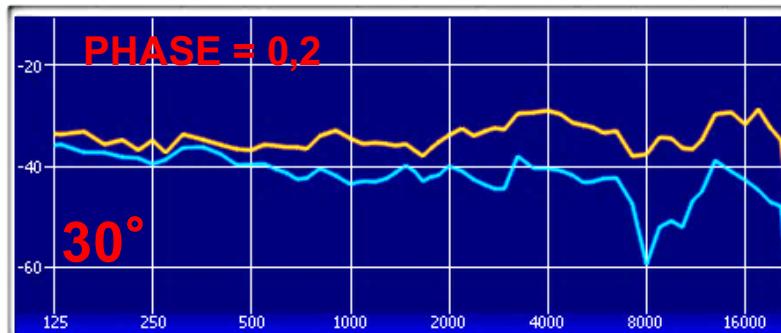
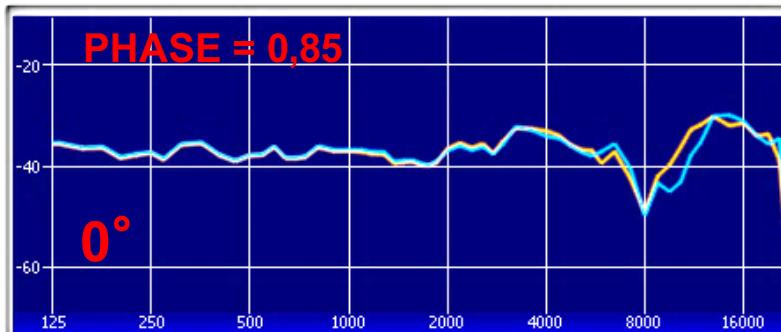
HRIR\_FULL2DEG.sofa

<http://audiogroup.web.th-koeln.de/ku100nfhrir.fnk/>  
<http://sofacoustics.org/data/database/fnk/>



# PLUG-IN KU 100 SADIE

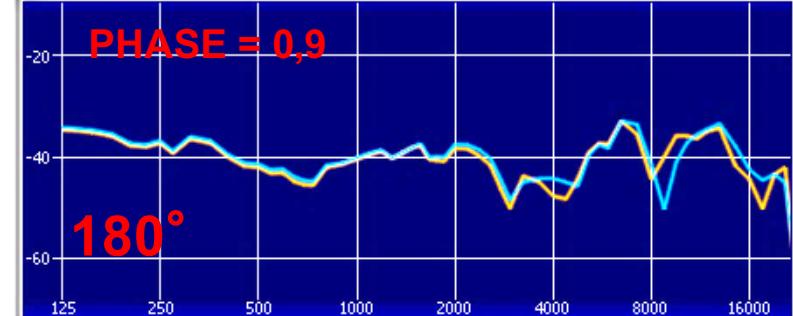
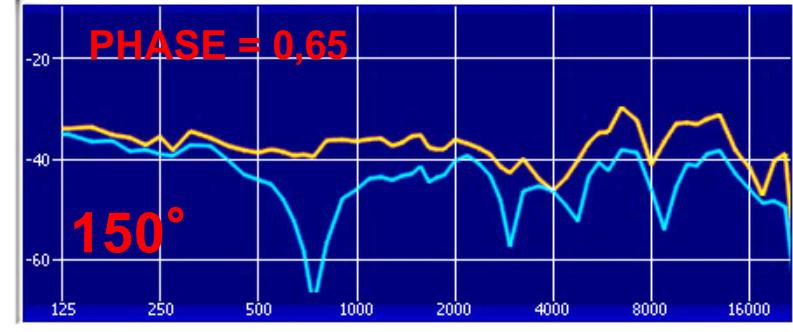
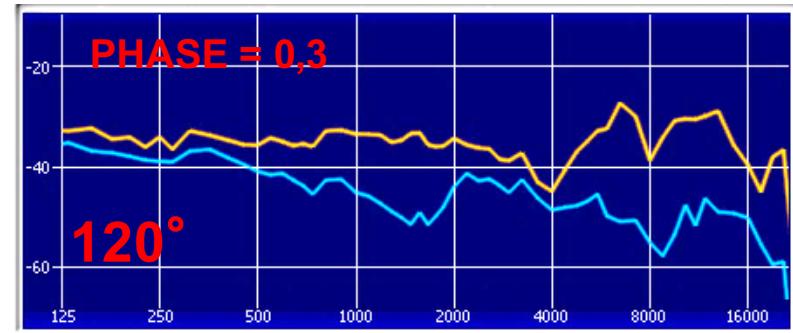
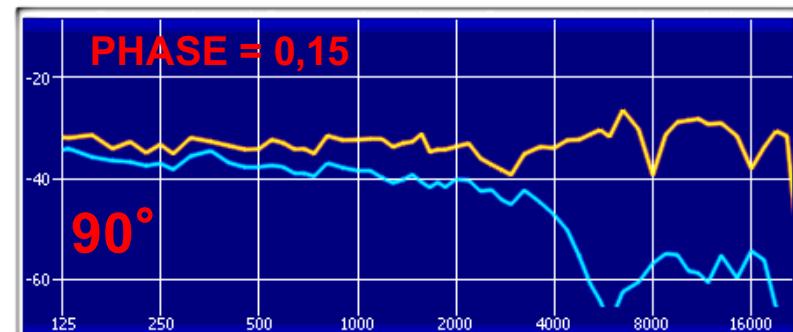
— Oreille Ipsilatéral  
— Oreille Contralatéral



Adopté par :  
Youtube 360  
 et Google VR

SADIE\_KU100\_DFC\_256\_order\_fir\_48000.sofa

<https://www.york.ac.uk/sadie-project/binaural.html>



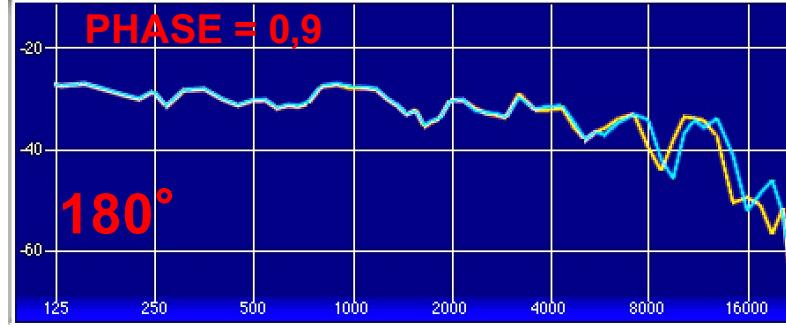
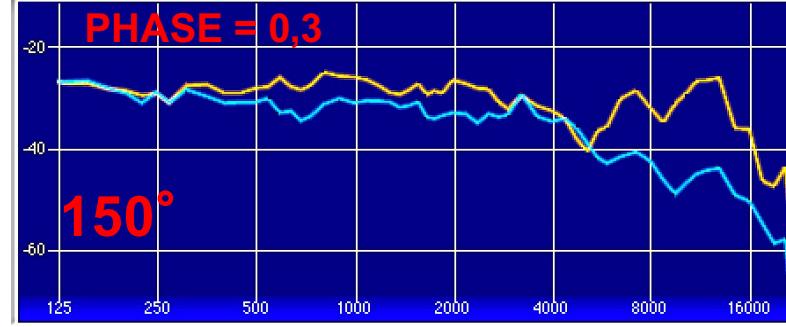
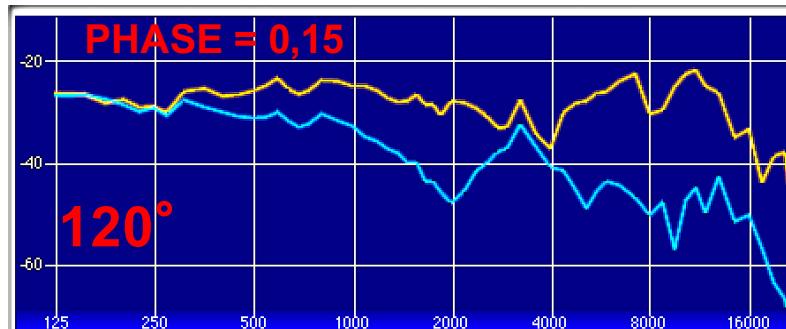
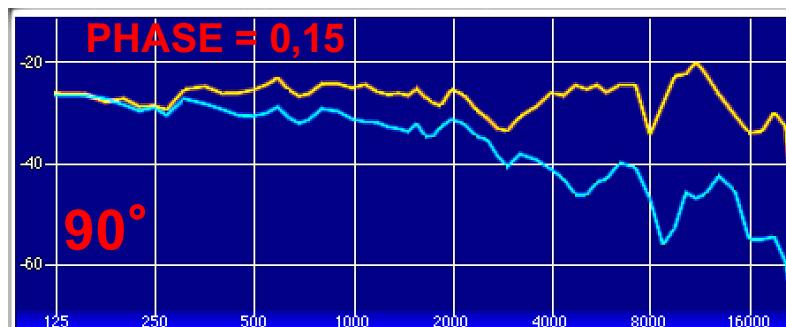
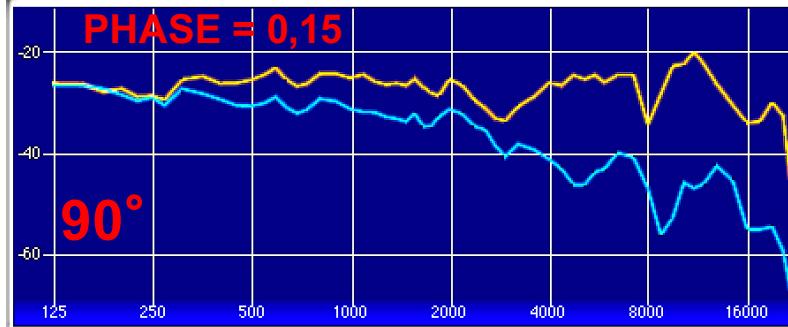
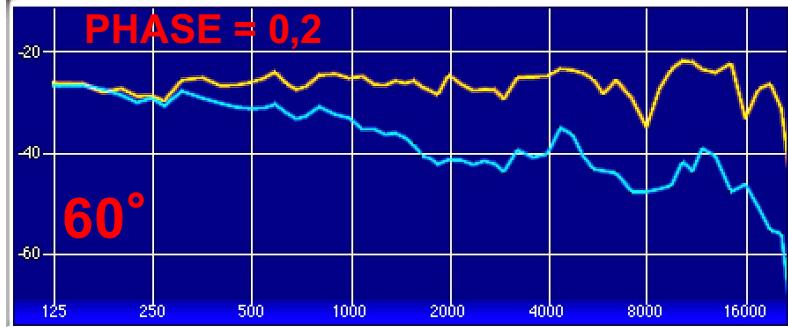
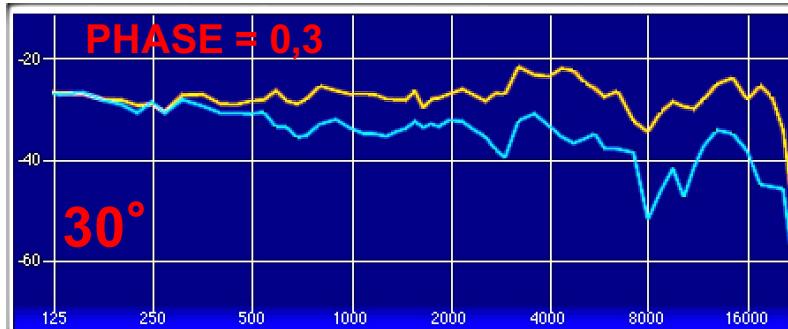
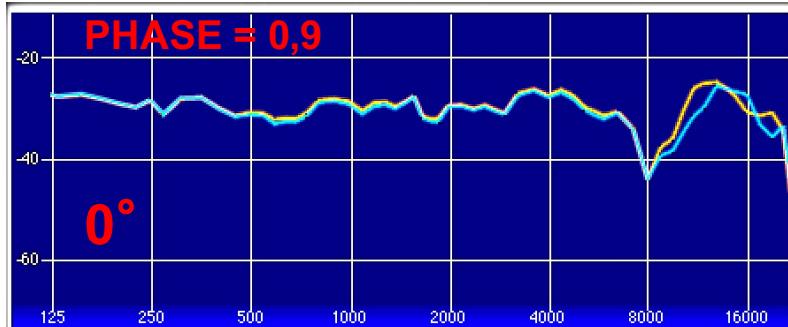
*University of York*



SENNHEISER

# PLUG-IN KU 100 ORBIT

— Oreille Ipsilatéral  
— Oreille Contralatéral



<https://fr-fr.sennheiser.com/ambeo-blueprints-downloads>

Version  
v 0.1.1 BETA

## Convertisseur de coordonnées

Auteur Daniel Rudrich Type Outil Contribution N / A Sortie N / A

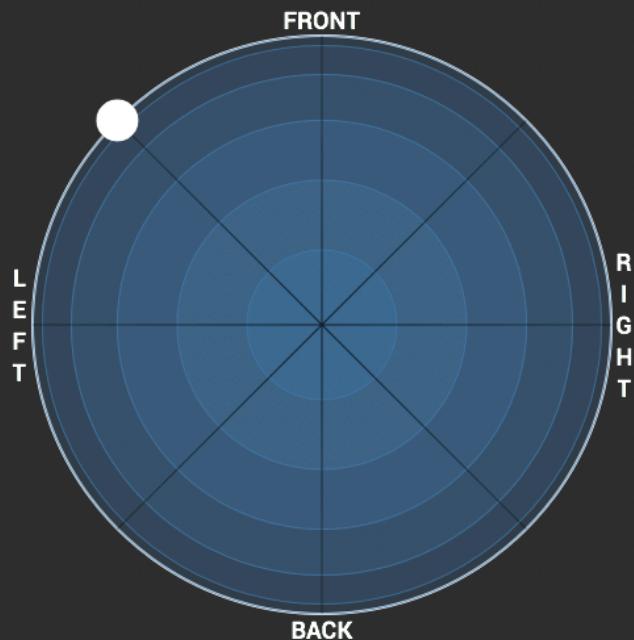
Le convertisseur de **coordonnées** convertit les paramètres VST d'une représentation sphérique en une carte, et inversement. Il n'y a pas de traitement audio, juste la conversion des paramètres. Une application typique serait la conversion de données de position xyz (par exemple dans une automatisation) en une représentation sphérique (azimut, élévation, rayon) pour l'utilisation de codeurs Ambisonic.

Un autre exemple serait l'utilisation avec le codeur de **salle** : en utilisant le panoramique sphérique (coordonnées sphériques) et en mappant les coordonnées cartésiennes calculées à partir du convertisseur de **coordonnées** sur la position source du codeur de **salle**, vous pouvez faire pivoter une source autour de l'écouteur dans le menu déroulant. salle virtuelle, en maintenant le rayon. Pour cela, vous devrez régler la normalisation du rayon sur toute valeur maximale souhaitée et définir les plages x et y sur 15,0 m, la plage z sur 10,0 m, car ces valeurs sont les limites de paramètre des sources x, y et la position z. Il est nécessaire d'ajuster les plages, car les paramètres VST sont normalisés dans une plage allant de 0 à 1.

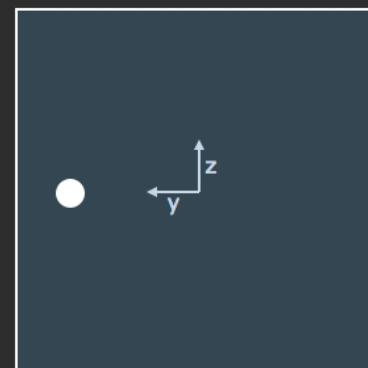
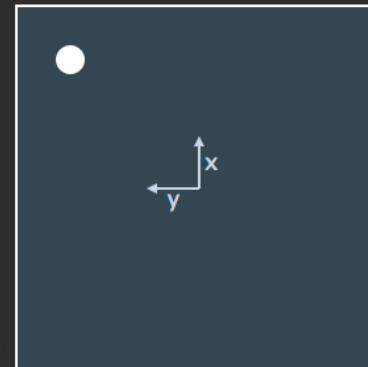
*FLIP = Inversion de l'axe*

# Coordinate Converter

## Spherical Coordinates



## Cartesian Coordinates



45.00 °  
Azimuth

Flip



0.00 °  
Elevation

Flip



1.000  
Radius

Flip



0.7071  
X

Flip



0.7071  
Y

Flip



0.0000  
Z

Flip

## Normalization / Range Settings

1.00 m

Radius

1.00 m

+/- X

1.00 m

+/- Y

1.00 m

+/- Z

## Reference Position

0.000 m

x Ref

0.000 m

y Ref

0.000 m

z Ref

# DirectionalCompressor

Auteur Daniel Rudrich Type Dynamique Contribution Signaux ambisoniques Sortie Signaux ambisoniques

Le compresseur **directionnel** est un compresseur / limiteur Ambisonic plus sophistiqué qui vous permet de contrôler la dynamique pour différentes régions spatiales. Exemple: un chanteur à l'avant pourrait être plus compressé que les autres. Vous pouvez réaliser un effet de ducking, avec lequel le niveau du chanteur peut contrôler le niveau de tout le reste. Utilisez les paramètres de panoramique et de largeur de la sphère pour créer un masque spatial. Vous pouvez écouter les signaux masqués et non masqués en les sélectionnant dans la liste déroulante. Deux compresseurs identiques sont disponibles. Dans chacun, vous pouvez choisir entre trois signaux d'entrée (complet = signal omni, signal masqué et signal non masqué = le reste). Vous pouvez également sélectionner la cible à laquelle les gains du compresseur sont appliqués.

IN SN3D Auto (0th) OUT Directional Compressor

---

FRONT  
L E F T BACK R I G H T

Spatial mask properties

Azimuth: 45.00 ° Elevation: 0.00 ° Width: 50.00 °

General settings

Pre Gain: 0.0 dB Full

Compressor 1 ON

Masked Masked

Threshold: -10.0 dB Knee: 0.0 dB MakeUp: 0.0 dB

Ratio: 4.0 : 1 Attack: 30.0 ms Release: 150.0 ms

Compressor 2 ON

Masked Masked

Threshold: -10.0 dB Knee: 0.0 dB MakeUp: 0.0 dB

Ratio: 4.0 : 1 Attack: 30.0 ms Release: 150.0 ms

iem OSC v1.2.1

## DirectivitéShaper

Auteur Daniel Rudrich Type Directivité codeur Contribution Signal audio mono Sortie Signaux de directivité

Ce plugin vous permet de filtrer votre signal d'entrée en quatre bandes indépendantes, auxquelles différents modèles de directivité peuvent être appliqués. Les bandes peuvent également être balayées spatialement.

Pour chaque bande, vous pouvez choisir parmi quatre types de filtres (passe-tout, passe-bas, passe-bande, passe-haut) qui se mélangent en continu entre différents ordres de directivité, ainsi que différentes formes: basique, maxRE et en phase. Ce dernier s'assure qu'il n'y aura pas de lobes latéraux, au prix d'un lobe principal plus large. Utilisez la sphère en haut à gauche pour faire un panoramique des bandes individuelles. Le point noir représente la sonde avec laquelle vous pouvez mesurer la réponse en fréquence pour cette direction (ligne blanche dans le visualiseur de filtre). Vous pouvez également verrouiller les orientations relatives de toutes les bandes sur la sonde en activant le bouton «Verrouiller les orientations». Maintenant, chaque bande tournera en fonction de l'orientation de la sonde. Voir le [Guide de Directivity Shaper](#) pour un manuel détaillé du plug-in.

