

institut für elektronische musik und akustik



## **IEM Report 28/05**

### **3D Sound-Mixer**

#### **Implementation eines kombinierten Ambisonic- und Bus-Mixers für den Einsatz in 3D Audio Environments**

#### **Autoren:**

**Winfried Ritsch, Thomas Musil, Johannes Zmólnig, Robert Höldrich**

Version 1.0, Stand 31.12.2005

## **Abstract**

Der „3D-Sound Mixer“ versteht Klang als räumliches, skulpturales Phänomen und zielt auf die Entwicklung eines universell einsetzbaren Software-Tools, das Klang in seiner räumlichen Verteilung steuern kann und dabei auf ein Kodierungsformat setzt, das unabhängig von der konkreten Wiedergabesituation – vom Kopfhörer über die Surround-Anlage im Wohnzimmer bis zu vielkanaligen Kuppelbeschallungen in großen Theatern – ist. Im vorliegenden Dokument wird die in Pure-Data implementierte Software beschrieben.

# Inhaltsverzeichnis

1. Einleitung .....	4
2. Architektur des 3D-Sound Mixer .....	7
2.1 Hardware-Architektur .....	7
Remote Control .....	9
2.2 Software-Architektur .....	9
2.2.1 Programmstruktur .....	9
2.2.2 Aufbau des Mixers .....	10
2.2.3 Kanalzug .....	11
2.2.4 Master-Sektion .....	14
2.2.5 Systemsteuerung .....	16
2.2.6 Plugins .....	16
2.2.7 Extensions .....	17
3. Bedienung des 3D-Sound-Mixers .....	18
3.1 Aufruf und Initialisierung .....	18
Startoptionen einstellen .....	18
Mainscreen und Initialisierung .....	19
CPU-Auslastung und Not-Aus .....	20
2.3 Eingangskanalzug .....	21
Quellenauswahl .....	22
Auxiliary-Wege .....	23
Filter-Sektion .....	23
Plugins .....	24
Direkt-Mode .....	24
Ambisonic-Mode .....	24
Volume und VU-Meter .....	25
2.4 Master-Sektion .....	25
Ambisonic-Decoder .....	26
Bus-Master und -Matrix .....	26
Settings und Cues .....	27
2.5 Extensions .....	28
4. Anwendungsbeispiele .....	29
4.1 Steuerung des 3D-Mixer mit Samplitude .....	29
4.2 Erweiterbarkeit mit eigenen Patches .....	30
Literatur .....	31

## 1. Einleitung

Klang und Musik wird in der europäischen Kultur spätestens seit der Renaissance ein räumlicher skulpturaler Aspekt zugemessen, aber erst die heute verfügbaren Technologien der neuen Medien ermöglichen die beliebige 3-dimensionale Verteilung von Sound, sei es im virtuellen Raum durch Kopfhörerwiedergabe oder im Realraum über Mehrkanal-Lautsprechersysteme. Das Projekt „3D-Sound-Mixer“ zielt auf die Entwicklung eines universell einsetzbaren Software-Tools, das Klang in seiner räumlichen Verteilung steuern kann und dabei auf eine Kodierung dieser Klangsteuerung setzt, die unabhängig von der konkreten Wiedergabesituation – vom Kopfhörer über die Surround-Anlage im Wohnzimmer bis zu vielkanaligen Kuppelbeschallungen in großen Theatern – ist. KomponistInnen, KlangkünstlerInnen und TonmeisterInnen soll damit die Möglichkeit an die Hand gegeben werden im eigenen Studio räumliches Klangdesign für beliebige Aufführungssituationen zu erstellen.

Dreidimensionale Klangwiedergabe wird bisher strikt getrennt für die Bereiche Kopfhörerwiedergabe und Lautsprecherwiedergabe gelöst. Bei der Kopfhörerwiedergabe wird das richtungsabhängige Übertragungsverhalten einer Schallquelle zum Kopf des virtuellen Hörers auf das Klangsignal aufgeprägt. Der Hörer hat den Eindruck den Sound aus der entsprechenden Richtung zu hören. Im idealen Fall wird die Kopfbewegung getrackt und die Kopfhörersignale so modifiziert, dass die Lokalisation der Schallquelle ortsfest erscheint.<sup>1</sup>

Bei mehrkanaliger Lautsprecherwiedergabe wird das Soundsignal richtungsabhängig zwischen mehreren Lautsprechern panoramisiert. Der Nachteil dieses Ansatzes ist, dass die Steuerparameter je nach konkreter Lautsprecherkonfiguration verändert werden müssen. Soll zB – wie im Tourneebetrieb üblich - eine Soundscape in verschiedenen Räumen, dh über verschiedene LS-Konfigurationen wiedergegeben werden, sind in jedem Saal aufwändige Adjustierungsarbeiten notwendig.

Ein antizipierende kreative Gestaltung von 3D-Sound in der Vorproduktion wird dadurch erschwert, dass der/die Künstlerin während der Gestaltungsphase in den seltensten Fällen auf die Endkonfiguration des LS-Setups zugreifen kann und

---

<sup>1</sup> Siehe Arbeiten von Sontacchi, Noisternig, Höldrich, Musil im Literaturverzeichnis und untern iem.at

deshalb entweder eine im besten Fall näherungsweise Realisation ihrer Ideen erreicht oder ganz auf das gestalterische Element „Raum“ verzichten muss.

Hier setzt das Projekt „3D-Soundmixer“ an, dessen auf Ambisonic basierender Ansatz der Schallfeldkodierung und –dekodierung die Steuerung der virtuellen Klangwege vollständig vom verwendeten Wiedergabe-Format trennt. Es ist daher möglich eine 3-dimensionale zeitlich veränderliche Klanglandschaft im einen Studio über Kopfhörer zu entwerfen und ohne Veränderung der Steuerparameter über verschiedene LS-Setups wiederzugeben. Nach dem Ambisonic-Prinzip<sup>2</sup> wird in der Einkodierung eine mehrkanalige lautsprecherunabhängige Repräsentation eines periphonen Schallfeldes generiert und diese anschließend für ein definiertes Lautsprecher-Setup mittels des Dekoders wiedergegeben. Allein der Dekoder berücksichtigt das konkrete Lautsprecher-Layout, sodass verschiedene Lautsprecherinstallationen nur eine Veränderung der Dekodergleichungen nach sich ziehen, die abstrakte Repräsentation des Schallfelds bleibt davon unberührt. Dadurch ist es - als Erweiterung oder Simulation des Lautsprechersystem – auch möglich die Wiedergabe durch eine binaurale Aufbereitung der Lautsprechersignale über Kopfhörer zu gestalten<sup>3</sup>. Als typisches Anwendungsszenario in der künstlerischen Produktion wäre der Fall anzusehen, bei dem die Künstlerin die Klangwege zuerst am eigenen Computer zuhause gestaltet und dabei über Kopfhörer hört, während die Anpassung an das Lautsprechersystem für die Konzertsituation in einer kurzen Einstellprobe erfolgen kann.

Das IEM bezog im Herbst 2000 neue Institutsräumlichkeiten. Zu diesen gehört auch ein mittelgroßer Aufführungsraum für Elektronische Musik, der IEM CUBE. Dessen Beschallungssystem wurde so gestaltet, dass es eine große Vielfalt von Aufführungs- und Wiedergabemöglichkeiten zur Verfügung stellt. Das Hauptsystem besteht aus einer Hemisphäre mit 24 Lautsprechern, für die eine erste Version des 3D-Ambisonic Mixers von Johannes Zmölnig in seiner Diplomarbeit realisiert wurde. Verbesserungen und Erweiterungen der Software wurden von Thomas Musil auf Grundlage der Forschungsarbeiten von Alois Sontacchi geschaffen.

Mit der in diesem Dokument beschriebenen Version des 3D-Mixers wird ein Software-Paket vorgestellt, das der verschiedenen Applikationen zu einem System

---

2 Siehe Arbeiten von Sontacchi et al im Literaturverzeichnis

3 Zur zeitvarianten Binauraltechnik mit Raumsimulation siehe Arbeiten von Sontacchi, Noisternig, Höldrich etc. unter iem.at

konvergieren lässt und das den Ambisonic-Mixer um ein traditionelles Bus-Konzept erweitert. Um möglichst große Flexibilität im Hinblick auf zukünftige Algorithmen und Hardware-Entwicklungen zu gewährleisten, war es notwendig eine modulare Software-Struktur zu verwenden. Durch diesen Aufbau ist es möglich, einzelne Komponenten bei Bedarf hinzuzufügen beziehungsweise obsoletere Module auszutauschen. Nach anfänglich getrennten Applikationen für die direkte Ansteuerung der Lautsprecher und das Ambisonic-System wurden diese Konzepte zu einem zusammengefasst und erweitert. Je nach Verfügbarkeit entsprechender Hardware können noch umfangreichere Lautsprecherkonfigurationen gesteuert werden.

Ein wichtiges Augenmerk wurde auf die Bedienbarkeit mit verschiedenen Eingabegeräten und Steuerschnittstellen gelegt. Die Bedienung erfolgt nicht ausschließlich mit Maus und Tastatur, sondern zusätzlich - wie das bei digitalen Mischpulten bereist Standard ist - mit Motor-Fadern, Drehknöpfen und Schalterelementen. Diese Steuermöglichkeit wurde über MIDI-Schnittstellen und mittels Netzwerk implementiert.