1

IN

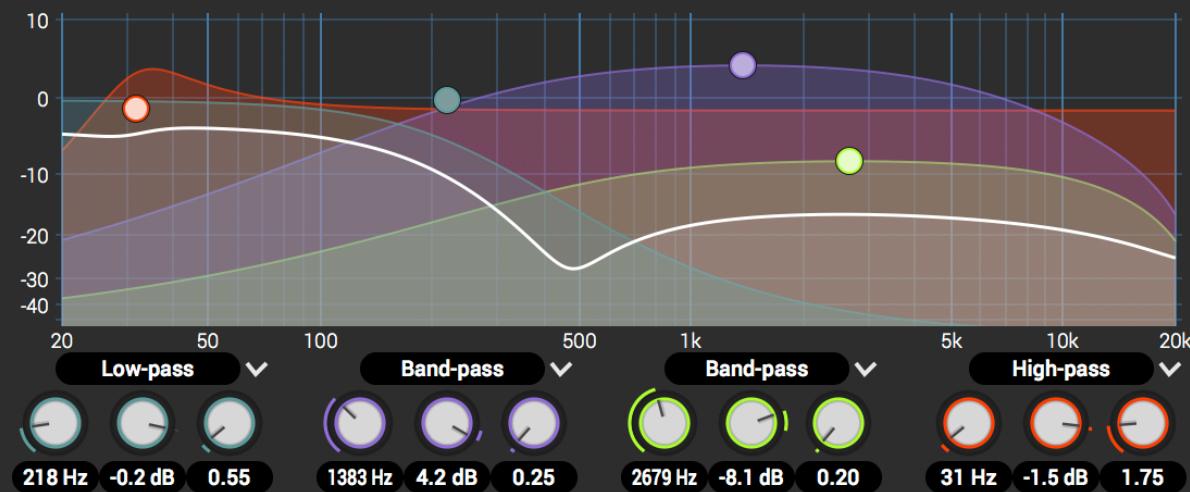
## DirectivityShaper

OUT

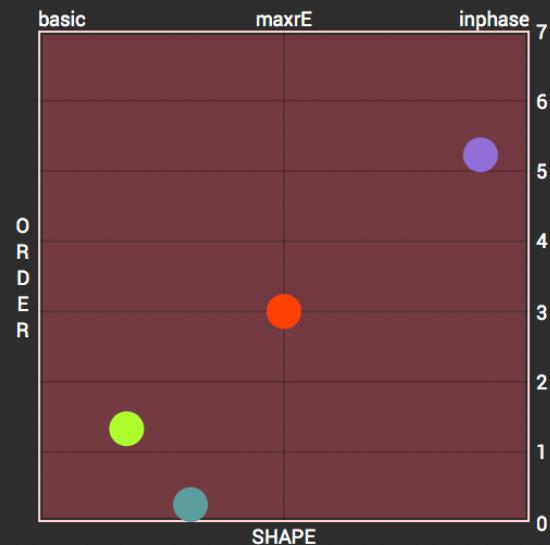
SN3D  
3rd (bus to...)

5

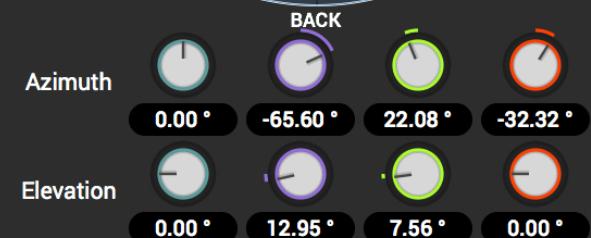
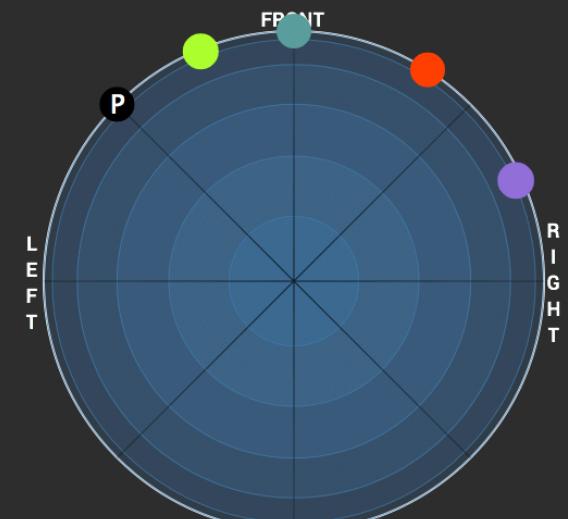
## Filter Bands and Probe Response



## Order and Shape



## Spatial Panning



## Probe Settings and Normalization



Normalization on-axis

# Guide de directivité

Un codeur dépendant de la fréquence et un outil pour créer des modèles de directivité de la source.

- 01 A propos de ce guide
- 02 Bandes de fréquence
- 03 Ordre Ambisonique
- 04 Panoramique spatial
- 05 Sonde
- 06 Signaux de directivité
- 07 RoomEncoder
- 08 À la fin: un devis

## A propos de ce guide

Le **Directivity Shaper** est un plug-in à deux volets: d'une part, il s'agit d'un codeur Ambisonic dépendant de la fréquence; de l'autre, il peut créer des signaux de directivité. Dans ce guide, nous allons d'abord examiner l'encodeur, puis visiter ces mystérieux signaux de directivité qui, pour être honnêtes, peuvent être un peu difficiles à comprendre, mais ouvriront un monde rempli de possibilités. Surtout lors de l'utilisation du **Room Encoder**.

## Bandes de fréquence

Examinons la première section du plug-in: **Bandes de filtrage et réponse au test** Cette section filtre votre *signal audio mono* (voir widget d'entrée) en quatre bandes de fréquence indépendantes. Ces bandes sont codées par couleur et les couleurs respectives seront également utilisées dans les autres sections. Mais concentrons-nous d'abord sur les filtres: vous pouvez régler chacun des quatre filtres à l'aide des commandes situées sous le visualiseur de filtres. Chacun d'entre eux peut être un filtre passe-tout, passe-bas, passe-bande ou passe-haut, avec des fréquences de centre / seuil de coupure ajustables, des gains et des facteurs de qualité. La réponse du filtre est visualisée avec la couleur respective de cette bande de filtre.

**Conseil :** L'utilisation de la molette de la souris avec le curseur de la souris sur l'un des boutons de filtre permet de régler son facteur de qualité.

Sans rien changer d'autre dans le plug-in que les filtres, la ligne blanche dans les visualiseurs de filtre indique la réponse en fréquence globale de ces quatre bandes de fréquences parallèles. Dans ce qui suit, cette ligne sera appelée **réponse de sonde**. J'expliquerai cela plus tard, cela aura du sens, je le promets!

## Ordre Ambisonique

Après avoir divisé notre signal d'entrée en quatre bandes de fréquences, nous allons sélectionner un ordre Ambisonic dans lequel nous voulons encoder nos bandes individuelles. Vous pouvez le faire à l'aide de la section **Order et Shape**. Pour le moment, nous souhaitons uniquement déplacer les quatre boutons colorés (représentant nos bandes de fréquences) sur l'axe des ordonnées (haut / bas). Cela garantit qu'un codage de base sera effectué, comme le ferait tout autre codeur.

La première chose que vous devriez remarquer est que vous pouvez ajuster les commandes individuelles en continu, au lieu de 1ère , 2ème et ainsi de suite. Cool, n'est ce pas? D'accord, même avec des problèmes d'imagerie à quoi ressemble un faisceau de premier ordre? Et maintenant, nous pouvons utiliser Ambisonics au 1.323e ordre? **Pas de soucis !** C'est pourquoi le visualiseur de directivité situé juste à côté du pavé tactile XY vous indique la forme (!) De votre faisceau. C'est la deuxième chose que vous devriez remarquer.

**Remarque :** assurez-vous que la valeur de votre ordre de sortie dans le coin supérieur droit du plug-in est adéquate. Sinon, le pavé XY sera marqué par une zone rouge dans laquelle vous ne devez pas déplacer les points.

*Ok, alors je suppose que vous avez déjà déplacé les boutons vers le milieu, même si je vous ai dit de ne pas le faire. Ensuite, il est temps de rencontrer M. Squidhead (ordre: 5.2, forme: 0.78).*

Retour à la base! Nous nous occuperons bientôt de maxrE, inPhase et de tout ce qui se situe entre les deux. Il est temps de...

## Panoramique spatial

Rappel rapide: signal mono divisé en quatre bandes de fréquences, définissez un ordre Ambisonic pour chaque bande. **Encodons!** Avec le sélecteur de sphère dans le coin supérieur droit, vous pouvez maintenant déplacer chacun des quatre signaux (même couleur!) Dans différentes directions. Les quatre signaux seront codés avec leurs directions et leurs ordres Ambisonic individuels, ce qui donnera un **codeur dépendant de la fréquence** . Cool! Premier chapitre terminé. Passer à autre chose

## Sonde

Vous avez peut-être remarqué ce bouton noir **P** dans le panoramique de la sphère. C'est la **sonde** . Il a deux fonctions:

### Fonction de sonde n ° 1

Vous pouvez déplacer la **sonde** dans le panoramique de la sphère et la **réponse de la sonde** dans le visualiseur de filtre vous montrera la réponse en fréquence des quatre bandes de fréquences mélangées avec leur panoramique individuel. Fondamentalement, qu'entendrait-on si vous écoutez uniquement cette direction? Avec toutes les

commandes définies sur 0, la **réponse de sonde** ne changera pas, car toutes les bandes seront codées sous forme de signaux multiples.

### Fonction de sonde n ° 2

Avez-vous remarqué le petit bouton à bascule nommé *Lock Directions* ? Activez-le, déplacez la **sonde** et observez ce qu'il advient des quatre boutons du panoramique. Ils avancent! Ils sont maintenant verrouillés à la sonde. Vous pouvez toujours les déplacer individuellement, par rapport à la sonde. À propos, le **Multi Encoder** a aussi cette fonctionnalité! Alors pourquoi auriez-vous besoin de cela? Imaginez que vous avez codé un complexe **un e v ERY c OOL de la ounding, f requence- d ependent b eam c onstellation** (ou tout simplement **cavcsfdbc**) et maintenant vous voulez déplacer que **cavcsfdbc** dans une autre direction sans changer leurs directions relatives. *Lock Directions* est votre ami!

## Signaux de directivité

Ok, il est temps de devenir sérieux. Jusqu'ici, l'encodage Ambisonic dépendant de la fréquence avec des ordres fractionnaires était facile, et alors? Part de gâteau. Une promenade dans le parc. Ce que nous faisons maintenant est de sortir des sentiers battus, eh bien, nous penserons à l'intérieur de la boîte à chaussures du **Room Encoder**.



Alors, quel est un signal de directivité? Pour répondre à cette question, nous devons d'abord examiner le processus d'encodage et de décodage dans Ambisonics: en général, lors du **codage d'** un signal audio dans Ambisonics, aucun **poids** n'est appliqué, ce qui crée un motif de faisceau que vous avez vu dans le visualiseur de directivité. Ces faisceaux ont un lobe principal et plusieurs lobes latéraux très distincts. **Remarque** : tous les autres lobes latéraux ont une phase inversée! Les modèles de faisceaux vous indiquent la quantité d'énergie projetée sur une sphère virtuelle autour de vous. Encodons donc une bande de fréquence dans la direction **avant**, avec **2ème ordre** et **baseforme**. La majeure partie de l'énergie de ce signal est projetée vers l'avant, comme vous le souhaitez. Mais il y a aussi de l'énergie dans le sens opposé, heureusement avec une atténuation de -10 dB (ce qui n'est pas beaucoup). Et il y a deux lobes latéraux pointant vers la gauche et la droite, avec environ 15dB et un **basculée** signe, car ils sont ceux à côté du lobe principal.

Cela pourrait être surprenant, car vous ne penseriez jamais à envoyer un signal dans la direction opposée, loin de la direction souhaitée! Je peux te rassurer, tout n'est pas perdu! Il y a toujours notre étape de **décodage**. La plupart des décodeurs appliquent des **poids** spéciaux à vos signaux Ambisonic avant de les décoder réellement vers vos haut-parleurs. Vous devez connaître deux poids spécifiques: **maxrE** et **inPhase**.

Regardons notre faisceau de second ordre, décodé avec les poids **maxrE**. Nous pouvons simuler cela en déplaçant notre contrôle de forme sur **maxrE** (ou tout simplement en définissant la valeur de la forme sur 0,5). Nos lobes ont changé! Le lobe principal est devenu un peu plus large, mais les lobes secondaires indésirables sont maintenant encore plus atténués. Avec **inPhase** (tout à fait à droite, ou 1,0), il n'y aura **pas de lobes latéraux**, toutefois, au prix d'un lobe principal plus large.

**Remarque** : le décodeur **AllRA** créera un décodeur avec une pondération **maximale**. Cependant, vous pouvez éditer le fichier de configuration exporté, remplacer **maxrE** par **inPhase** et le charger dans le décodeur **simple** afin de faire l'expérience des pondérations **inPhase**.

Pour résumer cette excursion: le processus de codage-décodage peut être considéré comme:

**CODER -> MASSES DE + DECODAGE**

La première partie est pris en charge par un **codeur Ambisonic** et les deux derniers par un **décodeur**.

Revenons maintenant à notre question: **qu'est-ce qu'un signal de directivité?**

**Réponse :** Un *signal de directivité* est un signal Ambisonic, mais avec des poids spécifiques qui lui sont appliqués. Ces poids peuvent dépendre de la fréquence. En revanche: les *signaux Ambisonic* ont **pas** le poids qui leur sont appliqués: de base.

Dans le **Directivity Shaper**, vous pouvez régler ces poids en déplaçant les bandes de fréquence individuelles dans le pavé XY.

Question de suivi: **à quoi servent les signaux de directivité? Mon décodeur n'applique-t-il pas déjà des poids?**

**Réponse** Oui, votre décodeur applique déjà des poids à vos signaux. Alors, pourquoi devriez-vous appliquer des poids vous-même?

## **Il est maintenant temps de sortir des sentiers battus!**

Tout d'abord: appliquer des poids n'a que peu de sens lorsque vous utilisez les signaux comme signaux Ambisonic normaux et que vous les décodez en signaux haut-parleur. Cela n'aurait de sens que si votre décodeur n'applique *pas* de poids. Cependant, tout ce qui sonne bien est bon!

**Conseil :** Vous pouvez remplacer l'attribut du fichier de configuration par **aucun**, puis aucune pondération n'est appliquée.

## **Gedankenexperiment (je suis surpris que ce soit un mot anglais)**

*Ne voyons pas les signaux codés et pondérés sous forme de signaux projetés / codés sur une sphère autour de vous, émettant de l'énergie vers vous (à l'intérieur). Voyons plutôt cela comme une minuscule sphère, tellement minuscule que c'est fondamentalement un point. Vous pourriez l'appeler une source . Une source irradiant de l'énergie vers l'extérieur. Loin de son centre. L'énergie, qui provient d'un signal audio mono, divisé en quatre bandes de fréquences et orientée dans différentes directions.*

Alors, où se situe cette **source**? Comment puis-je l'écouter? Puis-je utiliser un haut-parleur?

Si vous avez un haut-parleur, vous pouvez émettre un son dans différentes directions: **oui!**

Cependant, vous pouvez également utiliser un *haut-parleur virtuel* (ou *une source virtuelle*) dans une *pièce virtuelle* et écouter les signaux émis par la source et réfléchis par les murs de la pièce, qui se trouvent pratiquement dans cette pièce. Je pense que vous savez déjà où je veux en venir.

## **RoomEncoder**

Allumons le codeur de **salle** directement après le transformateur de **directivité**.

**Avertissement :** Assurez-vous absolument que le réglage de sortie du **Directivity Shaper** correspond au réglage d'entrée du **Room Coder**. Sinon et avec des réglages d'ordre malheureux, des signaux très forts pourraient être générés!

Le point orange est cette **source**, dont le diagramme de rayonnement peut être défini avec **Directivity Shaper**. Par exemple, avec les bandes orientées vers l'avant, ces signaux seront émis vers le mur avant. L'auditeur est représenté par le point vert. Il reçoit le signal direct (par défaut venant de l'avant gauche) et toutes les réflexions sur le mur (venant de différentes directions).

**Qu'est-ce qui se passe sous le capot?** Le codeur de **salle** décode le signal de directivité d'entrée dans chaque direction nécessaire pour calculer le signal direct et les réflexions, **sans** pondération. Décodage sans poids? Bien que nous ayons déjà appliqué des poids dans le **Directivity Shaper**! Les signaux seront atténusés en fonction de la distance et des propriétés de réflexion, puis encodés en un signal Ambisonic, sans pondération, bons signaux de base Ambisonic.

*Signaux ambisoniques, vous pouvez ensuite les décoder vers votre réseau de haut-parleurs. Euh... est-ce que cette solution s'appelle pièce dans pièce? Que se passerait-il si j'envoyais ces signaux dans une autre instance de **Room Encoder**? Ou même le nourrir? Chambre récursive?*

## À la fin: un devis

*Les signaux de directivité, qui - pour être honnêtes - peuvent être un peu difficiles à comprendre, mais ouvriront un monde plein de possibilités.*

Daniel Rudrich, il y a quelques minutes.

---

Plug-in Suite de IEM - Institut de musique électronique et d'acoustique

IEM.at

Docs

Site créé avec [Hugo](#) sur le thème du [kube](#).

# Compensateur de distance

Auteur Daniel Rudrich Type Compensation de haut-parleur Contribution Signaux audio Sortie Signaux audio

Avec le compensateur de **distance**, vous pouvez insérer les distances entre la position d'écoute et chaque haut-parleur. Le plug-in calculera les délais et les gains nécessaires afin de compenser les différences de distance. Donc, ce plug-in est le meilleur endroit à la fin de votre chaîne de signal (par exemple juste après les décodeurs).

La compensation de gain et de retard peut être activée / désactivée séparément et leurs paramètres de calcul peuvent également être définis. En ce qui concerne les délais, il est possible de spécifier la vitesse du son, soit 343,2 m / s (à 20 ° C) par défaut. Les gains sont calculés à l'aide d'un exposant rapport de distance, qui vaut 1,0 dans les conditions du champ libre. Dans les pièces, des valeurs plus petites pourraient être plus appropriées en raison des réflexions sur les murs. De plus, vous pouvez spécifier s'il doit y avoir uniquement des atténuations ou si vous utilisez le réglage zéro-moyenne, ce qui aura également des gains positifs appliqués aux haut-parleurs plus éloignés de l'auditeur.

Il est également possible de charger les informations de distance via des fichiers de configuration contenant un objet LoudspeakerLayout (voir le [Guide des fichiers de configuration](#) pour plus d'informations). Une fois le LoudspeakerLayout chargé, vous pouvez définir la position de référence à laquelle les haut-parleurs sont compensés.