Parallel zum Ambisonic-basierten 3D-Sound Mixer wurde ein **Busmixer** implementiert, womit jeder Eingangskanal auf Busse geroutet werden kann, die dann über eine Gewichtungsmatrix auf alle Ausgänge verteilt werden. Funktionen wie Auxiliary-Wege oder Kanal-Plugins sind ebenso implementiert wie Monitoring mit Stummschaltung und Solo-Funktion.

Die Software wurde mittels „Puredata“ (PD) implementiert und ist so aufgebaut, dass sie einfach durch Plugins, Extensions oder eigene PD-Patches erweitert werden kann (zur näheren Erklärung siehe unten). Zusätzlich wurde eine Aufteilung in zwei Programmteile für das grafische Interface und die Signalverarbeitung vorgenommen, um auf möglichst geringe Latenzzeiten zu erreichen.

## 2. Architektur des 3D-Sound Mixer

Die folgende Beschreibung bezieht sich auf eine Maximalapplikation.

### 2.1 Hardware-Architektur

Das System besteht aus einer Audio-Workstation, den Benutzerschnittstellen und den Schnittstellen zur Fernsteuerung (Remote Control) von anderen Geräten und/oder aus Fremdprogrammen.

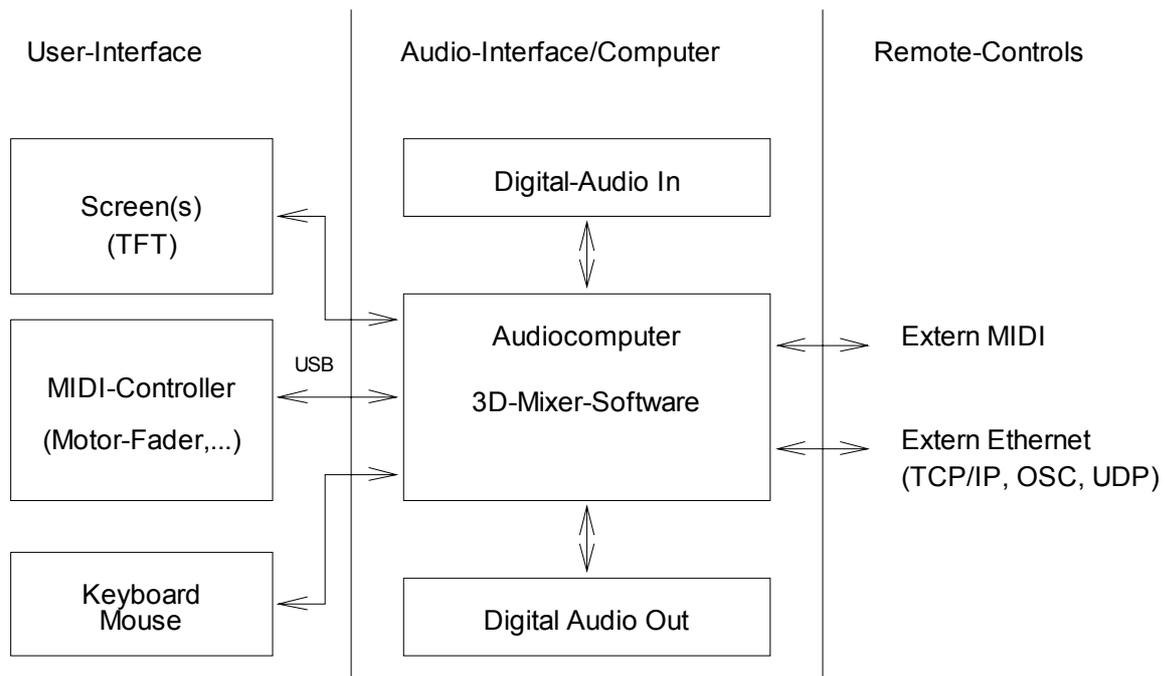


Abbildung 1: Hardware-System des 3D-Sound- Mixer

### Typische Hardware-Konfiguration und Benutzerschnittstelle

Es wird eine Audio-Workstation mit einem mehrkanaligen Audio-Interface vorgeschlagen (typisch 24 Kanäle, zB RME HDSP9652 ). Die Rechenleistung sollte entsprechend hoch sein, damit eine ausreichende



Abbildung 2: Typische Konfiguration der Benutzerschnittstellen

Reserve für kurze Latenzzeiten in pd vorhanden ist. Getestet wurde das System auf einem AMD Athlon64 X2 Dual Core Processor 4200+, bei dem eine durchgehende Systemlast unter 50% erreicht wurde. Damit sind für zusätzliche Anwendungen (Harddiskrekorder, etc) noch Leistungsreserven vorhanden. Die Software wurde unter einem Debian Linux Betriebssystem entwickelt und getestet, obwohl sie auch unter Mac OS-X und Windows verwendbar sein sollte. Es wird empfohlen aufgrund der umfangreichen grafischen Oberfläche das System mit zwei Monitoren mit erweitertem Desktop in einer Auflösung von 2650x1024 zu betreiben.

Die Hauptbedienelemente sind Maus und Tastatur. Zusätzlich werden als Steuereinheit MIDI-Fader-boxen unterstützt, die über USB angeschlossen sein sollten, um ausreichende Bandbreite bei der Übertragung der Steuerdaten zu erreichen. (siehe Abb 3)

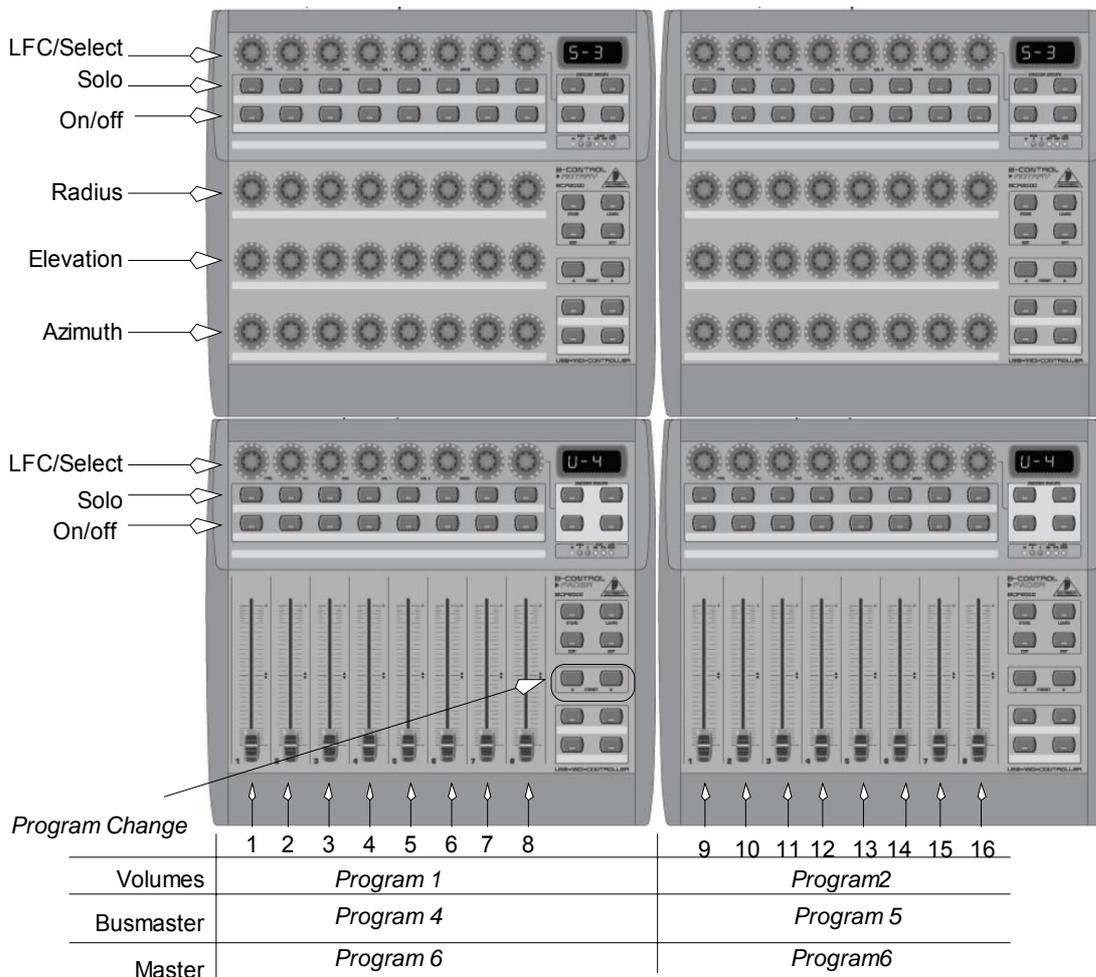


Abbildung 3: MIDI-Controller von Behringer zu einer Bedieneinheit zusammengefasst

## Remote Control

Alle Parameter des Systems kann sowohl über eine externe MIDI-Schnittstelle als auch über Ethernet gesteuert werden. Als Remote Control dienen typischerweise vier Controller von Behringer, welche zu einen 16-kanaligen Steuerpult zusammengefasst wurden. (siehe Abb.3)

## 2.2 Software-Architektur

Die Mixerapplikation wurde mit der grafischen Programmiersprache PureData programmiert. Diese verarbeitet alle Audiosignale mit 32-Bit Fließkommazahlen ( 24 Bit Mantisse und 8 Bit Exponent). Daraus ergibt sich ein entsprechender Dynamikbereich und entsprechende Genauigkeit bei der Parameterverarbeitung.

PureData lässt sich unter verschiedenen Betriebssystemen betreiben, so auch diese Applikation. Da einige Teile des Mixers mittels External - das sind in C-geschriebene Erweiterungen für PureData - realisiert wurden, ergibt sich die Notwendigkeit, diese auch für das entsprechende Ziel-Betriebssystem zu kompilieren. Ansonsten lässt sich die Software erweitern und verändern, ohne dass eine Kompilation notwendig ist.