**IN**  7 (bus too s... ▾ **DistanceCompensator**

---

**Settings**

Gain compensation  
Distance-Gain exponent 1.0  
Normalization Attenuation only ▾

Delay compensation  
Speed of sound 343.2 m/s

**From loudspeaker layout**

**LOAD LAYOUT**

**Reference position**  
0.00 m 0.00 m 0.00 m  
x y z

**UPDATE REFERENCE**

---

**Loudspeaker Distances**

1	 5.00 m
2	 5.00 m
3	 5.00 m
4	 5.00 m
5	 5.00 m
6	 5.00 m
7	 5.00 m

**OUT**

iem ○ osc v0.2.1

# Fichiers de configuration

Ce guide explique comment les fichiers de configuration sont structurés.

- 01 Informations générales
- 02 Structure des fichiers de configuration
- 03 L'objet TransformationMatrix
- 04 L'objet décodeur
- 05 L'objet LoudspeakerLayout
- 06 Comment créer une matrice de décodeur
- 07 Exemple de configuration de décodeur
- 08 Exemple de configuration de TransformationMatrix

## Informations générales

Tous les fichiers de configuration du décodeur **simple** et du multiplicateur de **matrice** utilisent le format JSON. Voir <https://json.org> pour des informations détaillées sur ce format. Nous sommes conscients que nous introduisons un changement radical en passant des fichiers de configuration de Kronlachner au format JSON. Cependant, nous pensons que le format JSON est plus évolutif et évolutif, en particulier pour les fonctionnalités que nous prévoyons d'introduire à l'avenir (par exemple, MultibandDecoder). De plus, il est plus facile d'enregistrer et de décoder, il est lisible par l'homme et des éditeurs de texte décents vous alerteront si les données ne sont pas conformes à JSON.

L'idée est de disposer d'un fichier de configuration pour l'ensemble de votre projet, comprenant des informations sur le décodage, le routage, la position des haut-parleurs et des opérations de matrice supplémentaires.

## Structure des fichiers de configuration

La structure de base ne contient que les attributs de nom et de description de la configuration et ressemble à ceci:

```
{  
    "Name" : "name of this configuration",  
    "Description" : "description text"  
}
```

Il peut être étendu en ajoutant des objets à la configuration. Pour l'instant, seuls les objets suivants sont définis:

- [TransformationMatrix](#)
- [Décodeur](#)
- [Haut-parleur](#)

Chacun d'entre eux a quelques autres sous-objets, décrits ci-dessous.

Ainsi, une configuration de décodeur pourrait ressembler à ceci (notez que vous devez ajouter les attributs nécessaires à l'objet décodeur):

```
{  
    "Name" : "name of this configuration",  
    "Description" : "description text",  
    "Decoder" : {  
        <put decoder information here>  
    }  
}
```

Il y aura plus d'objets à l'avenir. Par exemple, les définitions des haut-parleurs permettent de visualiser leur position; ou un décodeur multibande constitué d'un ensemble de décodeurs avec des informations supplémentaires sur les bandes de fréquence.

## L' TransformationMatrix objet

Avec l' `TransformationMatrix` objet, vous pouvez intégrer les fonctionnalités de transformation à votre configuration. Vous pouvez définir les attributs suivants:

- **Name** : Le nom de votre matrice sous forme de chaîne. (**optionnel**)
- **Description** : La description de votre matrice sous forme de chaîne. (**optionnel**)
- **TransformationMatrix** : La matrice de transformation donnée sous la forme d'un tableau à deux dimensions de flotteurs. Voir exemple ci-dessous.

## L' Decoder objet

Avec l' `Decoder` objet, vous pouvez ajouter une fonctionnalité de décodeur à votre configuration. Vous pouvez définir les attributs suivants:

- `Name` : Le nom de votre décodeur sous forme de chaîne. (**optionnel**)
- `Description` : La description de votre décodeur sous forme de chaîne. (**optionnel**)
- `ExpectedInputNormalization` : Définit la normalisation attendue du signal d'entrée Ambisonic. Peut être "n3d" ou "sn3d".
- `Weights` : Définissez le type de pondération utilisé par votre décodeur. Peut être `none`, `inPhase` ou `maxRE`.
- `WeightsAlreadyApplied` : Définit si les poids sont déjà appliqués à la matrice (valeur booléenne: vrai ou faux).
- `SubwooferChannel` : Définit le canal du subwoofer comme un entier. S'il est défini, le décodeur **simple** utilisera le mode subwoofer discret et acheminera le signal d'entrée filtré vers ce canal. (**optionnel**)
- `Matrix` : La matrice de décodeur donnée comme un tableau à deux dimensions de flotteurs. Voir [comment faire](#) et [exemple](#) ci-dessous.
- `Routing` : Définit un routage de sortie comme un tableau d'entiers. Chaque signal de haut-parleur sera acheminé vers le numéro de canal déclaré. Voir exemple ci-dessous. (**optionnel**)

## Remarques

La spécification de l'attribut `weights` est nécessaire pour la fonctionnalité d'ordre variable. Le plug-in du décodeur appliquera automatiquement les poids corrects en fonction de l'ordre Ambisonic sélectionné du signal d'entrée. Si les poids sont déjà appliqués à la matrice, le décodeur extraira l'ordre Ambisonic de la taille d'entrée de la matrice et créera en interne une matrice sans les poids.

## L' `LoudspeakerLayout` objet

L' `LoudspeakerLayout` objet contient des informations sur la disposition des haut-parleurs, qui peuvent être chargées dans le plug-in **AllRA Decoder**, par exemple pour créer un décodeur. Vous pouvez définir les attributs suivants:

- `Name` : Le nom de votre disposition de haut-parleur sous forme de chaîne. (**optionnel**)
- `Description` : Description de la disposition de vos haut-parleurs sous forme de chaîne. (**optionnel**)
- `Loudspeakers` : Ceci est un tableau contenant la liste des haut-parleurs.

Chaque haut-parleur de l' `Loudspeakers` objet (tableau) contient des informations sur ses coordonnées dans la représentation sphérique, un indicateur indiquant s'il s'agit d'un haut-parleur imaginaire (important pour l'approche AllRAD), un numéro de canal permettant d'acheminer un signal vers et une valeur de gain.

Voici une liste de ces attributs:

- `Azimuth` : L'angle d'azimut du haut-parleur en degrés, en tant que valeur flottante. Avant serait  $0^\circ$ , droite  $-90^\circ$ , gauche  $90^\circ$  (règle de la main droite).
- `Elevation` : L'angle d'élévation en degrés sous forme de valeur flottante. Le plan horizontal est à  $0^\circ$ , le zénith à  $90^\circ$  et le nadir à  $-90^\circ$ . L'utilisation de valeurs supérieures à  $90^\circ$  ou inférieures à  $-90^\circ$  n'est pas recommandée.
- `Radius` : Distance de la position d'écoute en mètres comme valeur flottante.
- `IsImaginary` : Indique si le haut-parleur est imaginaire (valeur booléenne: vrai ou faux).
- `Channel` : Le numéro de canal auquel les signaux seront acheminés sous forme de valeur entière.
- `Gain` : Un gain linéaire sous forme de valeur flottante.

## Remarques

Le plug-in **AllRA Decoder** utilise le `Gain` et `Radius` uniquement pour des haut-parleurs imaginaires, de sorte que le gain n'affectera pas les vrais haut-parleurs. Consultez le [Guide AllRADecoder](#) pour plus d'informations. Vous pouvez également utiliser le décodeur **AllRA** pour exporter la disposition des haut-parleurs dans un fichier de configuration.

## Comment créer une matrice de décodeur

Outre une configuration de haut-parleur bien conçue, un décodeur Ambisonic décent est essentiel pour la lecture d'un signal audio Ambisonic. Créer un décodeur n'est pas une mince affaire et il existe plusieurs stratégies pour le faire. À l'IEM, nous utilisons la stratégie [AllRAD](#) pour créer nos décodeurs. Nous avons créé le plug-in **AllRA Decoder** pour concevoir un décodeur Ambisonic adapté à votre configuration d'enceintes. Comme son nom l'indique, il utilise également l'approche AllRAD. Voir le [Guide AllRADecoder](#) pour un [tutoriel](#).

Comme alternative, le [Ambisonic décodeur ToolBox](#) est un outil pratique pour Matlab et Octave, qui soutient également que **tous - R ound A mbisonic D ecoding**. Vous pouvez l'utiliser pour créer une matrice de décodeur et l'exporter avec celle de Matlab `jsonencode`.

## Exemple de configuration de décodeur

Cette configuration contient un objet `Decoder`, dont la matrice est de taille `12x36` (sortie: 12 haut-parleurs; entrée: Ambisonics de cinquième ordre). Le canal de sortie 4 est ignoré par le routage, car ce canal est celui auquel le subwoofer est connecté.

```
{
  "Name" : "Produktionsstudio decoder",
  "Description" : "",
  "Decoder" : {
    "Name" : "Produktionsstudio decoder 05",
    "Description" : "This is a decoder for the IEM Produktionsstudio.",
    "ExpectedInputNormalization" : "n3d",
    "Weights" : "maxrE",
    "WeightsAlreadyApplied" : true,
    "SubwooferChannel" : 4,
    "Matrix" :
      [[0.524667,0.469572,-0.209473,0.540143,0.535468,-0.176045,-0.20627,-0.207037,0.07898,
        0.278071,-0.171502,-0.07617,0.080337,-0.087312,-0.031127,-0.168645,0.052887,-0.068575,
        -0.033469,0.013674,0.034073,0.018059,-0.004703,0.03349,-0.135629,-0.00736,-0.01102,
        -0.003696,-0.006332,0.01286,0.000698,0.015034,0.000453,0.00238,0.013449,-0.038748],

      [0.52467,-0.469604,-0.20946,0.540139,-0.535474,0.17603,-0.2063,-0.207068,0.078934,
        -0.278033,0.171505,0.076174,0.080367,-0.087303,-0.031177,-0.168678,-0.052849,0.068601,
        0.033452,-0.013671,0.034052,0.018066,-0.004701,0.033452,-0.135628,0.007376,0.011046,
        0.003689,0.006338,-0.012852,0.000708,0.015035,0.000457,0.002368,0.013438,-0.038739],

      [0.48232,9E-6,-0.046894,0.673631,5E-6,-2E-6,-0.19333,-0.042387,0.538062,
        -6E-6,-1E-6,2E-5,-0.030678,-0.09633,-0.027138,0.356096,-1.2E-5,-1E-6,
        2.9E-5,-6E-6,0.006201,-0.048245,-0.016251,-0.020807,0.186287,-9E-6,-2E-6,
        1.7E-5,-6E-6,-3E-6,0.014683,-0.0051,-0.023478,0.004609,-0.013344,0.06439],

      [0.85076,0.503559,-0.266537,-0.952247,-0.541567,-0.191062,-0.337222,0.256713,0.330954,
        0.232285,0.174682,-0.082557,0.078565,0.147907,-0.038838,0.043984,-8E-6,-0.05472,
        0.034604,0.016569,0.042964,0.003534,-0.010906,-0.036433,-0.073705,-0.028779,-6E-6,
        -0.003308,0.00572,0.014052,0.007977,-0.014555,-0.011586,-0.006084,0.011077,0.01356],

      [0.850906,-0.503624,-0.26653,-0.952374,0.54162,0.191019,-0.337233,0.256664,0.330977,
        -0.232295,-0.174643,0.08258,0.078468,0.147914,-0.038837,0.043988,1.8E-5,0.05471,
        -0.034603,-0.016557,0.042992,0.003586,-0.010914,-0.03641,-0.073692,0.028764,2E-6,
        0.003306,-0.005728,-0.014047,0.007974,-0.014559,-0.011597,-0.006063,0.011066,0.013551],

      [0.75186,1,-0.156013,0.078043,0.104081,-0.172615,-0.296441,-0.029433,-0.689955,
        -0.346121,-0.035164,-0.144883,0.009708,-0.012092,0.089756,-0.071726,-0.022223,0.030376,
        -0.005582,-0.03736,0.022544,0.002024,0.022081,0.020618,0.109559,0.011133,0.0051,
        -0.006416,-0.001135,0.003645,0.015434,0.002047,0.023454,0.000338,-0.005592,-0.003701],

      [0.751767,-0.999922,-0.156027,0.07806,-0.104104,0.172666,-0.296466,-0.029437,-0.689946,
        0.346149,0.035161,0.144916,0.009764,-0.012099,0.08978,-0.071742,0.022225,-0.030366,
        0.005591,0.037336,0.02256,0.002004,0.022082,0.020616,0.109586,-0.011143,-0.0051,
        0.006428,0.001146,-0.003654,0.015432,0.002053,0.023463,0.000337,-0.005577,-0.003705],

      [0.637174,0.451681,0.620473,0.452977,0.287188,0.468269,0.139043,0.468783,0.001525,
        0.051188,0.271546,0.194653,-0.14563,0.194029,0.000703,-0.04915,0.000835,0.034811,
        0.09865,0.001314,-0.108523,0.000868,-0.000392,-0.033847,0.00361,0.006586,0.000343,
        0.002819,0.001608,-0.01643,-0.017683,-0.016347,-0.000279,-0.00303,0.005699,0.006292],

      [0.637184,-0.451677,0.62047,0.452974,-0.287146,-0.46826,0.13902,0.468777,0.001515,
        -0.051165,-0.2715,-0.194646,-0.145654,0.194027,0.000686,-0.049135,-0.00084,-0.034777,
        -0.09864,-0.001314,-0.108533,0.000869,-0.000403,-0.033837,0.003608,-0.006587,-0.000337,
        -0.002805,-0.001621,0.016428,-0.017681,-0.016347,-0.000283,-0.003032,0.005699,0.006286],
```

```

[0.638428,0.453622,0.620934,-0.453725,-0.288777,0.469057,0.138133,-0.469196,3E-6,
0.050871,-0.272325,0.193751,-0.146243,-0.193823,2.8E-5,0.050953,3.7E-5,0.034674,
-0.098291,0.000659,-0.108277,-0.000645,1.8E-5,0.034723,0.003108,0.006353,1.2E-5,
0.002849,-0.001321,-0.016311,-0.017459,0.016335,-6E-6,0.002865,0.005478,-0.006362],

[0.638375,-0.453609,0.620938,-0.453603,0.288705,-0.469075,0.138194,-0.469142,-6.4E-5,
-0.05085,0.272288,-0.193774,-0.146229,-0.19386,-5E-6,0.050949,-2.4E-5,-0.03467,
0.098293,-0.000651,-0.108301,-0.000667,3E-5,0.034709,0.003096,-0.006357,-1E-6,
-0.002853,0.001324,0.016324,-0.017463,0.016343,4E-6,0.002853,0.005472,-0.006355],

[0.413602,1.9E-5,0.610689,5.3E-5,1.9E-5,3E-5,0.565073,7.3E-5,1E-5,
1E-6,2.7E-5,2.6E-5,0.38914,5.4E-5,1.7E-5,-2.1E-5,2E-6,1E-6,
1.9E-5,1.3E-5,0.192325,1.8E-5,1.6E-5,-2.8E-5,-0.0034,3E-6,3E-6,
1E-6,5E-6,1E-6,0.054697,-3E-6,8E-6,-1.7E-5,-0.004302,-3E-6]],

"Routing" : [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13]
}

}

```

## Exemple de configuration de TransformationMatrix

Cet exemple montre une configuration simple qui utilise un TransformationMatrix.



Plug-in Suite de IEM - Institut de musique électronique et d'acoustique

IEM.at

Docs

Site créé avec [Hugo](#) sur le thème du [kube](#).

## DualDelay

Auteur Daniel Rudrich Type Effet de retard Contribution Signaux ambisoniques Sortie Signaux ambisoniques

Le Dual Delay est la version plug-in de notre [RotationalDelay que](#) nous avons utilisé pour la première fois lors du concert d'Al Di Meola. Il existe deux lignes à retard pouvant être configurées indépendamment. La caractéristique la plus intéressante est la rotation de tout le champ sonore à chaque passage sur la ligne à retard. Le timbre du délai peut être façonné avec les filtres passe-haut et passe-bas. La sortie de chaque ligne à retard peut être renvoyée à elle-même ou à l'autre et mélangée au signal de sortie.

IN SN3D Auto (0th) OUT DualDelay

---

**Rotation & Delay**

10.0 °	500.0 ms	0.00 Hz	0.00 ms
Yaw	Delay	Rate	LFO Depth

-7.5 °	375.0 ms	0.00 Hz	0.00 ms
Yaw	Delay	Rate	LFO Depth

---

**Spectral Filter**

100.0 Hz	- +	HighPass	Cutoff Frequency	- + 20000.0 Hz	LowPass
----------	-----	----------	------------------	----------------	---------

100.0 Hz	- +	HighPass	Cutoff Frequency	- + 20000.0 Hz	LowPass
----------	-----	----------	------------------	----------------	---------

---

**Feedback**

-8.0 dB	-20.0 dB
Self	Cross

---

**Output Mix**

-6.0 dB	0.0 dB	-6.0 dB
Delay I	Dry	Delay II

---

**Feedback**

-8.0 dB	-20.0 dB
Self	Cross

jem ○ OSC v0.3.1

# Efficient Spatial Ambisonic Effects for Live Audio

(*Effiziente räumliche Ambisonics-Effekte zur Live-Beschallung*)

D. Rudrich, F. Zotter, M. Frank

*Institute of Electronic Music and Acoustics*

*University of Music and Performing Arts Graz, Austria*

*Email: {rudrich, zotter, frank}@iem.at*

## Abstract

Animated by the award-winning, free ambiX/mcfx VST plugins that improve efficiency and applicability of 3D audio productions in Ambisonics, this contribution demonstrates an example of their successful usage and extension as live effects in a latency-critical sound-reinforcement situation. We present 3D-audio live effects developed for and used during the concert performance of Al Di Meola – Elysium Unplugged & More in July 2016, Graz:

- Spatial Chorus spins the sound source around the audience at controllable speed, similar to the effect inside a Leslie speaker.
- Multi Delay creates an immersive reverb-like sound by a large number of parallel feedback delays. In contrast to conventional reverbs, the clarity of the direct sound is not impaired.
- Rotational Delay consists of two feedback delays with a slightly clockwise/anti-clockwise shifting of the delayed sound. This effect envelopes the audience with a flow of previously played phrases.

## 1. Introduction

Recent developments in Ambisonics enabled its practical application in the production of 3D audio as discussed in [1]. Scientific works of D. Malham [2], A. Sontacchi [3], maybe most of all the one of J. Daniel [4–6] were fundamental in making higher-order Ambisonics applicable, and the various software tools such as iem\_ambi for Pd<sup>1</sup>, AmbDec<sup>2</sup>, WigWare<sup>3</sup>, Ambisonic Studio<sup>4</sup>, ICST Ambisonics Tools for Max/MSP<sup>5</sup>, Ambisonic Toolkit for SuperCollider<sup>6</sup>, AmbIEM for SuperCollider<sup>7</sup>, HOA VST plugins<sup>8</sup>. An extensive list can be found on Wikipedia<sup>9</sup>.

The most recent improvement of the applicability were advancements in reliable decoding [7, 8] and plugins to extensively produce in Ambisonics, such as [9–13].

As outlined in [14], Ambisonics can be used as a capable multi-venue infrastructure for several playback facilities [15]. Currently, its distribution and storage format is adopted by new media companies such as youtube/google/oculus, suggesting to use the ambiX file format [16].