### 2.2.1 Programmstruktur

Die 3D-Mixer-Applikation besteht aus zwei Programmen, welche über TCP/IP kommunizieren, wobei eine die grafische Oberfläche betreibt und die zweite die DSP-Engine darstellt. Diese werden mittels eines Start-Skript an einem Computer gestartet. Sie benutzen dieselbe Dateistruktur.

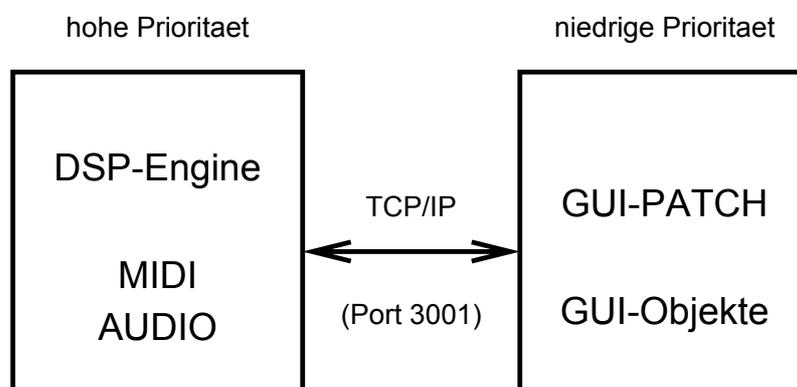


Abbildung 4: Trennung von GUI und DSP

Alle zeitkritischen Prozesse inklusive der Steuerdaten werden in der DSP-Engine verarbeitet. Über den GUI-Prozess ist die Steuerung des DSP-Patches möglich. Alle Parameter des DSP-Engine können parallel durch verschiedene Schnittstellen - wie zB MIDI - gesteuert werden, wobei die aktualisierten Steuerparameter auch an den GUI-Prozess zur Anzeige weitergeleitet werden. Die Struktur der Software ermöglicht eine einfache Erweiterung um weitere Steuermöglichkeiten, indem parallel Patches erstellt oder geladen werden können.

### **2.2.2 Aufbau des Mixers**

Im Folgendem wird nur auf die DSP-Engine eingegangen. Der Mixer ist aus einer frei definierbaren Zahl von Eingangskanälen aufgebaut. Jeder Kanal hat verschiedene Funktionselemente, die zugeschaltet werden können. Die Signale werden in die Master-Sektion geleitet und werden dort weiterverarbeitet.

Plugins sind Module, die optional in einen Kanalzug geschaltet werden können, Extensions sind Module, die zur Master-Sektion hinzugefügt werden können und somit die Funktion des Mixers erweitern.

Es gibt einen fixen Satz von Parametern, auf die auch in den Plugins oder Extensions zugegriffen werden kann. So wird im Kanalzug festgelegt, ob er den AmbisonicDecoder versorgt oder den Busmaster. Diese Routings schließen sich aus bedienungstechnischen und auch Performance-Gründen wechselseitig aus. Da

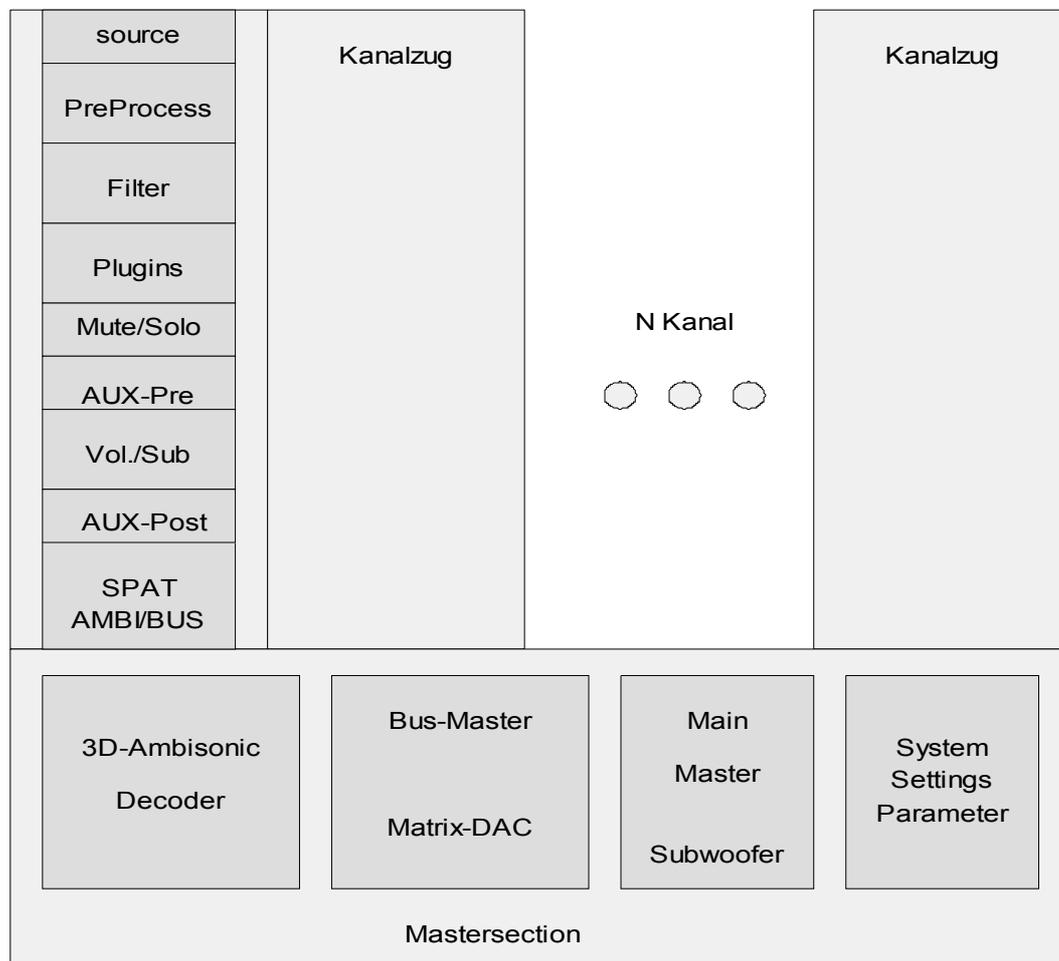


Abbildung 5: Mixerstruktur

aber gleiche Quellen mehrere Kanalzüge verwenden können, stehen alle Möglichkeiten zur Verteilung des Eingangssignals offen. Der Aufbau ist in Abbildung 5 ersichtlich.

### 2.2.3 Kanalzug

Die Reihenfolge der Elemente ist wie in Abbildung 5 dargestellt implementiert. Die AUX-Wege können vor oder nach der Lautstärken-Regelung geschaltet werden. Eine Deaktivierung eines Elementes bewirkt eine Überbrückung.

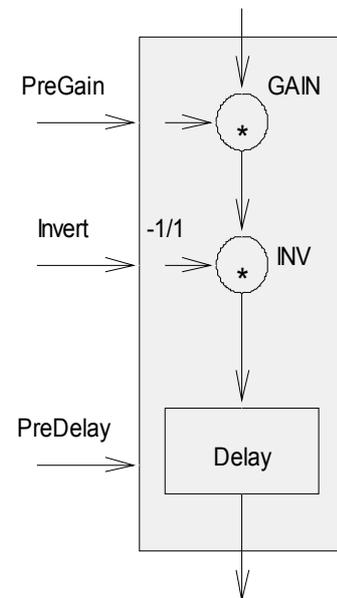
## Source Select

In der Auswahl des Quellensignals werden alle im System vorgesehenen Signale angeboten. Diese sind entweder Signale der ADCs, AUX-Summen oder auch Signale von Erweiterungen, wie zum Beispiel Soundfile-Player oder Testsignale. Diese Liste wird dynamisch generiert.

## Vorverstärkung und Delay

Mit der Möglichkeit einer Vorverstärkung (*PreGain*) können die Eingangsspiegel angepasst werden. Dies wird vor allem bei Steuerungen durch Fremdprogramme verwendet, wobei mit dem Gain noch nachgeregelt werden kann.

Die Eingangsverzögerung (*PreDelay*) ist einem Bereich von 0-1 Sekunde einstellbar und dient etwa zur Laufzeitentzerrung bei verschiedenen Lautsprecherentfernungen.



## Plugins

Pro Kanal können bis zu vier Plugins aus einer Liste frei gewählt werden. Bei der Auswahl von Plugins muss auf die Leistungskapazität des Rechner geachtet werden. Beispiele sind ein Compander- oder ein Distanz-Plugin, die auf andere Kanalsteuerdaten zugreifen können.

## Parametrische Filter

Die Filter bestehen aus Highcut und Lowcut-Filter, die Besselfilter 3.Ordnung darstellen. Zusätzlich gibt es Hochpass- und Tiefpassfilter 2.Ordnung, die sich mittels einer Verstärkung weich hinzuschalten lassen. Die zwei Bandpässe besitzen einen Güteparameter für die Bandbreite und dienen hauptsächlich zur Formantenbildung und Frequenzgangkorrektur. Diese Filter sind sehr

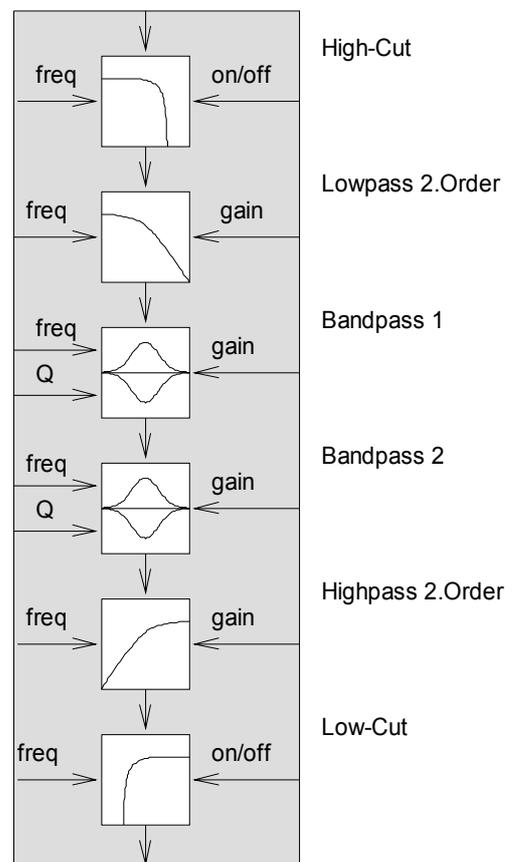


Abbildung 7: Filter im Kanalzug

recheneffizient implementiert und nicht für dynamische Parameteränderungen geeignet. Für dynamische Filter und Resonanzfilter sollten daher eher Plugins verwendet werden.

## Stummschaltung und Solo

Mute (Stummschaltung) dient zum passiv schalten von Kanälen. Der Kanal wird aus dem DSP-Prozess genommen und somit Rechenleistung gespart. Damit ist auch eine Skalierung der Rechenleistung möglich. Die Solofunktion ist eine Logik hinter den Mute-Schaltern, die alle Kanäle bis auf die als Solo markierten stumm schaltet. Damit können die einzelne Quellen getrennt gehört werden.

## Aux-Wege

Es wurden 8 AUX-Wege implementiert, wobei pro Kanal jeweils 4 Abzweigungen zu wählbaren AUX-Kanälen vor (*Pre*) und nach (*Post*) dem Lautstärkenregler wirken.

Die Aux-Summen werden von den Extensions verwendet oder können wieder als Quelle für einen Kanalzug dienen. Damit sind umfangreiche Verteilungen auf mehrere Kanäle möglich.

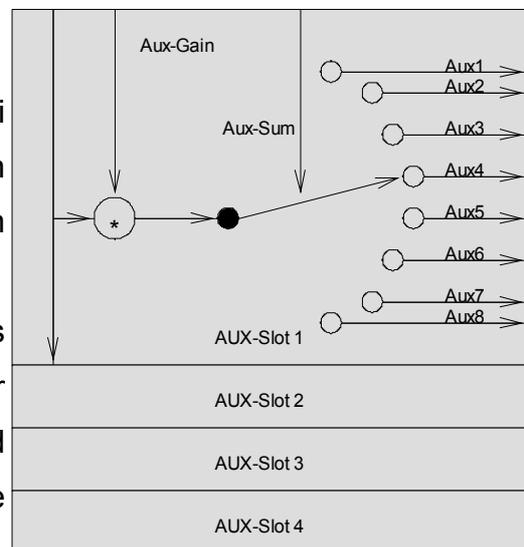


Abbildung 8: Auxwege im Kanal

## Fader und Subwoofer-Kanal

Die Lautstärke des Kanal wird über den Volumefader geregelt. Dahinter kann mit einen zweiten Regler der Kanal für Subwoofer (LFC, Low Frequency Channel) oder sonstige Mono-Summen generiert werden.

## Spatialisierung Ambisonic oder Bus-Direkt

Bei der Spatialisierung kann alternativ zwischen Bus- und Abbildung 9: Volume und LFC Ambisonic-Encodierung gewählt werden. Die Ambisonic-Encodierung wird durch drei Parameter gesteuert: Azimuth, Elevation und Radius. Der Azimuth ist in einem Winkel von -180 bis 180 Grad, die Evaluation von 0 bis 90 Grad einstellbar. Beim Radius handelt sich um eine zweistufige Regelung, die bei einem Radius grösser 1 den Pegel entsprechend einem Distanzgesetz zurückregelt. Bei einem Radius kleiner 1 wird die Quelle breiter, das bedeutet die Anteile für die Ambisonic-Kanäle höherer

Ordnung werden reduziert. Bei einem Radius von 0 wird die Quelle über alle Ausgangskanäle in gleicher Stärke gespielt. Der Parameter Radius wird auch im Plugin „Distanz“ verwendet, das noch einen Doppeleffekt für bewegte Quellen (durch ein variables Delay) und die entfernungsabhängige Luftdämpfung (durch einen Tiefpass) hinzufügt. Damit lässt sich gut eine Bewegung von Zentrum weg abbilden.

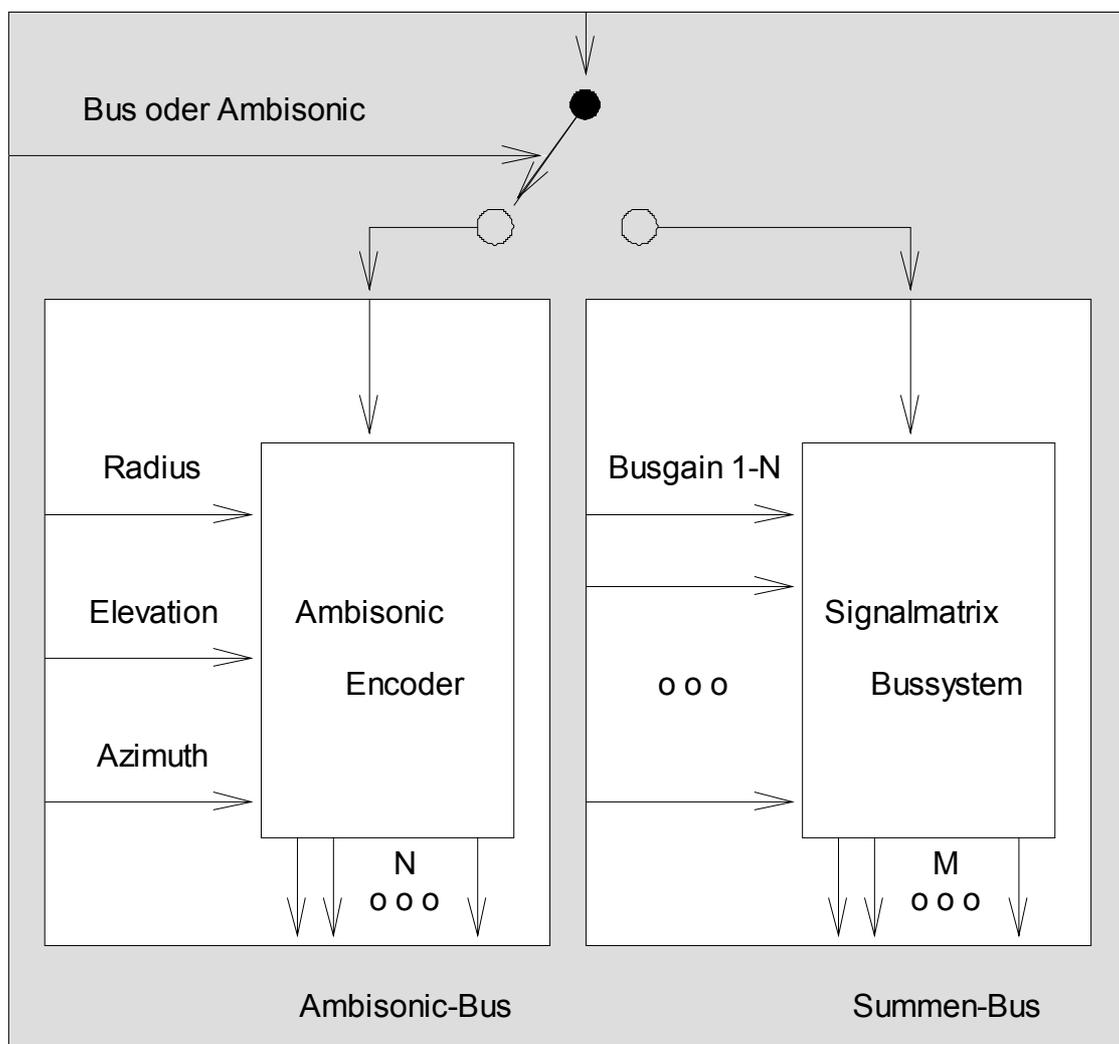


Abbildung 10: Spatialisierung Ambisonic oder Busmatrix

Bei der Bus-Matrix wird das Signal auf die implementierten Summen-Busse mit entsprechenden Verstärkungswerten von -99 bis 0dB aufgeteilt.

## 2.2.4 Master-Sektion

In der Master-Sektion sind alle Module zusammengefasst, die nur einmal im Mixer vorkommen. Hier werden alle Busse zusammen geführt, die Ausgänge verwaltet und zusätzliche Signale für weitere interne Verarbeitung generiert.