Besides the complex broadcasting and multi-venue playback capacities, efficient spatialization of sound sources is achieved for large audiences [17, 18], and it provides the possibilities to develop new immersive real-time audio effects, e.g. those based on SDM [19], as discussed in [17].



**Fig. 1:** Ligeti hall with its 29 loudspeakers hanging from the ceiling, and musicians in concert.

Much has been achieved recently in terms of Ambisonic effects, recently, with the ambiX plugin suite, which is able to implement some of the spatial mastering concepts [20], Ambisonic Widening [21], and more natural things such as mirroring, rotation, but also warping, as described in [10, 22]. The mcfx plugin suite allows to implement complex spatial convolutive SDM-based reverberation or the virtualization of spatial MIMO systems [23].

So far, Ambisonics is applicable (semi-) offline in audio sequencers. With the usage in Al Di Meola's concert in July 2016 in Graz as both spatialization and live effect unit, we prove that its efficiency is also real-time capable with low latencies.

<sup>1</sup>by Thomas Musil, [https://git.iem.at/pd/heim\\_ambi](https://git.iem.at/pd/heim_ambi)

<sup>2</sup>by Fons Adriaensen <http://kokkinizita.linuxaudio.org>

<sup>3</sup>by Bruce Wiggins, [http://www.brucewiggins.co.uk/?page\\_id=78](http://www.brucewiggins.co.uk/?page_id=78)

<sup>4</sup>by Daniel Courville, <http://www.radio.uqam.ca/ambisonic/>

<sup>5</sup>by Jan Schacher and Philippe Kocher,

[https://www.zhdk.ch/index.php?id=icst\\_ambisonicsexternals](https://www.zhdk.ch/index.php?id=icst_ambisonicsexternals)

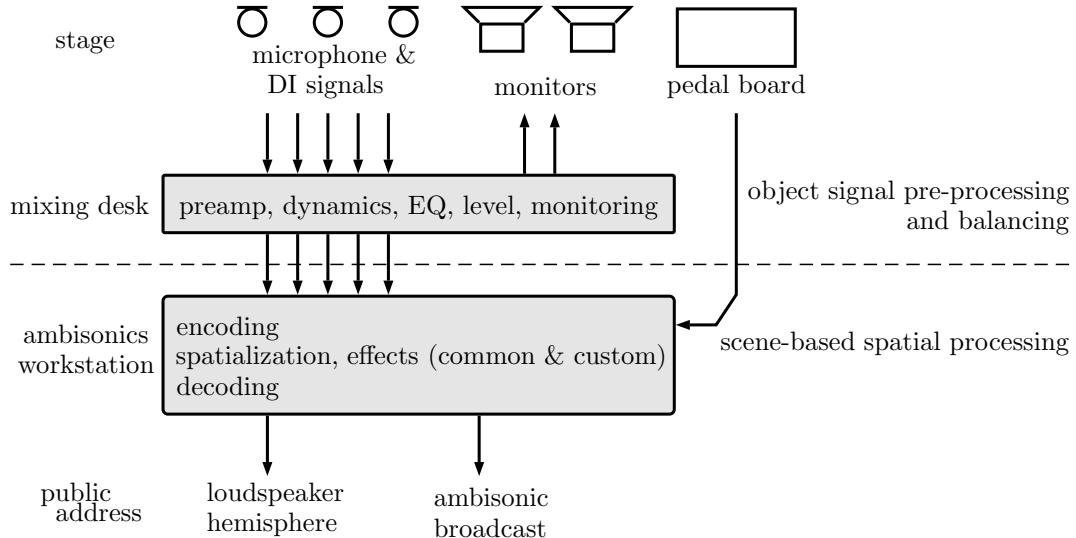
<sup>6</sup>by Joseph Anderson, <http://www.ambisonictoolkit.net/>

<sup>7</sup>by Christopher Frauenberger et al

<https://github.com/supercollider-quarks/AmbIEM>

<sup>8</sup><http://www.mshparisnord.fr/hoalibrary/>

<sup>9</sup>[https://en.wikipedia.org/wiki/List\\_of\\_Ambisonic\\_Software](https://en.wikipedia.org/wiki/List_of_Ambisonic_Software)



**Fig. 2:** Audio Setup of Al Di Meola Concert in Graz, July 2016.

## 2. Concert Audio Setup

The concert audio setup is depicted in Fig. 2. The *mixing desk* was used for standard audio signal processing of the microphone and direct signals from stage including leveling, dynamics, EQ and monitoring. The processed signals were sent to the *ambisonics workstation* (Reaper with ambiX and mcfx plugin suite) for Ambisonic encoding and effect processing (spatialization, widening, reverb, common and custom effects) with 5<sup>th</sup>-order resolution (36 channels). The blocksize for the Ambisonic processing was set to  $2 \times 128$  samples, resulting in a latency of about 6 ms. A *pedal board* on stage allowed a remote engaging and control of the custom effects by the musician.

The full scene was decoded to the loudspeaker array of the venue (hemisphere with 29 loudspeakers). Also the Ambisonic signals and a video stream were broadcasted live to another venue (12-loudspeaker hemisphere) to address a larger audience. Both venues used 5<sup>th</sup>-order AllRAD [7].

## 3. Common Effects

The following spatialization/room effects were applied to every input channel and considered direction, source width/depth, first reflections, and reverberation.

### 3.1. Spatialization

To bring the mono signals into the Ambisonics domain, the *ambix encoder* plugin was used. It allows signal spatialization to any position of the Ambisonic sphere, either statically or dynamically using its built-in OSC interface. E.g. an automatization can be triggered moving the source on a predefined trajectory. Also with sending random azimuth and elevation angles, the signal can be spatialized to random positions. Especially for percussive instruments this creates an interesting and vivid effect.

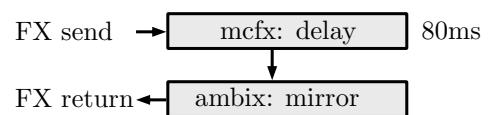
### 3.2. Widening

With the *ambix widening* plugin a source can be widened [21] which was used to create depth, which is useful to separate two sound sources with similar direction and level, e.g. rhythm and lead guitar. This achieved by choosing the single-sided mode to prevent pre-echos and a modulation of about 10 ms to create an impulse response similar to early reflections. The modulation depth can be used to control the perceived depth.

### 3.3. Slapback Delay

The *Ambisonic slapback delay* is a simple but effective way to create a first reflection of the signal. This is easily done by mirroring the input signal at the origin and applying a short delay as shown in Fig. 3. It was used to efficiently add a room impression to spatially pronounce transient signals.

For setting the delay time its important to consider the physical delay introduced by the geometry of the loudspeaker array, i.e. for rim positions the slapback signal should not arrive before the direct sound but should also not be too long to avoid a splitting into two separate auditory events. For the concert a delay of 80 ms was used.



**Fig. 3:** Signal flow of Slapback Delay.

### 3.4. Convolution Reverb

To further add more diffuse room information to the signals while taking advantage of the Ambisonic format a 5<sup>th</sup>-order Ambisonic convolution reverb (*mcfx convolver*) was employed. The impulse responses were achieved by upmixing a first-order Ambisonic impulse response from the openAir<sup>10</sup> library, as proposed in [17].

<sup>10</sup>freely available at <http://www.openairlib.net>

## 4. Custom Effects

The following set of custom effects was made available to the musicians themselves for individual applying and controlling. The effects can be seen as an additional tool for musical expression and emphasis.

The *Multi-Delay* and *Rotational-Delay* effects are based on feedback loops. Many DAWs do not support software feedbacks due to the risk of user-induced audio instability. Although the DAW *Reaper* supports the activation of recursive signal routing, this required excessive CPU load, causing audible clicks, so that this option was avoided. As a workaround, we implemented feedback via the audio driver. Within the *RME TotalMix* application, it was possible to create feedback loops integrated into Reaper's processing by live monitoring of the corresponding inputs. Another possible solution is to use virtual sound cards with signal routing capabilities.

### 4.1. Spatial Chorus

This effect was inspired by the legendary *Leslie Cabinet* often used with the Hammond organ, which creates a signal modulation by rotating the loudspeakers.

Here, not the loudspeakers are rotated but the signal itself: a fast dynamic spatialization spins the signal around the audience, resulting in a chorus-/tremolo-like sound. The rotation speed can be controlled via a USB expression pedal, similar to the *break*, *slow*, and *fast* settings of the Leslie speed settings. This effect works very well with sustaining sounds and rotation speeds between 720 and 5400 °/s (2 to 15 Hz).

### 4.2. Multi-Delay

Due to a large set of parallel feedback delays, the *Multi-Delay* effect creates a reverb-like sound, which envelopes the audience while preserving the clarity of the direct sound.

The signal flow is depicted in Fig. 4. It consists of 36 feedback delays with different delay times, each of them affecting one of the 36 spherical harmonics. The delay times range from 100 ms to 300 ms. The feedback attenuation is set to about 2.4 dB, which directly controls the length of the reverb. Additional filters attenuate low-/high-end content with each feedback iteration, leading to a thinning out of the signal. As a last step, the delayed harmonics get rotated by 72°.

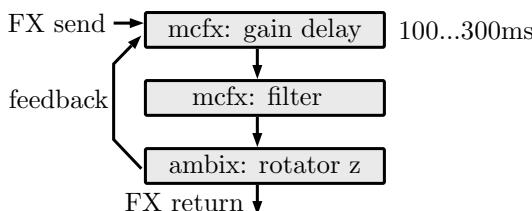


Fig. 4: Signal flow of Multi-Delay.

Due to the rotation and the application in the spherical harmonics domain the spatialization of one delayed signal changes with each iteration. This again leads to a quasi-random spatio-temporal appearance, which is typical for a diffuse reverb. In contrast to conventional reverbs, this spatialized reverb preserves the clarity of the direct sound, even at long reverberation times.

### 4.3. Rotational Delay

This effect embodies a dynamically spatialized enhancement of a simple stereo delay. Instead of being spatialized to a fixed position, each iteration of the feedback delay gets rotated clockwise (or anti-clockwise, respectively) by a fixed amount around the z-axis (see Fig. 5). The delay times and the rotation values for the two feedback delays are chosen in a way that the rotation speeds for the clockwise and anti-clockwise rotational delay are about the same: 9°/300 ms and 10°/350 ms, respectively, resulting in a rotation speed of about 30 °/s. Additionally, each iteration is filtered with a low- and high-pass causing coloration of the delays. As a result, the audience is enveloped by a stream of previously played phrases moving sideways front to back. To prevent the delays from moving to the front again, a directional filter attenuates the delayed signal once reaching the area behind the audience.

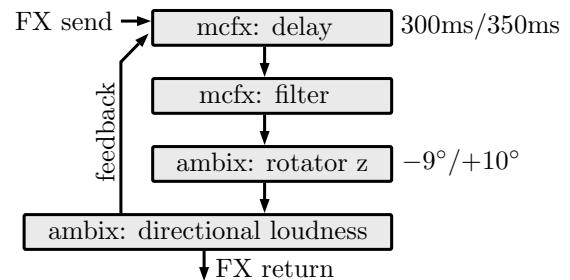


Fig. 5: Signal flow of Rotational Delay with values for the left and right delay, respectively.

## 5. Conclusions

With Al Di Meola's concert in Graz, 2016, Fig. 1, we could successfully show that *ambiX* plugins and the *Reaper* DAW allow to build spatial audio effects with high-enough throughput for low-latency performance. The prerequisite for this endeavor to become successful was to enrich the virtuous performance with acoustic instruments by an impressive 3D audio experience while preserving the genre-based sound expectations of both the musicians and the audience.

Due to the encouraging responses, we consider implementing the above-described effects into separate VST plugins for improved usability. For instance the *Multi-Delay*'s 36-channel feedback delay network can also be used as a kernel for a reverb within other standard playback formats as 5.1, 7.1 or the like. What is more, separate VST plugins circumvent the need of feedback routing within the DAW or audio driver.

## 6. Acknowledgments

Many thanks to Stefan Zaradic for initiative and idea, and for connecting us with Al Di Meola and Sennheiser Austria (Hans Freunberger, Wolfgang Guse), whose funding was gratefully acknowledged. We thank Elisabeth Freismuth for hosting the event at our university, the IEM-team (Alois Sontacchi, Stefan Warum, Vincent Ederle) and the audio and stage crew of MUMUTH for their support.

## References

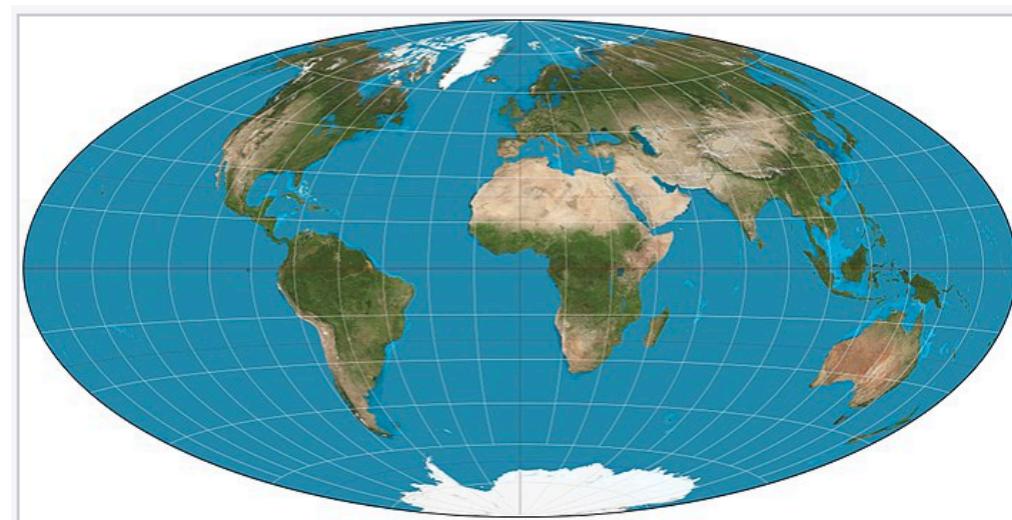
- [1] M. Frank, F. Zotter, and A. Sontacchi, "Producing 3D audio in Ambisonics," in *57th AES Conf., Hollywood CA*, 2015.
- [2] D. G. Malham and A. Myatt, "3D Sound Spatialization using Ambisonic Techniques," *Computer Music Journal*, vol. 19, no. 4, pp. 58–70, 1995.
- [3] A. Sontacchi, "Dreidimensionale Schallfeldreproduktion für Lautsprecher- und Kopfhöreranwendungen," PhD Thesis, TU Graz, 2003.
- [4] J. Daniel, J.-B. Rault, and J.-D. Polack, "Acoustic properties and perceptive implications of stereophonic phenomena," in *AES 6th Int. Conf.: Spatial Sound Reproduction*, 1999.
- [5] J. Daniel, *Représentation des champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*. Université de Paris 6: Thèse de doctorat, 2000.
- [6] J. Daniel, "Evolving views on hoa: From technological to pragmatic concerns," in *Proc. 1st Ambisonics Symposium*, 2009.
- [7] F. Zotter and M. Frank, "All-round Ambisonic panning and decoding," *Journal of the Audio Eng. Soc.*, vol. 60, no. 10, 2012.
- [8] A. J. Heller and E. M. Benjamin, "The Ambisonic decoder toolbox: Extensions for partial-coverage loudspeaker arrays," in *Linux Audio Conference, ZKM Karlsruhe*, 2014.
- [9] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," in *Linux Audio Conference, IEM Graz*, 2013.
- [10] M. Kronlachner, "Plug-in suite for mastering the production and playback in surround sound and Ambisonics," in *AES Student Design Competition 2014, Gold Award*, 2014.
- [11] P. Lecomte and P.-A. Gauthier, "Real-time 3D Ambisonics using faust, processing, pure data, and OSC," in *Proc. DAFx-15*, 2015.
- [12] T. Carpentier, M. Noisternig, and O. Warusfel, "Twenty years of ircam spat: Looking back, looking forward," in *Proc. ICMC*, 2015.
- [13] A. Politis and D. Poirier-Quinot, "JSAmbisonics: A web audio library for interactive spatial sound processing on the web," in *Proc. Interactive Audio Systems Symposium, York*, 2016.
- [14] M. Gurevic and D. D. S. Bertet, "Ambisonic spatialization for networked music performance," in *Proc. ICAD*, 2011.
- [15] M. Frank and A. Sontacchi, "Case Study on Ambisonics for Multi-Venue and Multi-Targe Concerts and Broadcasts," *J. Audio Eng. Soc*, 2017, in press.
- [16] C. Nachbar, F. Zotter, A. Sontacchi, and E. Deleflie, "ambix - suggesting an Ambisonic format," in *Proc. 3rd Ambisonics Symposium*, Lexington, Kentucky, 2011.
- [17] M. Frank and F. Zotter, "Spatial impression and directional resolution in the reproduction of reverberation," in *Fortschritte der Akustik, DAGA*, 2016.
- [18] P. Stitt, S. Bertet, and M. van Walstijn, "Off-centre localisation performance of ambisonics and HOA for large and small loudspeaker array radii," *Acta Acustica united with Acustica*, vol. 100, no. 5, 2014.
- [19] S. Tervo, J. Pätynen, A. Kuusinen, and T. Lokki, "Spatial decomposition method for room impulse responses," *J. Audio Eng. Soc*, vol. 61, no. 1/2, pp. 17–28, 2013. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=16664>
- [20] F. Melchior, U. Michaelis, and R. Steffens, "Spatial mastering - a new concept for spatial sound design in object-based audio scenes," in *ICMC*, 2011.
- [21] F. Zotter, M. Frank, M. Kronlachner, and J.-W. Choi, "Efficient phantom source widening and diffuseness in Ambisonics," in *Joint EAA Symposium on Auralization and Ambisonics*, Berlin, 2014.
- [22] M. Kronlachner, "Spatial transformations for the alteration of Ambisonic recordings," M. Thesis, University of Music and Performing Arts, Graz, Institute of Electronic Music and Acoustics, 2014.
- [23] M. Zaunschirm, M. Frank, and F. Zotter, "An interactive virtual icosahedral loudspeaker array," in *Fortschritte der Akustik, DAGA*, 2016.