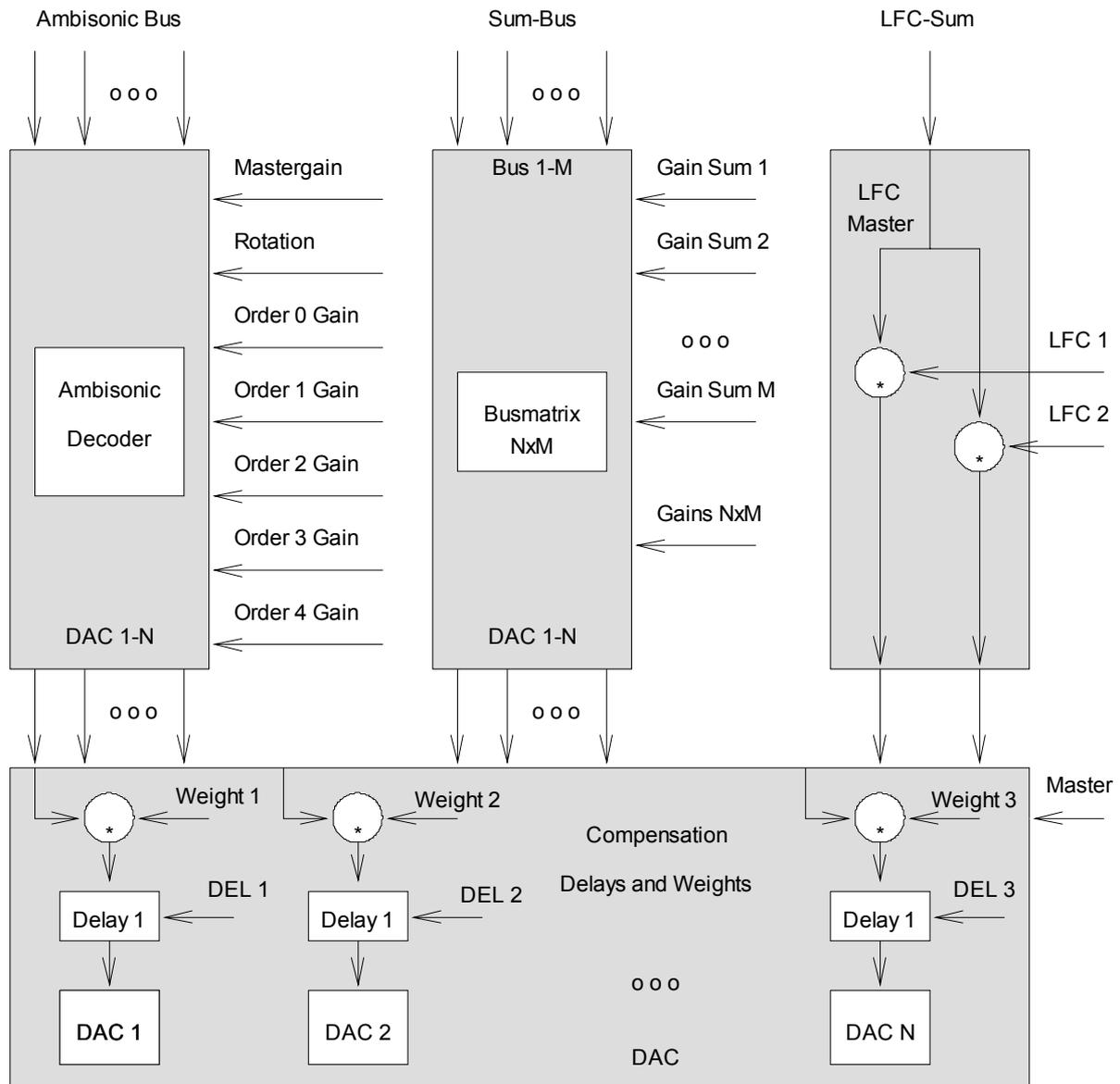


Abbildung 11: Mastersektion

Die Lautsprechersignale, die über den Ambisonic-Decoder oder die Bus-Summen gemischt werden, werden Delay- und verstärkungskompensiert; Delay- und Gain-Werte werden durch „Einmessen des Systems“ ermittelt.

### Ambisonic-Decoder

Der Ambisonic-Decoder dekodiert das Ambisonic-Signal in Lautsprecherkanäle. Dabei wurde der Decoder für ein bestimmtes Lautsprecher-Setup, also speziellen Winkeln der Lautsprecher vom Zentrum aus gesehen, konfiguriert.

Durch Vorschalten einer Rotationsmatrix kann der gesamte virtuelle Klangraum gedreht werden. Weiters wurde ein Verstärkungsregler für den gesamten Decoder eingebaut.

## **Summenmatrix**

In der Summenmatrix werden alle Bus-Summen in eine Kreuzschiene gespeist, deren Knoten regelbar sind. Die Kreuzschiene verteilt diese Summensignale auf die Ausgangskanäle und damit auf die DACs. Dadurch kann für jeden Bus eine gewichtete Lautsprecherverteilung gebildet werden.

## **LFC-Master**

Das LFC-Signal wird wie in Abbildung 11 gezeigt in zwei Summen aufgeteilt, um bis zu zwei Subwoofer-Kanäle<sup>4</sup> ansteuern zu können. Zusätzlich sind optional Tiefpass-Filter eingebaut.

## **Kompensationstufe**

Mit der Kompensationstufe können die unterschiedlichen Abstände der Lautsprecher zum Zentrum mittels Verzögerungsleitungen (Delays) und die unterschiedlichen Verstärkungen über den Signalweg justiert werden. Diese Werte werden durch „Einmessen des Systems“ mit einer getrennten Software ermittelt und in einer Datei gespeichert, die beim Start eingelesen wird<sup>5</sup>. Eine korrekte Laufzeit und Pegelkalibrierung ist vor allem für eine zufrieden stellende Funktion der Ambisonic-Spatialisierung notwendig.

### **2.2.5 Systemsteuerung**

In der Systemsteuerung kann das Setup des Mixers und dessen Parameter gespeichert und wieder geladen werden. „Setup-Daten“ sind Verschaltungen und Kalibrierungen innerhalb des Mixers. Parameter sind Verstärkungswerte, Spatialisierungsdaten und sonstige veränderliche Werte des 3D-Mixer. Mit den Parametern können Szenen gespeichert und wieder aufgerufen werden.

Weiters werden die Verbindungen zwischen den Prozessen und die Initialisierung vorgenommen.

### **2.2.6 Plugins**

Plugins sind Programme, die sich als Inserts in einen Kanalzug einhängen können. Diese können frei programmiert und hinzugefügt werden. Als erstes Plugins ist ein

---

4 Im IEM CUBE als Kanäle 25 und 26 ausgeführt.

5 Eine korrekte Laufzeit und Pegelkalibrierung ist vor allem für eine zufrieden stellende Funktion der Ambisonic-Spatialisierung notwendig.

„Distanz“-Plugin implementiert worden, dessen Verhalten vom Radius-Wert gesteuert wird und somit vor allem für die Ambisonic-Spatialisierung gedacht ist. Ein zweites Plugin ist ein als „Compander“ bezeichneter Dynamikprozessor.

Plugins besitzen meist einen Signaleingang und ein Ausgangssignal, das weiterverarbeitet wird. Es kann jedoch intern alle Signale und Parameter des Systems verarbeiten sowie zusätzliche Signale bzw. Parameter generieren und versenden.

### **2.2.7 Extensions**

Extensions sind Erweiterungen im Master-Bereich. Typischerweise sind es Hall, Harddiskrekorder und anderes. Extensions werden vor allem durch die AUX-Busse gespeist und leiten ihre Ausgangssignale an eine beliebige Spatialisierungs-Sektion (Ambisonic-Decoder bzw. Bus-System) oder aber direkt an DACs. Zusätzlich können sie Signale generieren, die an den Kanalzügen als selektierbare Quelle anliegen.

#### **Einfacher Ambisonic Hall**

Als erste Extensions ist ein Ambisonic-Hall implementiert, der von auswählbaren AUX-Wegen gespeist wird und Erstreflexionen und Diffus-Hall aus verschiedenen Richtungen mittels Ambisonic spatialisiert.

#### **8-Kanal Soundfileplayer**

Als zweite Extension wurde ein 8-Kanal Soundfileplayer implementiert, der Audiodateien im WAV-Format mit 1-8 Kanälen abspielen kann und diese als zusätzliche Quellen SFPLAYER:1 bis SFPLAYER:8 zur Verfügung stellt. Dateien mit weniger als 8 Kanälen werden entsprechend auf weniger Quellen verteilt.

### 3. Bedienung des 3D-Sound-Mixers

Das Software-Paket wird in einer fixen Dateistruktur installiert (siehe Anhang). Darin befinden sich im Ordner *pd* die Startskripte und alle PD-Patches. Die folgende Beschreibung bezieht sich auf das Linux-System. Als Verstärkungs- und Amplitudeneinheiten werden standardmäßig dB verwendet, wobei die interne Darstellung in MIDI-dB - 100.0 entspricht 0dB bzw. 1-Verstärkung und 0.0 entspricht  $-\infty$  dB - erfolgt. Als Einheiten für Winkel werden Grad verwendet. Die Bedienung erfolgt über die grafische Oberfläche, ein Teil der Funktionen ist parallel mit den MIDI-Controller steuerbar.

#### 3.1 Aufruf und Initialisierung

Im Ordner *pd* befindet sich das Startskript *start.sh* und *startALL.sh*. Mit *start.sh* wird nur die DSP-ENGINE gestartet, mit *startALL.sh* die DSP-Engine und das GUI-Programm. Dies kann über eine Desktop-Verknüpfung mit einem Startsymbol oder durch eine Kommandozeile aufgerufen werden.