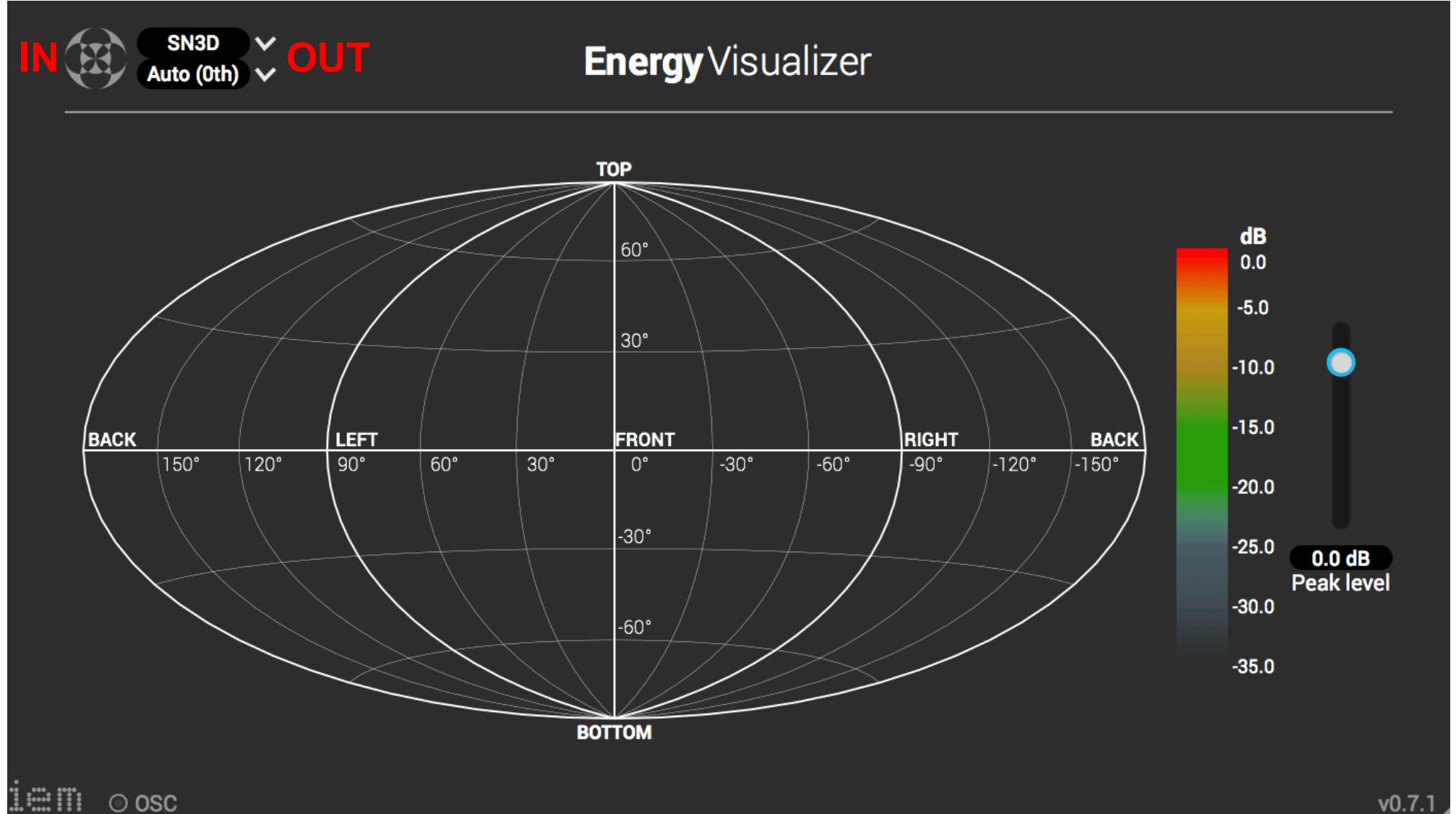
## EnergyVisualizer

Auteur Daniel Rudrich Type Visualiseur Contribution Signaux ambisoniques Sortie Signaux ambisoniques

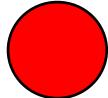
Le visualiseur d' énergie visualise la distribution d'énergie sur la sphère du signal d'entrée Ambisonic à l'aide d'une [projection de Hammer-Aitoff](#) (projection sphérique préservant la surface). La largeur de la plage dynamique visualisée est de 30 dB. La plage peut être déplacée vers le haut ou le bas à l'aide du curseur situé à côté de la projection. Les niveaux d'énergie sont codés par couleur avec une palette de couleurs à motivation perceptuelle. La palette de couleurs utilisée est illustrée à côté de la projection. Comme le signal Ambisonic est décodé à l'aide de 480 haut-parleurs virtuels avec un calcul RMS ultérieur et une visualisation OpenGL, EnergyVisualizer nécessite beaucoup de calculs. Assurez-vous de n'en utiliser que quelques-uns et désactivez-les s'ils ne sont pas nécessaires.

La **projection de Hammer** est une [projection cartographique à](#) surface égale décrite par [Ernst Hammer](#) en 1892. Utilisant la même forme extérieure elliptique 2: 1 que la [projection de Mollweide](#) , Hammer avait pour objectif de réduire la distorsion dans les régions des méridiens extérieurs, où elle est extrême Mollweide.





## FdnReverb

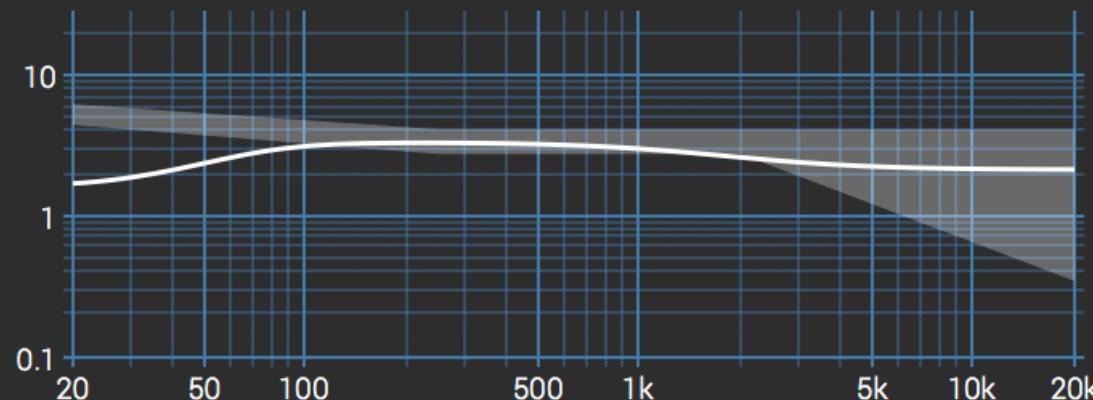


Auteur Sebastian Grill Type Réverbération Contribution Signaux audio / ambisoniques Sortie Signaux audio / ambisoniques

La réverbération **Fdn** est une réverbération à rétroaction, à délai et sur réseau. Vous pouvez l'utiliser avec des signaux audio uniques mais aussi avec des signaux Ambisonic, ce qui conduit à une réverbération diffuse spatiale et temporelle. Vous pouvez définir le temps de réverbération global et l'affiner en fonction de la fréquence à l'aide de deux filtres d'étage.

# FDNReverb

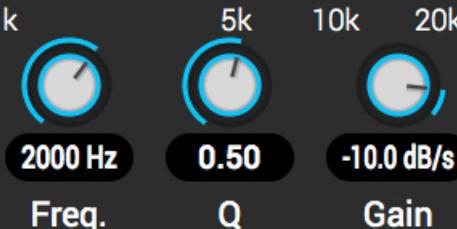
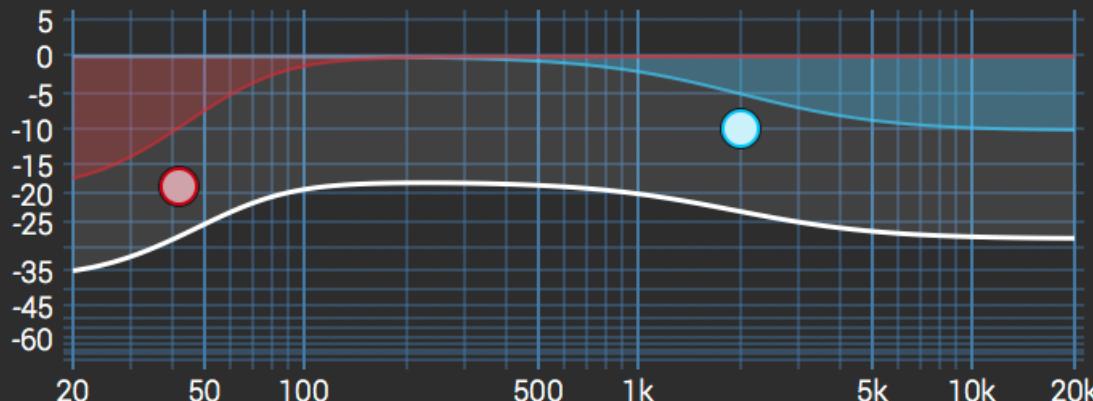
## Reverberation Time



## General Settings



## Filter Settings



# MatrixMultiplier

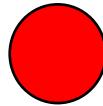
Auteur Daniel Rudrich Type Transformation de matrice Contribution Signaux audio / ambisoniques Sortie Signaux audio / ambisoniques

Avec le multiplicateur de matrice, vous pouvez charger une configuration contenant un objet TransformationMatrix (voir le [Guide des fichiers de configuration](#)) qui sera appliquée au signal d'entrée.

*Pour le Guide des fichiers de configuration  
voir le Plug-in : [Compensateur de distance](#) 6*



## MultiEncoder



Auteur Daniel Rudrich Type Codeur Contribution Signaux audio Sortie Signaux ambisoniques

Avec le **multi-** encodeur, vous pouvez encoder plusieurs sources avec un seul plug-in. Sélectionnez le nombre souhaité de sources d'entrée dans le coin supérieur gauche (jusqu'à 64). Chaque source peut être balayée, mise en sourdine et mise en solo individuellement. Le gain et la couleur de l'étiquette peuvent également être ajustés. Activez le bouton „Lock Directions“ pour laisser les sources coller au Master-Panner, qui peut être contrôlé séparément.

Double-cliquez sur la sphère pour passer à une représentation linéaire de l'altitude afin d'obtenir une résolution plus élevée à l'horizon. Chaque anneau représente 15 ° d'élévation.

4 ▼ IN MultiEncoder OUT SN3D ▼  
1st ▼

Encoder settings

#	Azimuth	Elevation	Gain
1	35.00 °	-15.00 °	0.0 dB
2	-35.00 °	-15.00 °	0.0 dB
3	120.00 °	15.00 °	0.0 dB
4	-120.00 °	15.00 °	0.0 dB

Master

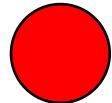
0.00 ° 0.00 ° 0.00 °  
Azimuth Elevation Roll

Lock Directions

Multicanal Quad  
Système Plug&Rec

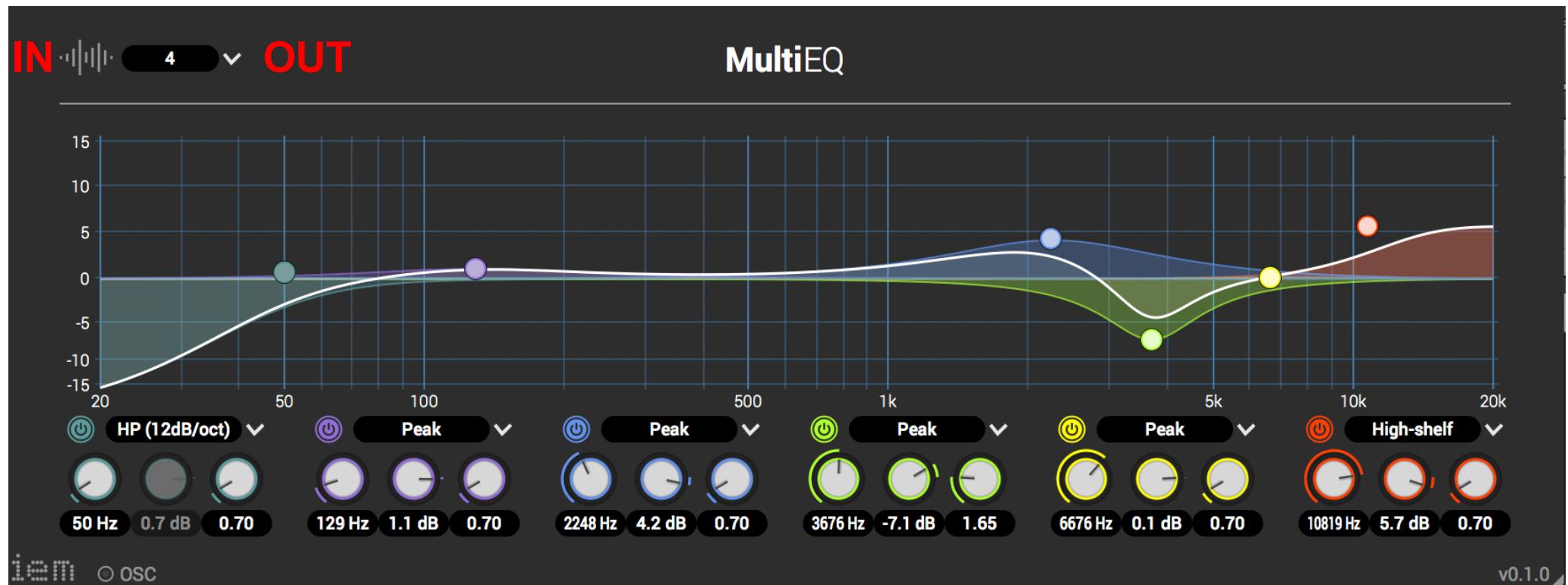
v0.5.1

## MultiEQ



Auteur Daniel Rudrich Type Filtre Contribution Signaux audio Sortie Signaux audio

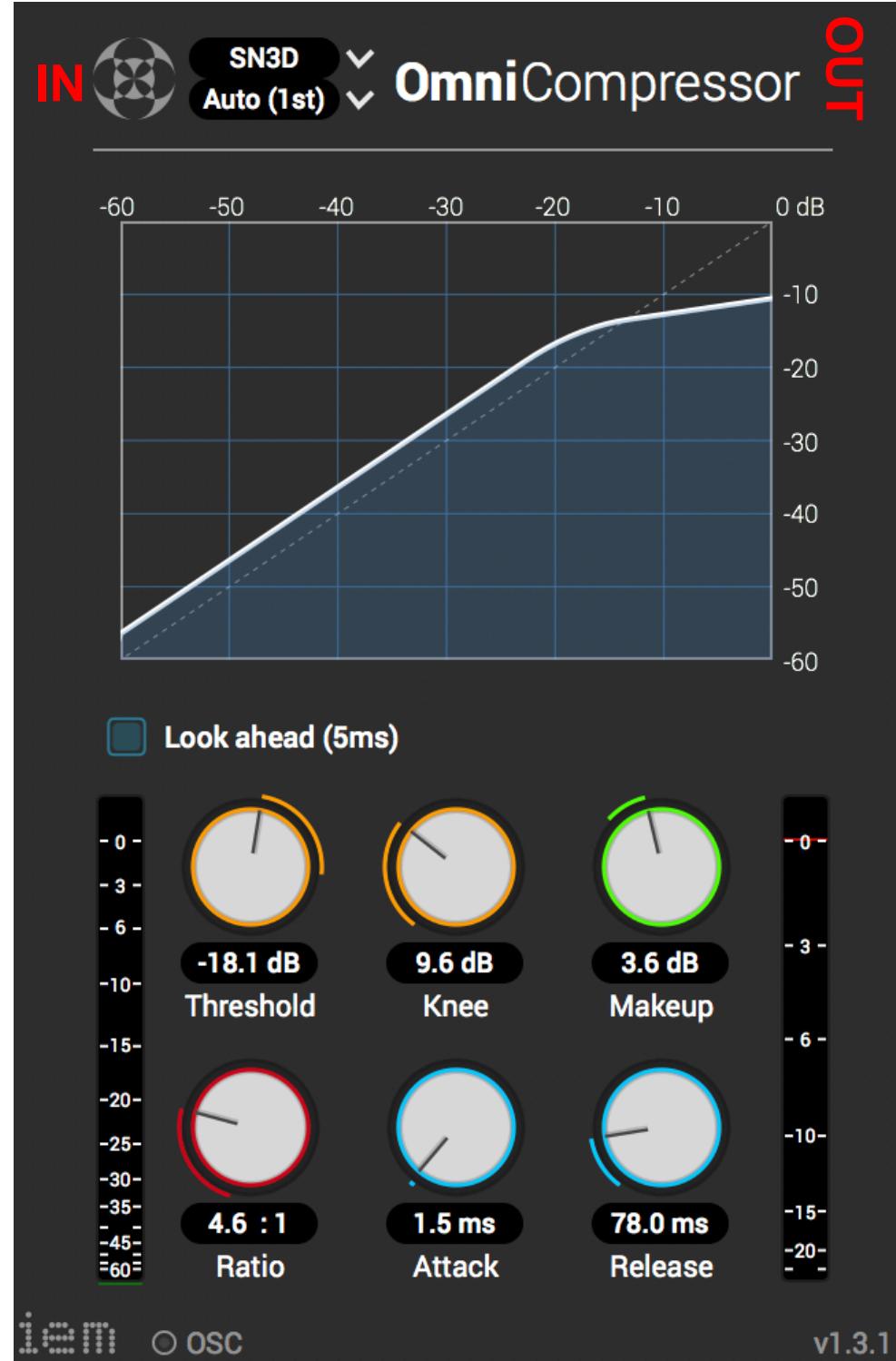
Le Multi EQ est un simple égaliseur multicanaux, filtrant jusqu'à 64 canaux audio. Chacun des six filtres peut être activé et désactivé et différents types de filtres peuvent être sélectionnés. Les premier et dernier filtres offrent des types de filtres supplémentaires: passe-bas / passe-haut de différents ordres de filtrage. Le 4ème ordre HP / LP peut être utilisé pour créer un filtre passif Linkwitz-Riley, idéal pour séparer l'audio en deux bandes, avec une réponse d'amplitude constante une fois additionnés.