#### Startoptionen einstellen

Die Startoptionen müssen eventuell noch an die Audiohardware angepasst werden. Dies geschieht im Aufrufskript *startALL.sh* mittels eines Texteditors. Dort können einige Performance-Einstellungen mit Optionen in der Variable *PDDSPFLAGS* vorgenommen werden, indem diese Zeile editiert wird:

```
PDDSPFLAGS=""
```

Beispiele sind:

```
PDDSPFLAGS="-32bit -rt -channels 26 -audiobuf 50 -alsamidi -mididev 1"
```

oder

```
PDDSPFLAGS="-alsa -channels 24 -r 44100"
```

Alle verwendbaren Parameter können durch den Aufruf von *pd* mit dem Parameter *-help* angezeigt werden. Es sollten jedoch nur Parameter, die die Performance beeinflussen, verändert werden (*pd* 0.39):

audio configuration flags:

- r <n>            -- specify sample rate
- audioindev ... -- audio in devices; e.g., "1,3" for first and third
- audiooutdev ... -- audio out devices (same)
- audiodev ...    -- specify input and output together
- inchannels ... -- audio input channels (by device, like "2" or "16,8")
- outchannels ... -- number of audio out channels (same)

- channels ... -- specify both input and output channels
- audiobuf <n> -- specify size of audio buffer in msec
- blocksize <n> -- specify audio I/O block size in sample frames
- sleepgrain <n> -- specify number of milliseconds to sleep when idle
- oss -- use OSS audio API
- 32bit ---- allow 32 bit OSS audio (for RME Hammerfall)
- alsa -- use ALSA audio API
- alsaadd <name> -- add an ALSA device name to list

MIDI configuration flags:

- midiindev ... -- midi in device list; e.g., "1,3" for first and third
- midioutdev ... -- midi out device list, same format
- mididev ... -- specify -midioutdev and -midiindev together
- nomidiin -- suppress MIDI input
- nomidiout -- suppress MIDI output
- nomidi -- suppress MIDI input and output
- alsamidi -- use ALSA midi API

### Mainscreen und Initialisierung

Nach dem Start erscheint der Hauptbildschirm wie in Abbildung 12 dargestellt.

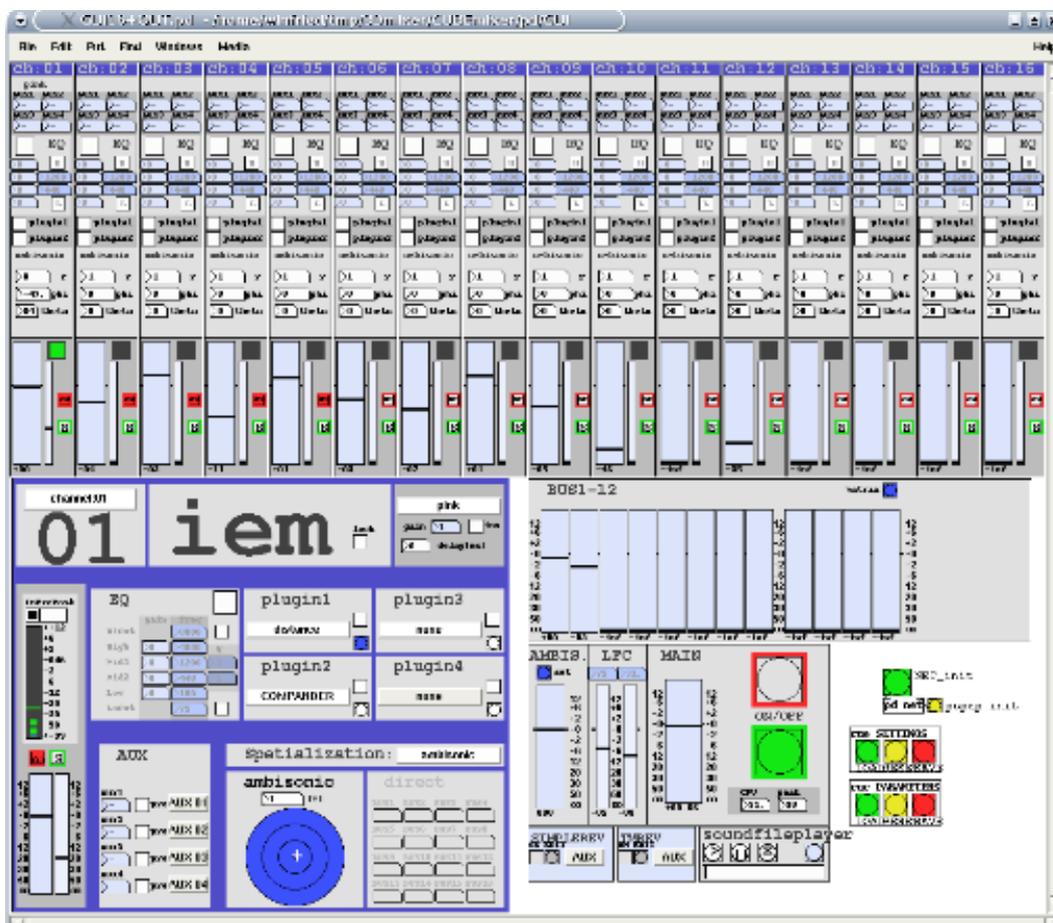


Abbildung 12: Mainscreen des 3D-Mixer

Die einzelnen Bereiche in diesem Fenster werden später noch im Detail dargestellt. Wenn der 3D-Sound-Mixer für eine Live-Aufführung verwendet wird, sollten mit dieser Darstellung in den meisten Fällen alle benötigten Elemente verfügbar sein. Wenn zusätzliche Dialoge benötigt werden, können diese auf einen zweiten Bildschirm in eigenen Fenstern positioniert werden.

Nach dem Start wird der Mixer initialisiert, dies dauert einige Sekunden. Danach wird der ON/OFF Schalter rot dargestellt und der Mixer ist im Standby-Betrieb (siehe Abbildung 13) . Er kann durch Betätigen des grünen Tasters eingeschaltet werden (siehe Abbildung 14a).

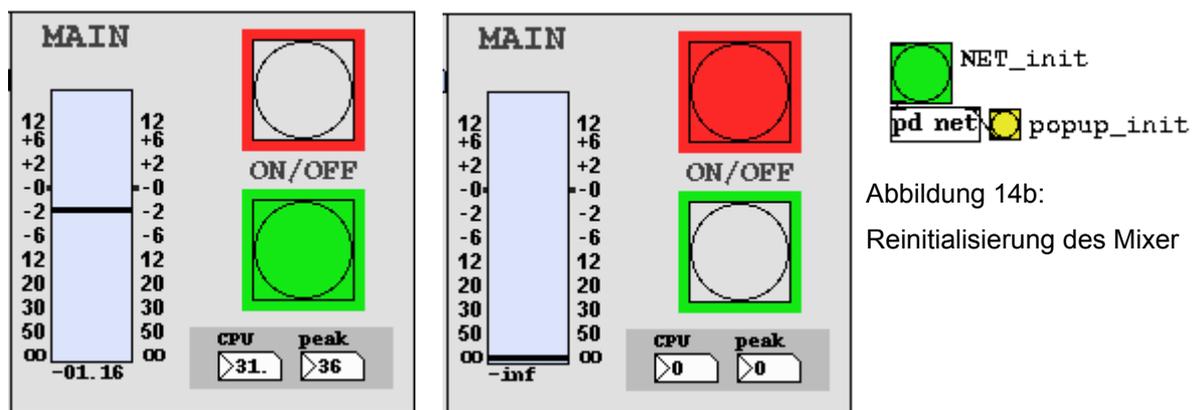


Abbildung 13: Mixer On

Abbildung 14a: Mixer Off

Wenn Änderungen im Patch vorgenommen wurden, kann der Mixer neu initialisiert und die Popup-Listen aktualisiert aufgebaut werden, indem - wie in Abbildung 14c dargestellt - zuerst *Net\_init* und anschließend *popup\_init* gedrückt wird.

### CPU-Auslastung und Not-Aus

Die aktuelle CPU-Auslastung durch den Mixer und die Spitzenauslastung (Peak) sind in Prozent getrennt angezeigt. Auf einem Linux-System sollte die CPU-Auslastung nicht 80 % und die Peak nicht 100 % überschreiten.

Wenn aufgrund eines Fehler (zB: Remote-Steuerung überlastet Computer) oder aufgrund zu geringer Rechenleistung die CPU-Last 100 % übersteigt, kann das System entsprechende Fehlsignale produzieren. Hier hilft das Betätigen des roten OFF-Schalter.

## 2.3 Eingangskanalzug

Die Kanalzug-Ansicht besteht aus mehreren Abschnitten, die getrennt aktiviert oder umgeschaltet werden können und eine Auswahl aller Parameter eines Kanals darstellen, wie dies in Abbildung 15 dargestellt ist.

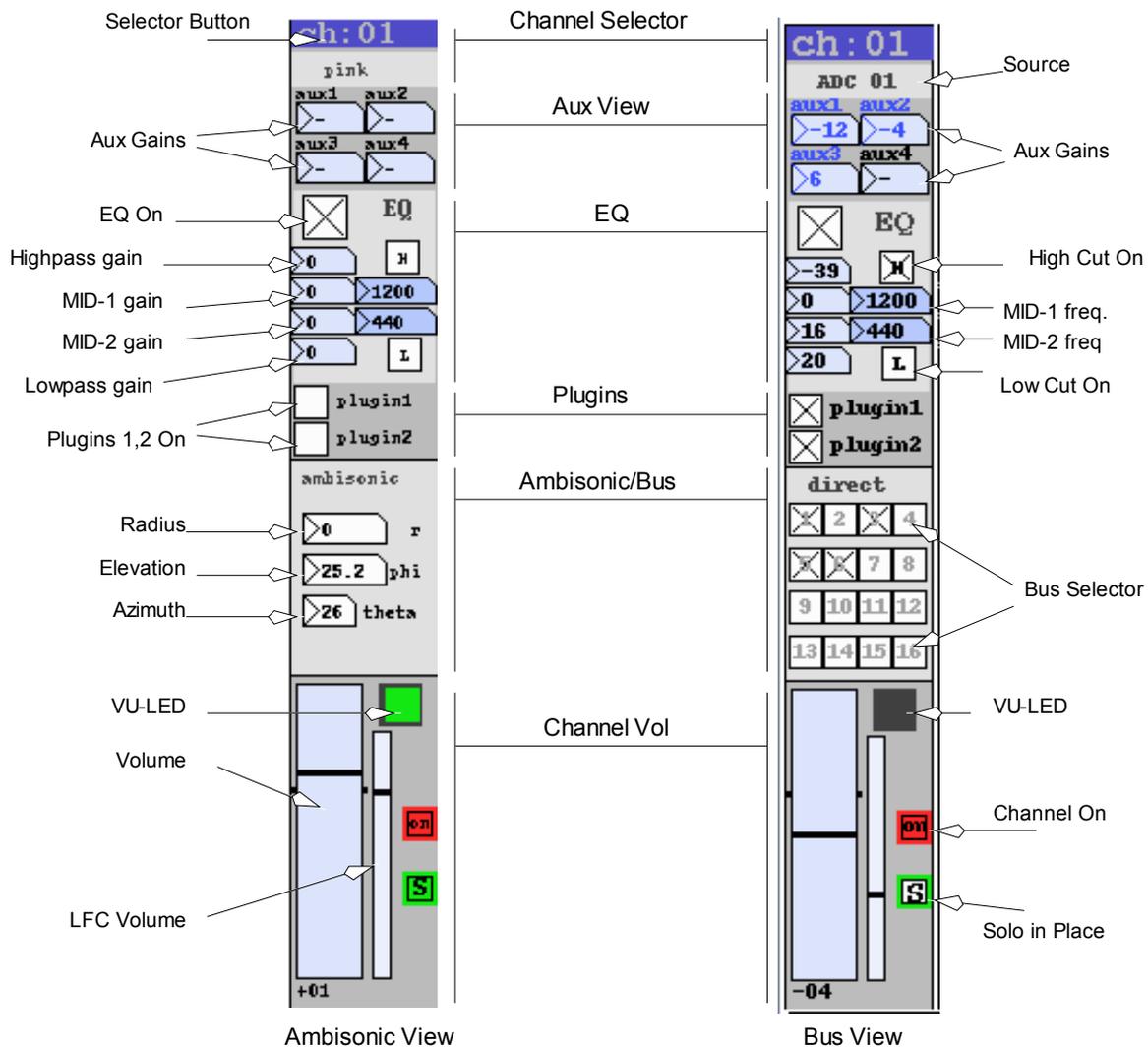


Abbildung 15: Kanalzug-Ansicht mit Ambisonic- und Busdarstellung

In der Kanalansicht wird der selektierte Kanal mit allen Parametern dargestellt (siehe Abbildung 16). Die Selektion erfolgt entweder durch einen Mausklick auf die Kanalnummer im Kanalzug oder durch ein Popup-Menu in der Kanalansicht und an den MIDI-Controller. In der Kanalansicht können die einzelnen Parameter nur editiert werden, wenn der **LOCK**-Knopf ausgeschaltet ist. Dies dient zur Sicherheit bei der Bedienung während Aufführungen und anderen kritischen Situationen. Es können nur noch Parameter verstellt werden, die die Performance nicht stören.

Bei der Spatialisation kann zwischen *direkt* und *ambisonic* Modus gewählt werden. Dies aktiviert die entsprechenden Felder in der Kanalansicht.

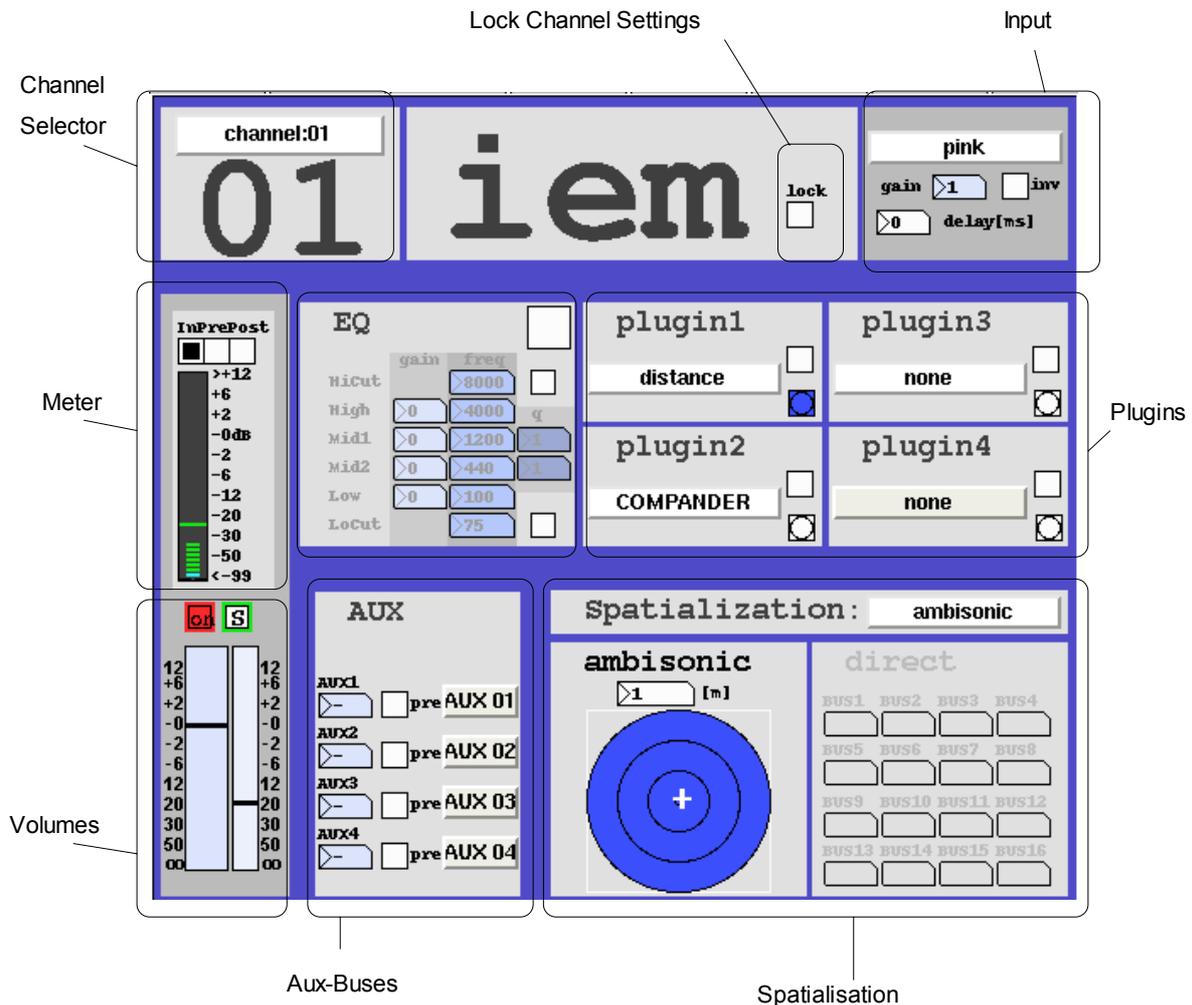


Abbildung 16: Kanalansicht mit den Bereichen

### Quellenauswahl

Die Quellenauswahl ist ein Popup-Menu, das dynamisch erzeugt wird. Hier kann zwischen ADC-Quellen, AUX-Bussen, Testsignalen oder Signalen der Extensions, wie zum Beispiel dem Soundfileplayer ausgewählt werden.

Weiters kann der Eingangssignalpegel durch *Pregain* eingestellt werden, was vor allem für dynamische Effekte wichtig ist. Das *Predelay* gibt die Möglichkeit einer

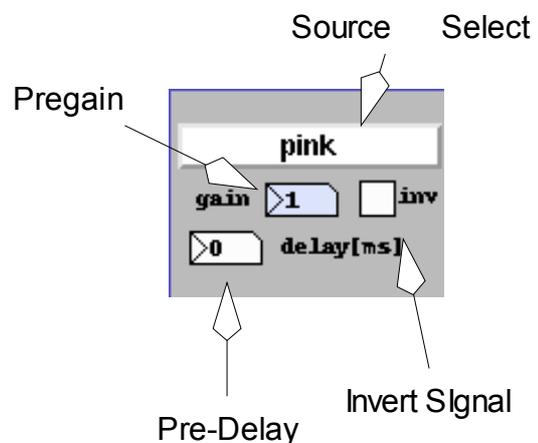


Abbildung 17: Kanaleingangsstufe

individuellen Zeitverzögerung des Signals. Die Phase wird mit dem *Invert Signal* Schalter definiert.

### Auxiliary-Wege

Es stehen vier Auxiliary-Abzweigungen zur Verfügung, für die mit dem Aux-Selector der entsprechende Send-Kanal gewählt werden kann. Der *Pre* Schalter definiert die Signalabzweigung vor oder nach dem Fader, der Pegel wird mit dem *AuxGain* eingestellt.