## OmniCompressor

Auteur Daniel Rudrich Type Dynamique Contribution Signaux ambisoniques Sortie Signaux ambisoniques

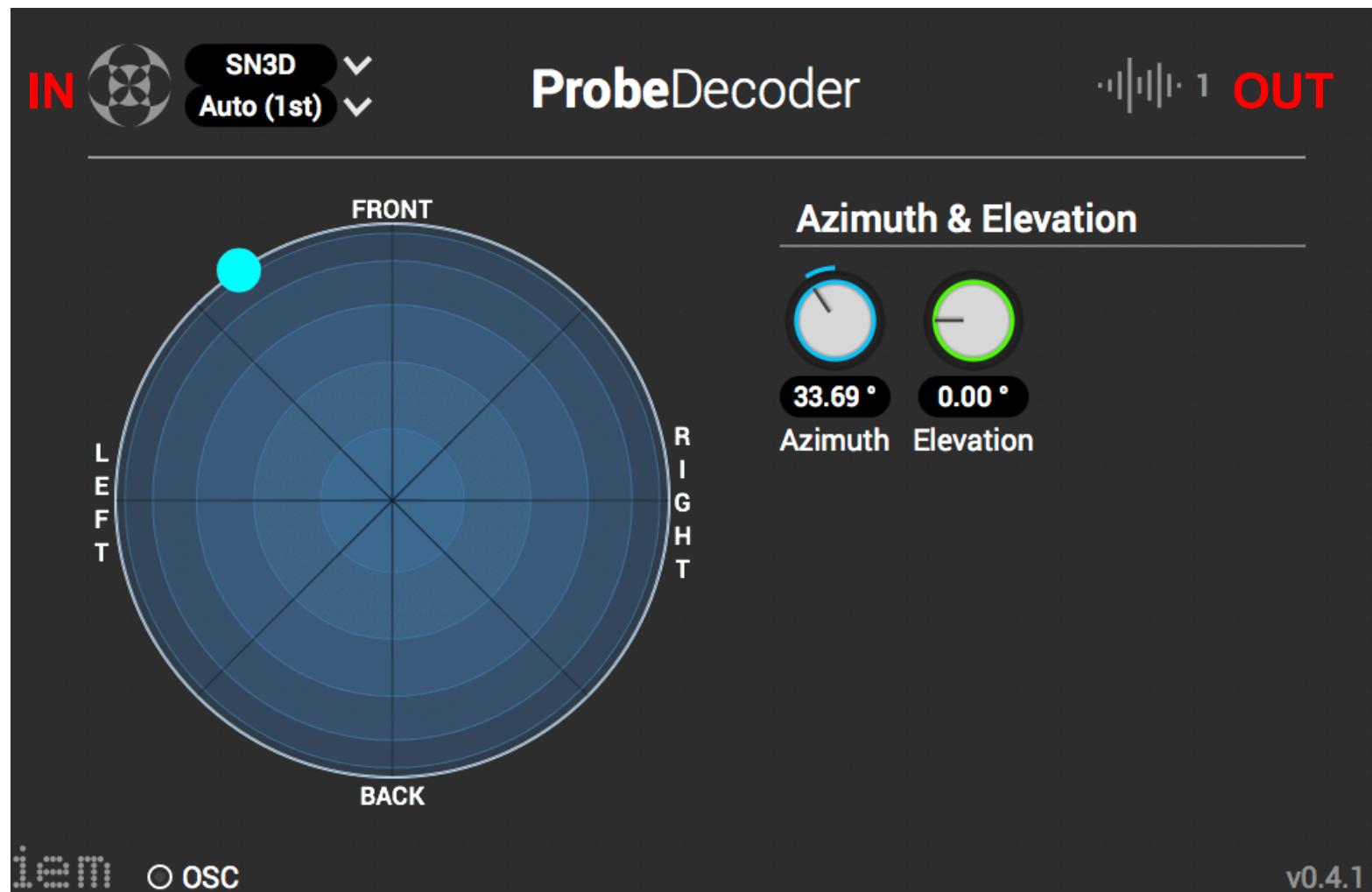
L' Omni Compressor est un compresseur Ambisonic qui peut également être utilisé comme limiteur. Cela fonctionne comme un compresseur audio ordinaire. Il utilise le premier canal d'entrée (canal W, d'où le nom "Omni") en tant que signal de commande et applique les gains calculés à l'ensemble du signal Ambisonic. Lorsque le rapport est défini à l'infini et que le temps d'attaque est défini à 0,0 ms, le compresseur agit en tant que limiteur de murs en briques. Avec ce paramètre, vous souhaiterez peut-être activer la fonctionnalité LookAead pour éviter les distorsions indésirables. Cela introduit un retard de 5 ms à votre signal. Il existe un paramètre caché (lors de la désactivation de l'interface graphique des plug-ins dans votre DAW) avec lequel vous pouvez désactiver le rapport de latence sur votre DAW pour le compenser. Cela améliore les performances de lecture de votre projet, mais le retard ne sera pas compensé par ce paramètre.



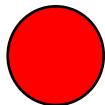
# ProbeDecoder

Auteur Daniel Rudrich Type Outil d'analyse / décodeur Contribution Signaux ambisoniques Sortie Signal audio mono

Le Probe Decoder est un petit outil permettant d'échantillonner / décoder le signal d'entrée Ambisonic dans une direction spécifique et d'écouter la sortie, principalement utilisée à des fins de test.



## RoomEncoder



Auteur Daniel Rudrich Type Encodeur + Reverb Contribution Signal audio mono ou signaux de directivité Sortie Signaux ambisoniques

Le Room Encoder est à ce jour le plug-in le plus complexe et le plus gourmand en calculs de la suite de plug-ins IEM. Grâce à elle, vous pouvez placer une source et un auditeur dans une pièce virtuelle en forme de boîte à chaussures et générer plus de 200 réflexions sur les murs. Vous pouvez déplacer librement la source et l'auditeur dans la pièce, ce qui générera automatiquement l'effet Doppler-shift. Utilisez le coefficient de réflexion global et les filtres à faible et haut degré pour modifier le caractère des réflexions.

Mais ce n'est pas tout: le RoomEncoder utilise DirectivitySignals (ou lorsqu'il est défini sur l'ordre des signaux monophoniques de l'ordre 0) comme signal d'entrée. Cela signifie que votre source peut avoir une directivité dépendant de la fréquence et variable dans le temps (par exemple, en changeant la direction de la source). Le plugin DirectivityShaper est un bon outil pour créer divers modèles de directivité.

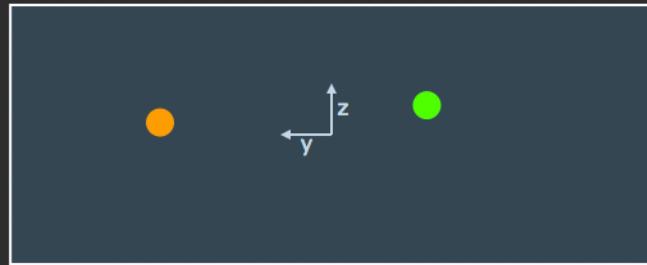
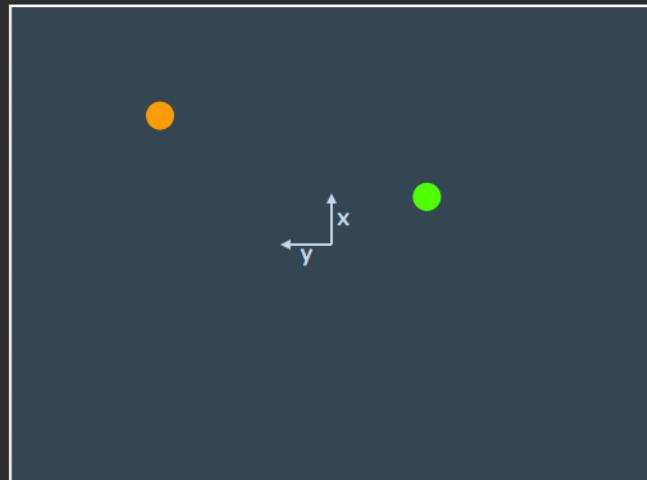
Vous pouvez synchroniser plusieurs instances de RoomEncoder pour rendre la même pièce à l'aide de l'un des quatre canaux de synchronisation. Vous pouvez également sélectionner les paramètres à synchroniser (position de l'auditeur, taille de la salle et propriétés de réflexion).



SN3D  
Auto (1st) ▾ IN

## RoomEncoder

OUT   
SN3D  
Auto (1st) ▾ 15



### Room Dimensions



9.37 m

Depth



12.62 m

Width



5.11 m

Height

### Source Position



2.541 m

x



3.381 m

y



0.262 m

z

### Listener Position



0.957 m

x



-1.840 m

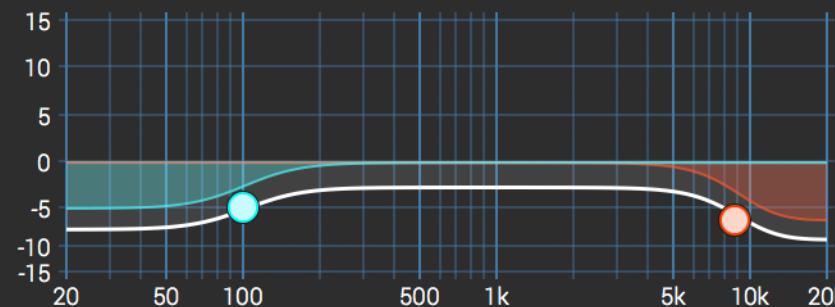
y



0.598 m

z

### Reflection Properties



-2.62 dB

Reflection Attenuation



99 Hz

Freq.



-5.0 dB

Gain



8861 Hz

Freq.



-6.4 dB

Gain



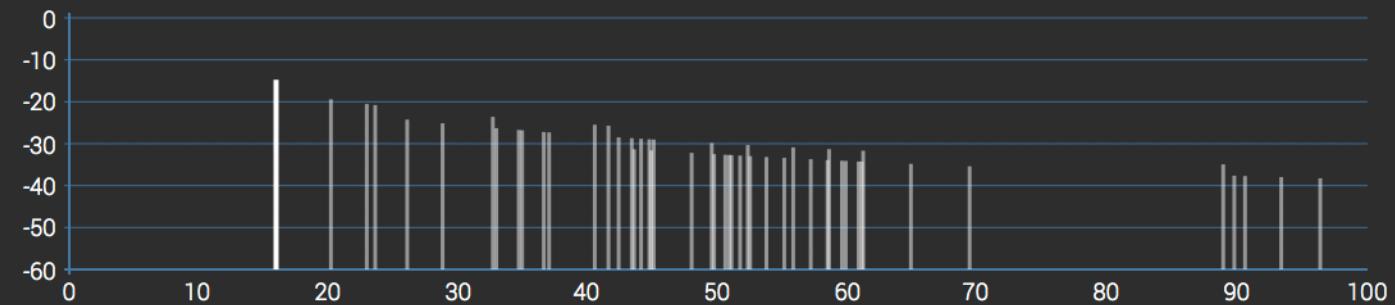
49

Number of Reflections

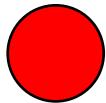
### Synchronize Room Settings

Synchronize to Channel none ▾

- Room Dimensions
- Listener Position
- Reflection Properties



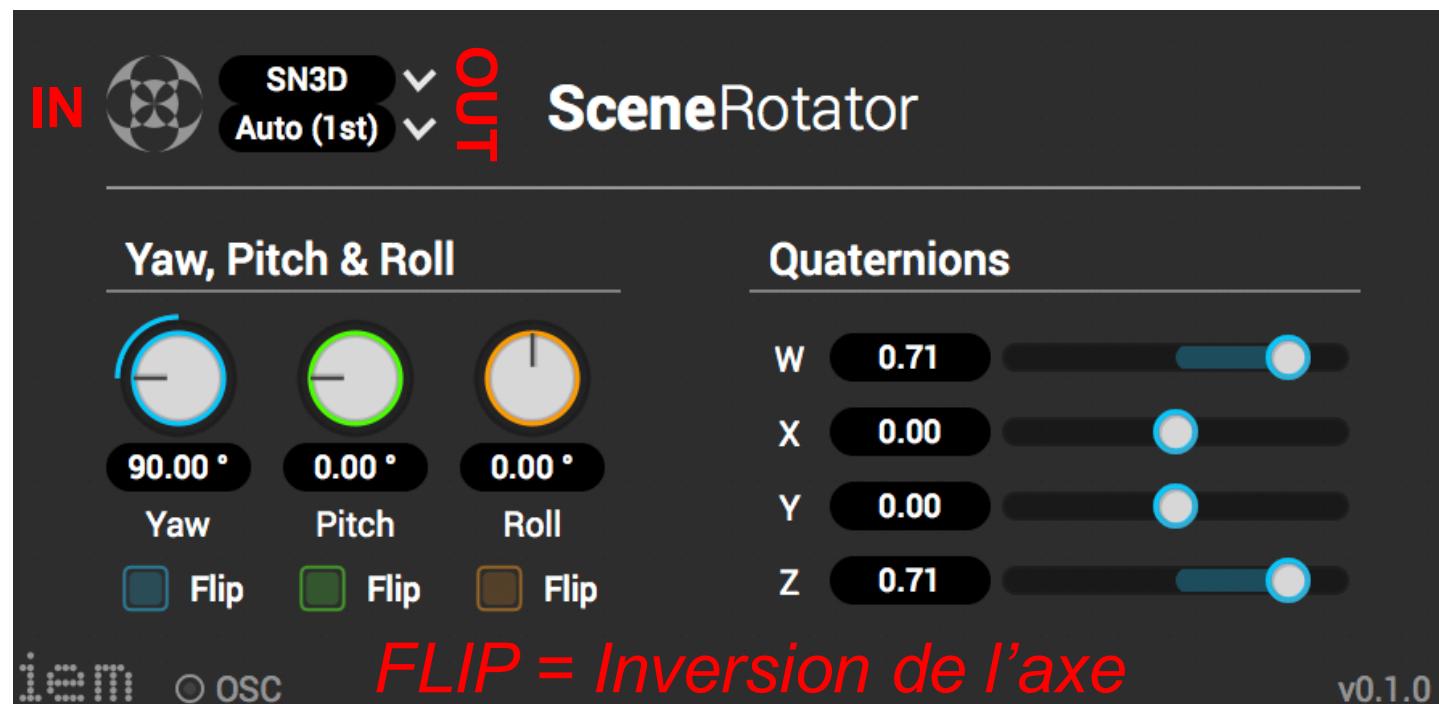
## SceneRotator



Auteur Daniel Rudrich Type Rotateur Contribution Signaux ambisoniques Sortie Signaux ambisoniques

La **scène** Rotator est un plug-in rotateurs Ambisonic, qui fait tourner la scène Ambisonic, d'où le nom! Il peut être utilisé avec les données de rotation Yaw-Pitch-Roll, ou alternativeley avec Quaternions. Chacun des trois axes de rotation peut être inversé. Les rotations se font alors dans le sens inverse. C'est très utile en combinaison avec la lecture binaurale pour compenser les mouvements de la tête.

**Remarque :** L'ordre des rotations est Roll-> Pitch-> Yaw. Il est donc parfaitement adapté à la compensation des mouvements de la tête. Un ordre de rotation sélectionnable sera ajouté bientôt!



# SimpleDecoder

Auteur Daniel Rudrich Type Décodeur Contribution Signaux ambisoniques Sortie Signaux audio

Le décodeur **simple** lit les fichiers de configuration JSON et décide le signal d'entrée Ambisonic en signaux de haut-parleur. Vous pouvez créer ces fichiers de configuration avec AllRA Decoder. Pour plus d'informations sur la structure des fichiers de configuration, consultez le [Guide des fichiers de configuration](#).

Lorsque vous réglez le mode subwoofer sur **discret** ou **virtuel**, le signal d'entrée Ambisonic est filtré sur deux bandes (en utilisant le filtre passe-haut / passe-bas du 4ème ordre). La bande supérieure est décrite en fonction du fichier de configuration chargé. Si réglé sur **discret**, le bas est envoyé au canal de subwoofer défini. Vous recevrez un avertissement, si vous choisissez un canal, qui est déjà utilisé par le décodeur. Si défini sur **virtuel**, le signal passe-bas est transmis à toutes les enceintes, créant ainsi un subwoofer *virtuel*.

The screenshot shows the SimpleDecoder configuration interface. At the top, there are input and output sections labeled "IN" and "OUT". Below these are sections for "Decoder Configuration" (with a "Load configuration" button and message "No configuration loaded."), "Frequency Bands" (a graph showing a red curve peaking at 50 Hz and a blue curve peaking at 55 Hz), and "Subwoofer" settings (mode set to "discrete", channel 1 selected). A red note in the bottom left corner points to the "Guide des fichiers de configuration".

**IN** SN3D  
Auto (0th)

**SimpleDecoder**

**OUT**

**Decoder Configuration**

Load configuration

No configuration loaded.

*Pour le Guide des fichiers de configuration, voir le Plug-in : [Compensateur de distance](#) 6*

**Frequency Bands**

20 50 100 500 1k 5k 10k 20k

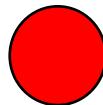
5.0 dB Gain 55 Hz Frequency 80 Hz Frequency

**Subwoofer**

discrete Subwoofer mode 1 Subwoofer Channel

iem osc v0.3.2

## StereoEncoder



Auteur Daniel Rudrich Type Codeur Contribution Signaux audio mono ou stéréo Sortie Signaux ambisoniques

Utilisez le plug-in **Stereo Encoder** pour coder des signaux audio mono ou stéréo dans le domaine Ambisonic. Dans le coin supérieur droit, vous pouvez définir l'ordre et la normalisation souhaités pour Ambisonic. Assurez-vous que la taille de votre bus est suffisamment grande. Définissez le mode automatique et le plug-in s'adapte aux modifications de la taille du bus en utilisant l'ordre le plus élevé possible.

Utilisez les curseurs *Azimuth*, *Elevation* et *Roll* pour déplacer votre source. Avec le curseur *Largeur*, vous pouvez séparer les deux canaux d'entrée. Vous pouvez bien entendu utiliser le **Stereo Encoder** également pour les sources mono. De plus, il existe une entrée quaternion que vous pouvez utiliser comme alternative à l'*azimut* et à l'*élévation*, par exemple pour les données d'orientation des suiveurs (de tête).

Utilisez les modificateurs suivants et la molette de la souris pour effectuer un panoramique avec le panoramique de la sphère:

- ⌘ / CTRL : Azimut
- ALT : Élévation
- CMD / CTRL + ALT : Rouler
- SHIFT : Largeur

Double-cliquez sur la sphère pour passer à une représentation linéaire de l'altitude afin d'obtenir une résolution plus élevée à l'horizon. Chaque anneau représente 15 ° d'élévation.

**IN** 2 **OUT**

# Stereo Encoder

SN3D  
Auto (1st)

**Azimuth, Elevation, Roll, Width**

0.00 °	-15.00 °	0.00 °	70.00 °
Azimuth	Elevation	Roll	Width

-15 ±35

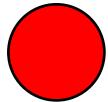
**Quaternions**

W	0.99	0.00
X	0.00	0.00
Y	0.13	0.00
Z	0.00	0.00

**Pour le XY du système Plug&Rec**

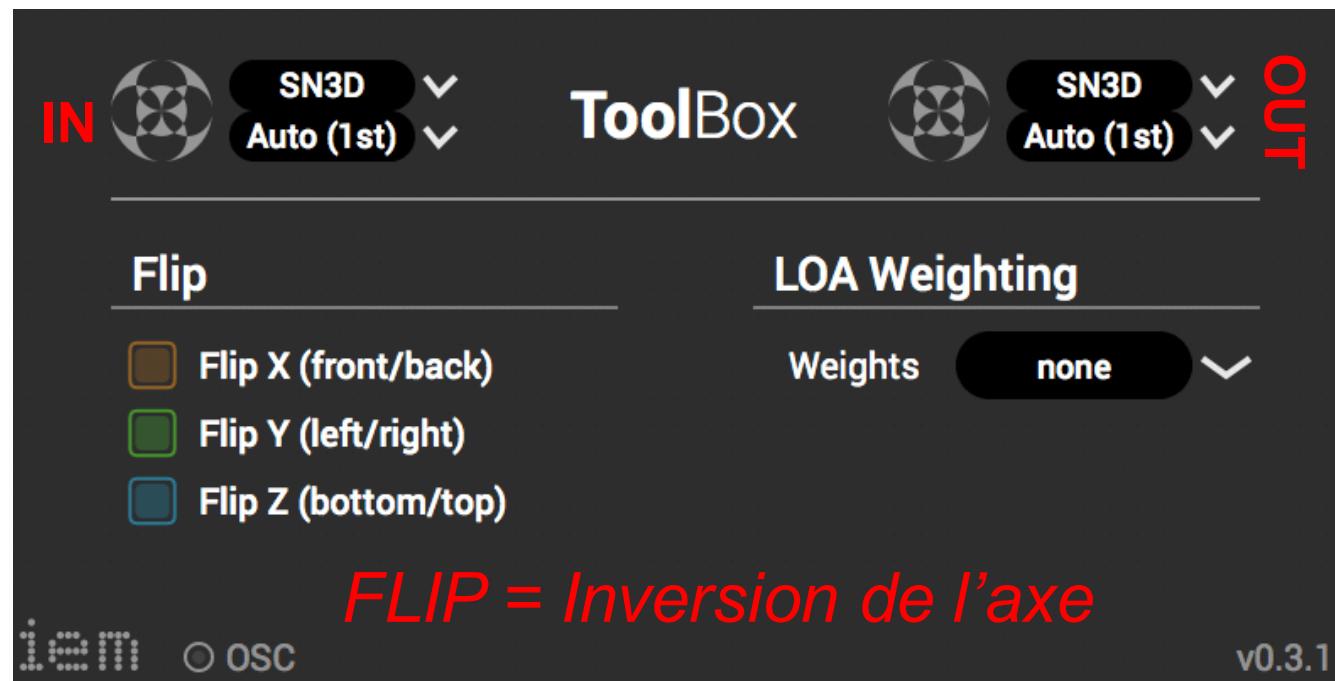
item OSC v1.3.4

## Boîte à outils



Auteur Daniel Rudrich Type Outil ambisonique Contribution Signaux ambisoniques Sortie Signaux ambisoniques

Ceci est un petit plug-in avec un nombre croissant de fonctionnalités pratiques. Pour l'instant, vous pouvez inverser le signal d'entrée Ambisonic le long des axes x, y et / ou z. Vous pouvez également l'utiliser comme adaptateur entre des plug-ins avec des paramètres de commande automatiques et des plug-ins avec des paramètres de commande fixes. La fonction de pondération LOA (Ambisonics d'ordre inférieur) vous permet de mélanger les signaux Ambisonic avec différents ordres. Exemple: utilisez la Boîte à outils avec le paramètre 1er (entrée), 5ème (sortie) et poids maximum pour un signal de microphone FOA (LOA) afin de l'inclure dans votre mix de 5ème ordre. Notez que les poids doivent correspondre à ceux du décodeur.



# Tutoriel: Routage de base dans REAPER

Ce tutoriel montre comment configurer une session Reaper pour une production Ambisonic à partir de zéro.

- 01 Ce que nous ferons et ce dont vous aurez besoin
- 02 Routage
- 03 Étape 1: Création des trois bus principaux
- 04 Étape 2: Connexion des bus
- 05 Étape 3: Ajouter une sortie matérielle pour le bus Loudspeakers
- 06 Étape 4: Ajouter des pistes source
- 07 Étape 5: Testez votre routage et enregistrez en tant que modèle
- 08 Modèles de téléchargement

## Ce que nous ferons et ce dont vous aurez besoin

Dans ce didacticiel, nous allons créer un projet de modèle REAPER HOA (Ordre Supérieur) Ambisonic, avec le routage nécessaire pour la lecture au haut-parleur et au casque. Nous allons partir de zéro avec un projet vide et le préparer pour une production Ambisonic. Nous vous recommandons de faire le routage au moins une fois pour vous familiariser avec les fonctions de routage de REAPER. Cependant, à la fin de ce didacticiel, vous trouverez des liens de téléchargement pour le projet terminé.

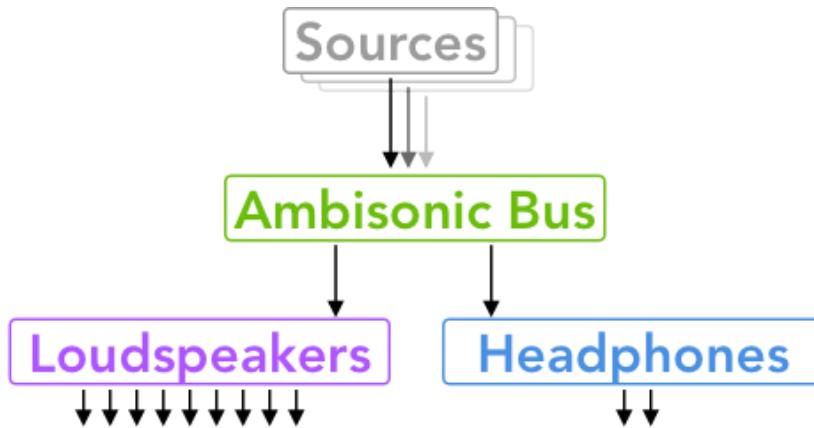
Tout ce dont vous avez besoin est:

- le DAW REAPER
- Plug-ins Ambisonic

## Routage

Parlons d'abord du routage que nous voulons réaliser. La structure de projet la plus polyvalente consiste à avoir un bus principal Ambisonic et deux bus de sortie séparés pour la lecture des haut-parleurs et des écouteurs. De cette manière, nous pouvons exporter notre mix Ambisonic, un mixage numérique basé sur les canaux vers les haut-parleurs, ainsi qu'une piste binaurale pour la lecture au casque.

Voir le schéma de routage suivant.



Nous aurons à traiter avec plusieurs sources de types différents, par exemple des fichiers Ambisonic déjà mixés ou enregistrés (par exemple Ambiences) ou des fichiers audio uniques que nous souhaitons encoder dans Ambisonics. De toute façon, nous allons obtenir un signal Ambisonic. Ces signaux sont tous acheminés vers notre Ambisonic Bus, où nous pouvons, par exemple, appliquer une compression douce pour tout coller ensemble et / ou utiliser **Energy Visualizer** pour voir ce qui se passe. Lorsque nous rendons / rebondissons / exportons ce bus, nous obtenons un fichier audio Ambisonic multicanal, que nous pouvons envoyer partout dans le monde, que les gens décodent et écoutent. Quel temps merveilleux nous vivons!

Mais bien sûr, nous voulons aussi écouter ce que nous créons. C'est pourquoi nous dirigeons notre bus principal Ambisonic vers un Loudspeakers bus où se déroule le décodage des haut-parleurs. Ces signaux décodés peuvent ensuite être envoyés à notre matériel audio pour les lire sur des haut-parleurs.

Au cas où nous n'aurions pas autant de haut-parleurs disponibles, peut-être pendant un trajet en train, nous pourrions écouter notre mix Ambisonic avec des écouteurs. Par conséquent, le même signal Ambisonic est acheminé vers une distincte Headphones piste, où nous les décoder avec le **Binaural** décodeur aux signaux de casque binaural. Vous pouvez bien entendu utiliser le décodeur binaural de votre choix!

## Étape 1: Création des trois bus principaux

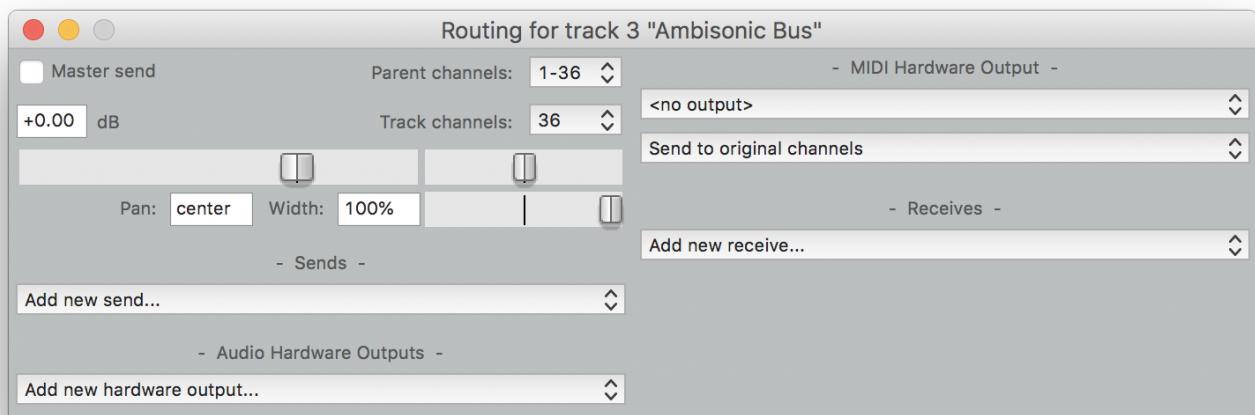
Ouvrons un nouveau projet dans REAPER et créer ces trois voies principales: Ambisonic Bus, Loudspeakers et Headphones . Double-cliquez simplement sur la zone vide de la vue du mélangeur:



Par défaut, REAPER crée des pistes stéréo, qui seront automatiquement routées vers le MASTER bus. Nous ne voulons pas de cela alors changeons cela: Cliquez sur le bouton Routage pour ouvrir la boîte de dialogue de routage du canal. Désactivez la case à cocher "Envoi principal" et réglez le canal sur la taille nécessaire pour votre commande Ambisonic souhaitée. Voici un peu d'aide:

1er -> 4  
2e -> 9  
3ème -> 16  
4ème -> 25  
5ème -> 36  
6ème -> 49  
7ème -> 64

Dans la capture d'écran suivante, vous pouvez voir que les bus ont été préparés pour Ambisonics de 5ème ordre.



Vous pouvez activer le boîtier de commande «Envoi principal» pour le Headphones bus afin que le son binaural soit envoyé à la piste MASTER et directement à la sortie stéréo principale de votre carte son.

## Étape 2: Connexion des bus

Faisons donc un routage réel. REAPER a une fonctionnalité de routage très soignée: vous pouvez simplement faire glisser le bouton de routage d'une piste vers celui d'une autre piste et la sortie de la première sera routée vers l'entrée de la seconde. Commençons donc par acheminer la sortie de Ambisonic Bus vers l'entrée du

Loudspeakers bus.



Lorsque la fenêtre contextuelle de routage s'ouvre, n'oubliez pas de sélectionner *Source multicanal* -> 1-36 afin que les 36 canaux (ou l'ordre que vous utilisez) soient routés. Répétez le même processus avec *Ambisonic Bus* et *Headphones*.

Vous avez peut-être déjà remarqué que j'ai affecté des couleurs aux trois bus conformément à notre diagramme de routage ci-dessus. J'ai également déjà inséré des plug-ins utiles dans chacun des trois bus:

- **Ambisonic Bus** : **Energy Visualizer**, pour voir ce que nous sommes censés entendre.
- **Loudspeakers** : **Simple Decoder**, pour la lecture de nos haut-parleurs.
- **Headphones** : **Binaural** décodeur, bien ..., vous savez ... pour le décodage binaural.

## Étape 3: Ajouter une sortie matérielle pour le Loudspeakers bus

Lorsque nous ouvrons la *boîte de dialogue de routage* du *Loudspeakers bus*, nous pouvons ajouter une sortie matérielle. Cliquez simplement sur le premier canal de la sortie matérielle de votre choix. Pour envoyer autant de canaux que vous avez de haut-parleurs, cliquez également ici en bas à gauche pour attribuer à une *source multicanal* le nombre de haut-parleurs.

## Étape 4: Ajouter des pistes source

Nous avons déjà préparé le chemin depuis notre *Ambisonic Bus* monde vers des membranes en mouvement et une pression acoustique en évolution rapide. Nous devons encore obtenir de l'audio! C'est ce que nous faisons maintenant.

Ajoutez de nouvelles pistes en double-cliquant dans une zone vide sous nos principaux bus. REAPER a une fonctionnalité de *regroupement* très pratique . Il suffit de sélectionner les pistes nouvellement ajoutées et de les faire glisser sur le *Ambisonic Bus* . Leurs sorties y seront automatiquement acheminées (avec l'*envoi par le parent* activé, ce qui est le comportement par défaut). N'oubliez pas de définir la taille de chaque piste en fonction de votre commande Ambisonic souhaitée!



## Étape 5: Testez votre routage et enregistrez en tant que modèle

Il est temps de faire glisser du son dans vos pistes, de le coder pour Ambisonics et de tester votre routage! N'oubliez pas de désactiver le bus Headphones ou le Loudspeakers bus, sauf si vous souhaitez utiliser les deux simultanément. Une fois que vous êtes satisfait du résultat, nous vous recommandons d'enregistrer ce projet en tant que modèle: *Fichier -> Modèle de projet -> Enregistrer le projet en tant que modèle...*. Donnez-lui un nom explicite, par exemple "Ambisonic Production o5". Pour les nouvelles productions Ambisonic, vous pouvez charger rapidement ce modèle et vous êtes prêt à partir!

## Modèles de téléchargement

Vous pouvez télécharger ici les modèles REAPER pour les productions Ambisonic des 3ème, 5ème et 7ème ordres.  
[Ambisonic Production Templates.zip](#)

N'oubliez pas d'ajouter une sortie matérielle sur le Loudspeakers bus.

---

Plug-in Suite de IEM - Institut de musique électronique et d'acoustique

[IEM.at](#)

[Docs](#)

Site créé avec [Hugo](#) sur le thème du [kube](#).

This file only contains the major changes of the IEM Plug-in Suite.  
For a list of all changes and bugfixes see the git log.

Please note, that breaking changes are marked with `!!BREAKING CHANGE!!`. They might lead to an unexpected behaviour and might not be compatible with your previous projects without making some adaptions. See the [Breaking changes article](<https://pluginsием.at/docs/breakingchanges/>) for more information.

## v1.9.0

- general changes

- plug-in host will be notified when parameter values change automatically (e.g. conversions)
- osc port can be changed on the fly when opened, plug-ins will close and re-open port immediately

- plug-in specific changes

- **Binaural\*\*Decoder**
  - output level reduced by 10dB
- **Distance\*\*Compensator**
  - fixes broken import of loudspeaker layouts
- **Energy\*\*Visualizer**
  - levels will only be calculated when the GUI is open
  - adds control to change visualized dynamic range
- **Scene\*\*Rotator**
  - adds quaternion inversion
  - adds selectable sequence of rotations

## v1.8.1

- plug-in specific changes

- **AllRA\*\*Decoder**
  - fixes: AllRADecoder forgetting layout when loading previous state

## v1.8.0

- general changes

- refactored widgets (internal change)

- new plug-ins:

- **Multi\*\*EQ**
  - equalizer plug-in with several filter types, filtering up to 64 audio channels
- **Scene\*\*Rotator**
  - simple rotator of Ambisonic scenes

- plug-in specific changes

- **Fdn\*\*Reverb**
  - mouse-wheel controls Q factors of filters within the FilterVisualizer

## v1.7.1

- general changes

- smaller GUI re-draw improvements
- support for GenericLayout in configuration files

- plug-in specific changes

- **Room\*\*Encoder**
  - fixes possible crashes when host's buffersize is smaller than reported (can happen with



<https://www.lesonbinaural.fr>

# Encodeur B360 Ambisonics de chez Waves

**299 \$ ??**

<https://www.waves.com/plugins/b360-ambisonics-encoder>

# Encodeur B360 Ambisonics

MAINTENANT 299  
\$

AJOUTER AU  
PANIER

★★★★★ 1 Obtenez une démo

Mises à niveau

Liste de souhaits

Également disponible: [Outils Ambisonics à 360 °](#), combinant B360 et Nx pour surveiller Ambisonics sur un casque stéréo.

Convertissez et mixez des pistes mono, stéréo et surround au format Ambisonics B pour créer des mixages audio à 360 ° pour les projets YouTube 360, Facebook 360 et VR.

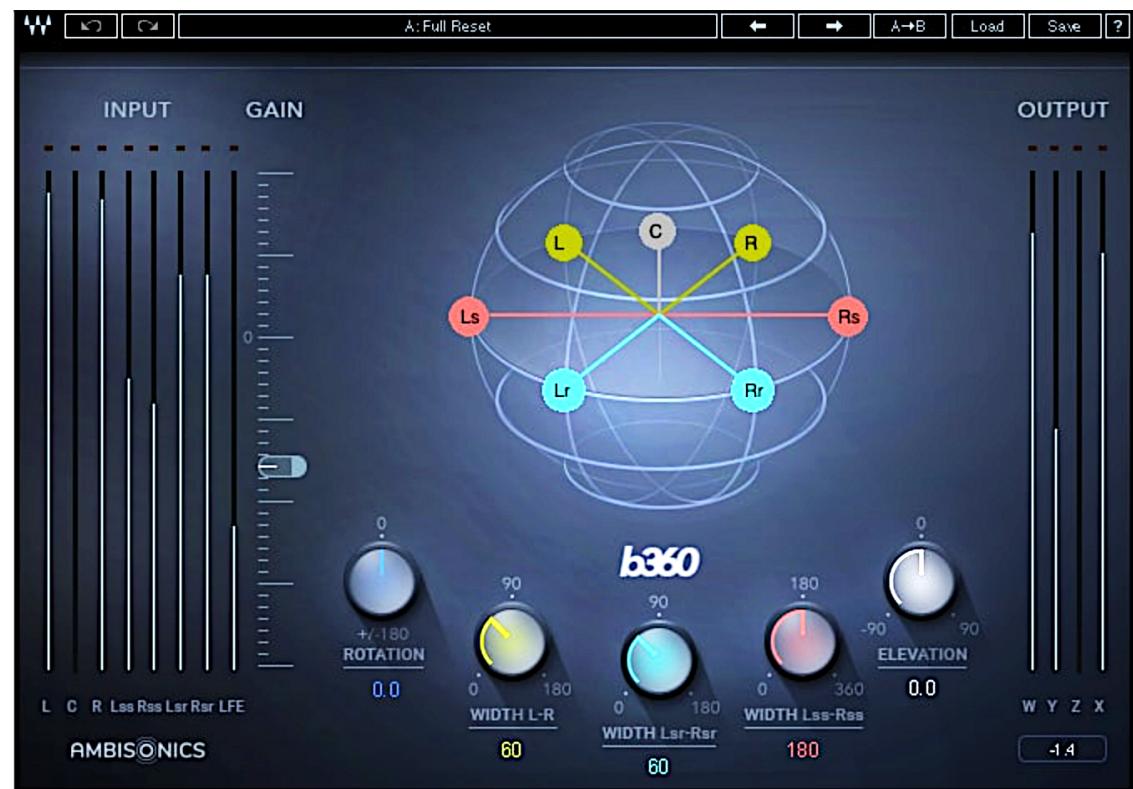
Ambisonics B-format est le standard de l'industrie pour le mixage audio dans un champ sonore sphérique à 360 degrés. Contrairement aux pistes mono, stéréo ou surround standard, qui acheminent des haut-parleurs individuels, le format B dispose de quatre canaux qui représentent l'ensemble du champ sonore sphérique. Lorsque vous souhaitez inclure des pistes mono, stéréo ou surround dans un mix Ambisonics, vous devez d'abord les convertir au format B.

Le codeur B360 Ambisonics vous permet de le faire via deux flux de travail de base:

- ▶ Vous pouvez insérer le B360 sur l'ensemble de votre mix stéréo ou surround et le fichier original sera converti au format Ambisonics.
- ▶ Ou vous pouvez insérer B360 sur chaque canal de votre session, pour un positionnement précis de ce canal dans votre mix Ambisonics.

Dans les deux cas, le B360 vous permet de contrôler la largeur, l'élévation et la rotation du mixage, ainsi que de positionner les différents éléments de votre mix dans le champ sonore tridimensionnel.

Avec une interface graphique intuitive et des commandes rationalisées, le B360 est conçu pour rendre votre flux de travail audio Ambisonics plus



- ▶ Le moyen standard de l'industrie pour diffuser de l'audio pour les vidéos VR et 360 °
- ▶ Convertissez les formats mono, stéréo et surround au format B Ambisonics de premier ordre
- ▶ Positionnez précisément les éléments dans votre mix audio à 360 °
- ▶ Interface intuitive; s'inscrit dans votre flux de travail de mélange traditionnel
- ▶ Contrôles de largeur, de rotation et d'élévation
- ▶ Compatible AmbiX
- ▶ Inclut les convertisseurs d'utilitaire AmbiX-à-FuMa et FuMa-à-AmbiX



# B360 Ambisonics Encoder

## User Guide



# Waves B360 Ambisonics Encoder

## User Guide

Welcome .....	3
Chapter 1 Introduction .....	3
<i>What is Ambisonics?</i> .....	4
Chapter 2 Getting Started .....	5
Chapter 3 Components .....	7
<i>Ambisonics Formats: FuMa and AmbiX</i> .....	8
Chapter 4 Signal Flow Examples .....	10
Chapter 5 Controls and Displays .....	12
<i>Controls</i> .....	13
<i>Displays</i> .....	15



# Welcome

Thank you for choosing Waves! In order to get the most out of your B360 Ambisonics Processor, please take some time to read this user guide. We also suggest that you become familiar with Waves Support at [www.waves.com/support](http://www.waves.com/support). There you will find an extensive answer base, the latest tech specs, detailed installation guides, new software updates, and current information on licensing and registration.

Sign up at the support site and you will receive personalized information about your registered products, reminders when updates are available, and information on your authorization status.

## Chapter 1 Introduction.

Full-immersion 3D surround sound is here to stay; it's long past the "gee whiz" stage. 360 Video on social media, installations, and VR game engines have become 360-degree surround sound dependent. Therefore, there's a need for surround content. There are several ways to create a 3-D surround soundscape. Some solutions are direction-dependent: their surround image is solid when facing in a specified direction, but falls apart when facing other directions. Other solutions rely on bussing discrete signals to loudspeakers within the listening space, which is interesting but limited.

This is why content creators are turning more and more to a standard that is not dependent on the number of output channels or loudspeaker placement and is solid in all directions. This standard is called Ambisonics: a four-channel encoded signal that contains all the information needed to reproduce a complete sphere of sound. Since an ambisonic signal is encoded, it cannot be monitored directly. It must be decoded with the Nx Ambisonics Virtual Mix room or other Ambisonics decoder.

Waves B360 converts standard audio tracks to B-format Ambisonics. This lets you mix with standard audio formats (from mono to 7.1) and convert the mix to a four-channel Ambisonics format. You can also mix ambisonic tracks with regular mix elements, which opens more sound design possibilities. B360 offers precise panning in up to seven image sources. The mix can be monitored via Nx with fully integrated head tracking, which helps you create very accurate, point-of-view reproduction of the sound sphere.



## What is Ambisonics?

Ambisonics is a long-established standard for recording, mixing, distributing, and playing 360-degree sound audio. The entire sound image of a space (360-degrees left and right, up and down) can be captured and encoded as a four-channel stream.

Ambisonics differs from classic surround sound technologies such as Quad, Dolby Digital, SDDS, DTS, and others since these formats create a surround image by sending specific mixes to specific loudspeakers. They can provide good imaging when static, but as the sound field rotates, the sound tends to “jump” from one speaker to another. Also, dedicated speaker systems are usually “front biased”: information from the side or rear speakers is not as focused as the sound from the front. Ambisonics instead treats a soundscape as an uninterrupted sphere of sound around a center point. This center point is where the microphone is placed while recording and where the listener’s sweet spot is located while playing back. Ambisonic channels cannot be fed directly to speakers; they must be decoded for playback.

Ambisonics captures a full spherical sound image with four microphone capsules in a tetrahedral array. The raw signal from these microphones is encoded to the four-channel “Ambisonics B-format,” which is the format used while mixing and monitoring.

There are two conventions within the B-format Ambisonics standard: FuMa and AmbiX. They are quite similar, but not interchangeable. Waves Ambisonics plugins use the AmbiX format. Ambisonic files that are FuMa B-format can be converted with the enclosed format-changing plugins, which are described in Chapter 3. Below is the B-format AmbiX channel sequence.

- 1 W Omnidirectional polar pattern. It contains all sounds in the sphere, from all directions at equal gain and phase.
- 2 Y Figure-of-8 bi-directional polar pattern pointing to the left
- 3 Z Figure-of-8 bi-directional polar pattern pointing up
- 4 X Figure-of-8 bi-directional polar pattern pointing forward

Each of the three figure-of-8 microphones has a positive side and a negative (invert phase) side. Phase differences between the four channels as well as gain differences, provide the sound information necessary to completely describe the spherical sound space. Therefore, it’s very important that all B-format channels are processed together.



## Chapter 2 Getting Started

The B360 plugin is used to transform standard formats into a four-channel Ambisonics B-format audio stream. You will use it primarily to create ambisonic mixes while monitoring through the Nx Ambisonics headphones, with or without head tracking.

### **Scenario 1: Setting up a session for Mixing Ambisonics and Monitoring (with Nx Ambisonics)**

For the most part, mixing for Ambisonics is the same as conventional mixing; only controlling imaging and monitoring are different.

1. Insert B360 on each playback track of your session. The B360 will open the correct component based on the channel type (mono, stereo, 7.1, etc.). Note: when working with Cubase or Nuendo, you must create a track of four channels or larger, even when the source material is mono or stereo. The B360 must be the last insert on each playback track. You cannot perform any processing downstream of the B360.
2. Add plugins to tracks as you like. EQ, dynamic, and noise reduction plugins will not interfere with the Ambisonics process, as long as they are placed before B360 in the processing chain. Do not use panners or imaging processors at this time.
3. Create a Quad track. This can be the mix master or a monitor buss. Route each of the playback tracks to this buss. This is your ambisonic mix track.
4. Insert an Nx Ambisonics plugin on this track. This enables Ambisonics monitoring, along with head tracking. Adjust Nx to suit your listening preferences. You can use the Nx plugin without head tracking, although the spherical sound experience is more satisfying with head tracking. Please refer to the Nx Virtual Mix Room user guide for details. You are now ready to monitor the mix.
5. Mix your project as you normally do. Use as many plugins as needed. Control the track's width, rotation, and elevation with the B360 images tools. Remember, B360 must be the last insert in each playback track.

Ambisonics B-format files require specialized tools for processing. Since phase is a critical component, the four channels must remain precisely time- and gain-aligned during processing.



## **Scenario 2: Adding new mix elements to an existing ambisonic mix**

You can add new, non-ambisonic elements to one (or more) existing ambisonic mixes. The mixing setup is the same as described above, except that the original ambisonic track now becomes a playback element. The new sounds are mixed against this track. Route the existing ambisonic track and the new added mix elements to the same four-channel master buss.

This technique is useful for creating a customized or localized version of a project, or for adding new details to a previously mixed ambisonic soundscape. For example, imagine you are creating a local language version of a VR game. The atmosphere and detail SFX are already in place in the ambisonic mix. Record, edit, and process the voices as you like and then place them in the mix using the B360 imagers. The new voice elements will fall precisely where you want them to be and they will sit in the mix quite comfortably.



## Chapter 3 Components

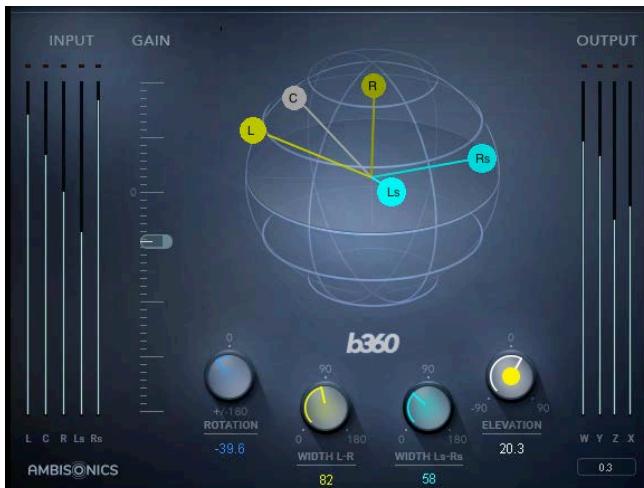
There are five B360 components. The format (number of channels) of the source track determines which component will be loaded. This sets the number and type of imaging controls.



Mono-to-Ambisonic



Stereo-to-Ambisonic



5.0-to-Ambisonic



5.1-to-Ambisonic





7.1-to-Ambisonics

## Ambisonics Formats: FuMa and AmbiX

There are two utility plugins that convert between AmbiX and FuMa Ambisonics formats. These plugins are for conversion only; there are no controls.



Ambisonics FuMa-to-AmbiX converter



Ambisonics AmbiX-toFuMa converter



## Why two B-format standards?

Ambisonics was developed in the 1960s by mathematicians and physicists. It was not created by industry or standardized by committees. It was born of devotees of math and music who set out to make a better way to reproduce multi-dimensional surround sound. They succeeded, but as is often the case, great solutions are not guaranteed commercial success. For many years, Ambisonics remained a favorite of a niche market, but was largely unnoticed by the larger market. Now, it's gaining traction as a viable format for fully immersive surround sound.

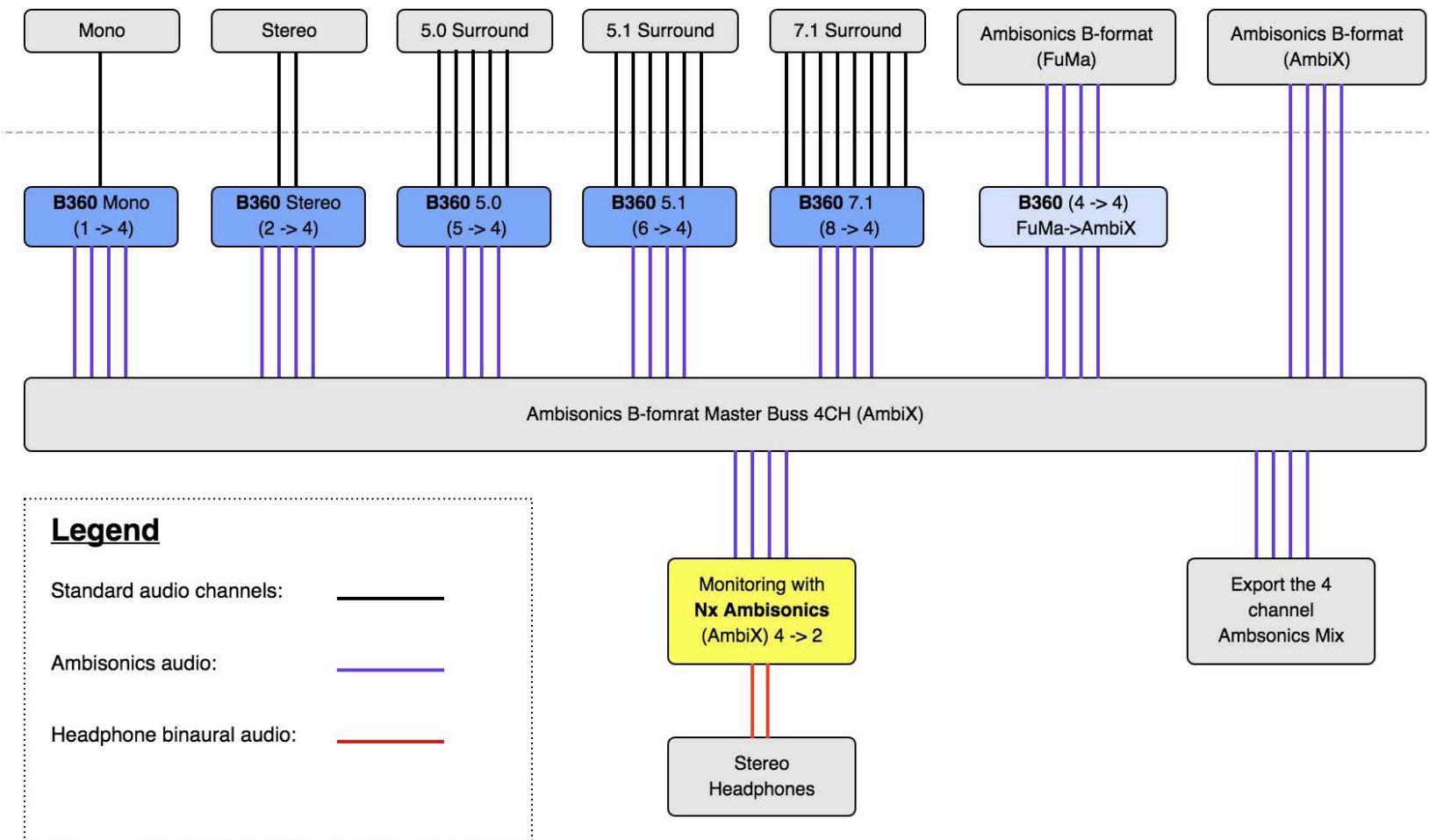
Since Ambisonics has never been a standard per se, the rules of how to use it were never completely established. The math is the math, but the definition of the B-format stream was not formalized. There are now two commercially used Ambisonics definitions: FuMa and AmbiX. The two are essentially the same—only track sequence and levels differ. B360 uses the AmbiX. So if you begin with an AmbiX format and you will distribute in AmbiX, then you don't need to make any conversions on either end of the process. But if you need to convert between FuMa and AmbiX, you must use the appropriate conversion tool (as shown above).



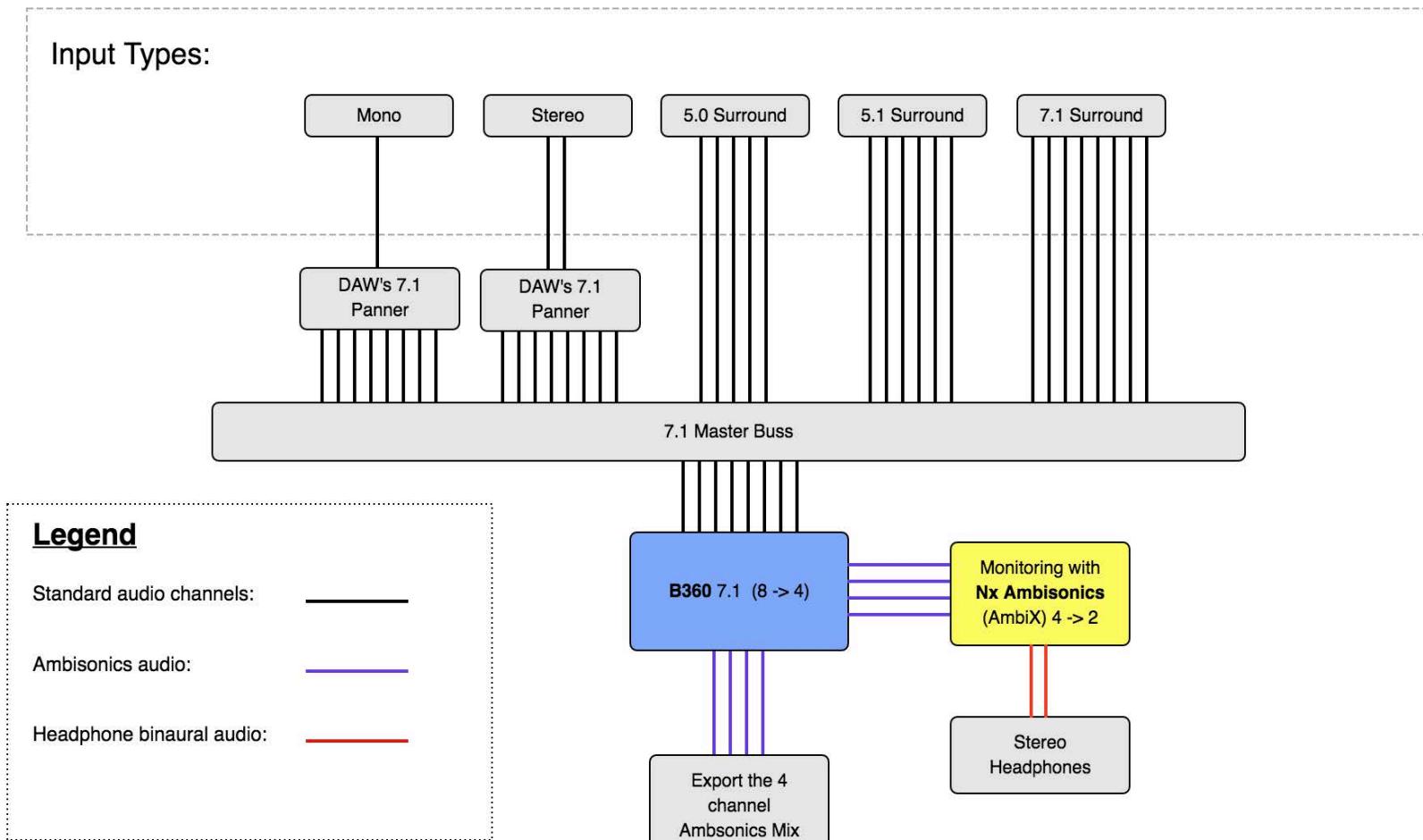
## Chapter 4 Signal Flow Examples

1. Use B360 as an Ambisonics panner on individual tracks. Monitor with Nx Ambisonics.

Input Types:



2. Use B360 as an Ambisonics encoder on an entire 5.1 or 7.1 mix buss. Monitor with Nx Ambisonics.



## Chapter 5 Controls and Displays

The B360 user interface is scaled. The number of controls is determined by the format of the playback track and the logic of each control remains the same no matter the size of the component.



## *Controls*

### **Input Fader**

Controls input level to the plugin. Ambisonics processing may increase the signal level. If the output meter indicates clipping as you pan the signal, reduce the input level.

Range: -24 dB – +12 dB

Reset: 0 dB

### **Rotation**

Rotates the elements of the track together. Width and elevation settings are maintained while rotating.

Range:  $\pm 180^\circ$ , continuous

Default and Reset:  $0^\circ$

### **Elevation**

Moves all of the elements up or down on the sound sphere.

Range:  $\pm 90^\circ$

Default and Reset:  $0^\circ$

### **Width L-R**

Controls the separation between the L and R channels.

Range:  $0\text{--}180^\circ$

Default and Reset:  $60^\circ$

### **Width Lsr-Rsr (7.1 component)**

Controls the width of the rear surround channel.

Range:  $0^\circ\text{--}180^\circ$

Default and Reset:  $60^\circ$



**Width Ls-Rs** (mono, stereo, 5.0 and 5.1 components)

Controls the width of the surround channels.

Range: 0°–180°

Default and Reset: 60°

**Width Lss-Rss** (7.1 only)

Controls the width of the side surround channel.

Range: 0°–360°

Default and Reset: 180°

Alt/Opt+Click on any control to reset it.



## *Displays*

There are three display sections:

### **Input Meters**

Displays the level of each channel of the input signal. The name of each channel is displayed below.

Range: Full scale, -infinity to 0 dBFS

### **Output Meters**

Output display of the output B-format ambisonic stream.

Channels: W, X, Y, Z

Range: Full scale, -infinity to 0 dBFS

Peak indicator (beneath the output meters): Indicates the peak level of the loudest signal

Click on the indicator to clear the value.

Clipping indicator: Turns red when an output channel exceeds 0 dBFS.

Click on a clip light to clear it.

### **Sound Sphere**

This display represents the channel's imaging within the spherical Ambisonics soundscape as experienced from the listening position. Small, color-coded globes indicate control settings.