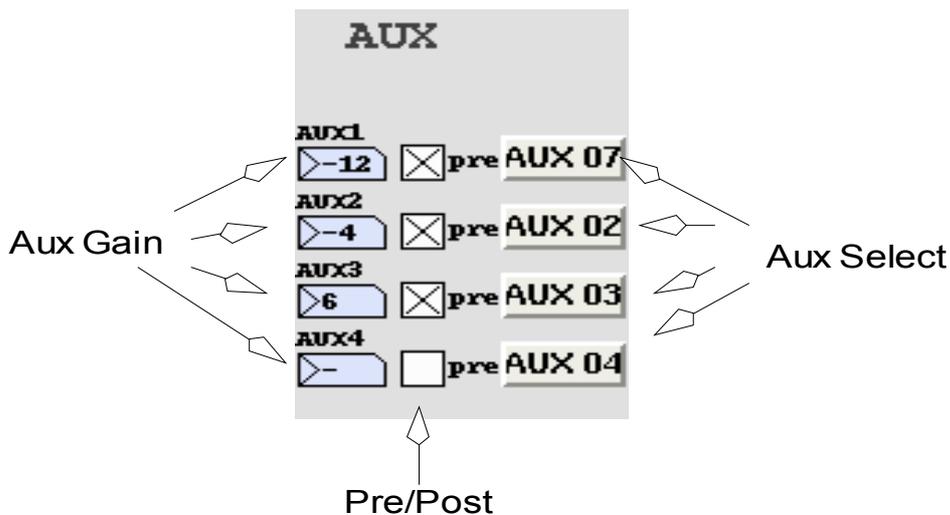


Abbildung 18: Auxiliary-Sektion

### Filter-Sektion

Die Filter-Sektion bietet die Möglichkeit sechs Filter-einheiten in den Kanalzug zu schalten. Wird eine Verstärkung (*Gain*) auf „-“ gestellt, so ist der Filter deaktiviert. Die Frequenzen der Filter sind frei wählbar. Bei den den Bandpassfiltern (**Mid1** und **Mid2**), kann die Bandbreite durch die Güte *q* eingestellt werden. *Hi-Cut* und *Low-Cut* können ein- und ausgeschaltet werden.

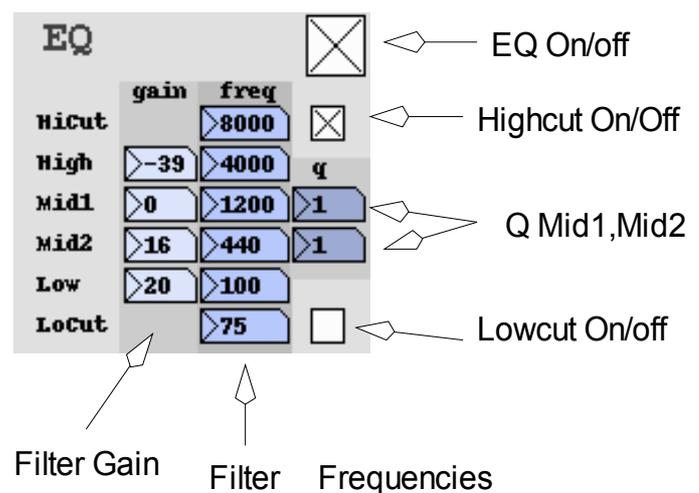


Abbildung 19: Kanalfiltereinstellungen

## Plugins

Mit dem Plugin-Selektor wird ein Popup-Menü geöffnet, bei dem aus den angebotenen Plugins gewählt werden kann. Diese können ein- und ausgeschaltet werden. Bei Betätigung des Edit-Knopfes öffnet sich der Edit-Dialog für das gewählte Plugin. Werden gleiche Plugins in verschiedenen Slots oder Kanälen verwendet, wird immer nur ein Dialogfenster mit den entsprechenden Parametern dargestellt.

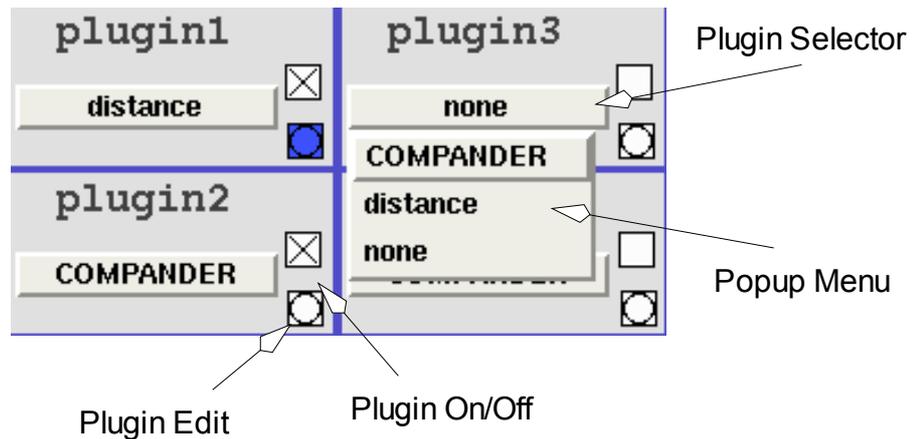


Abbildung 20 Plugin-Selektor

## Direkt-Mode

Das Routing eines Kanal im Direkt-Mode kann mittels einer Busrouting-Matrix eingestellt werden. Dazu werden die jeweiligen Wege mit einen Verstärkungswert belegt. „-“ bedeutet dabei kein Routing. Dies wird im Kanalzugdialog angezeigt. Wird im Kanalzugfenster ein Routing deaktiviert, so merkt sich die Matrix den Wert für die nächste Aktivierung.

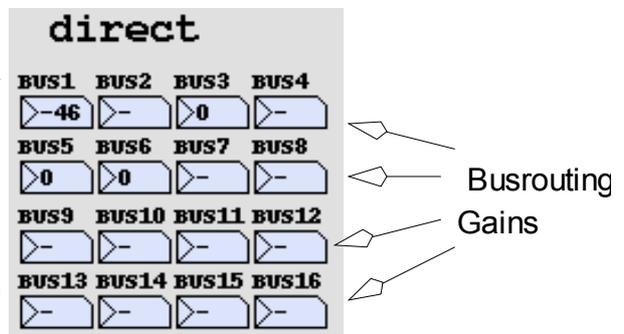


Abbildung 21: Busdialog im Kanalfenster

## Ambisonic-Mode

Im Ambisonic-Mode kann die *Azimuth*- und *Elevation*-Position des zu spatialisierenden Signals mittels eines speziellen 2D-Fader eingestellt werden. Damit kann eine Quelle dynamisch spatialisiert werden. Der *Radius* wird in einer Zahlenbox eingestellt. In Kombination

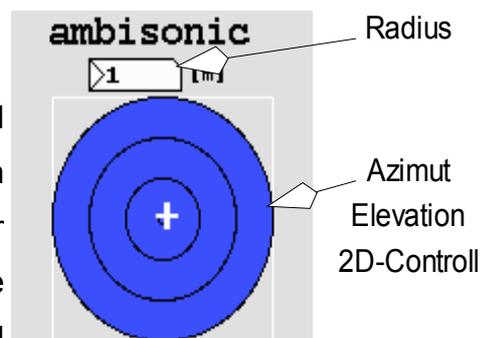


Abbildung 22: Ambisonic-Steuerung

mit den Distanz-Plugin kann der Effekt des Distanzparameters durch Luftdämpfung (mit einem Tiefpass) und Dopplereffekt (mit einem variablen Delay) verstärkt werden.

## Volume und VU-Meter

Im Kanaldialog kann die Lautstärke für die Spatialisierung und den LFC-Kanal über zwei Volume-Fader eingestellt werden. Ob ein Signal anliegt und welcher Pegel durch die Verstärkung erreicht wird, kann im VU-Meter durch entsprechende Auswahl des Abzweigpunktes im Kanalzug (beim Input, vor dem Fader *Pre* und nach dem Fader *Post*) angezeigt werden.

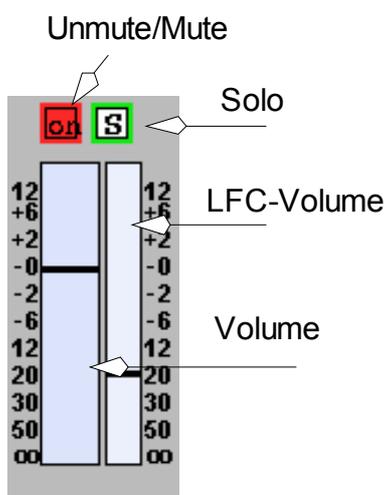


Abbildung 23a: Fader für Hautsignal und LFC

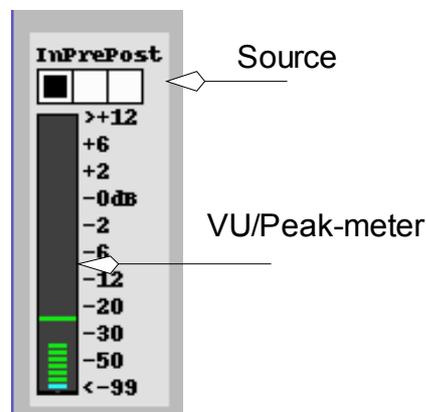


Abbildung 23b: VU-Meter

## 2.4 Master-Sektion

Zu den Parameter der Master-Sektion gehören die Volume-Regler für den Ambisonic-Encoder, die LFC-Kanäle und ein Main-Volume, das direkt auf die Ausgänge wirkt. Bei den LFC-Kanälen kann optional noch ein Tiefpass eingestellt werden. Mit dem *set*-Knopf im Ambisonic-Encoder kann der Decoder-Dialog in einem eigenen Fenster geöffnet werden (siehe unten).

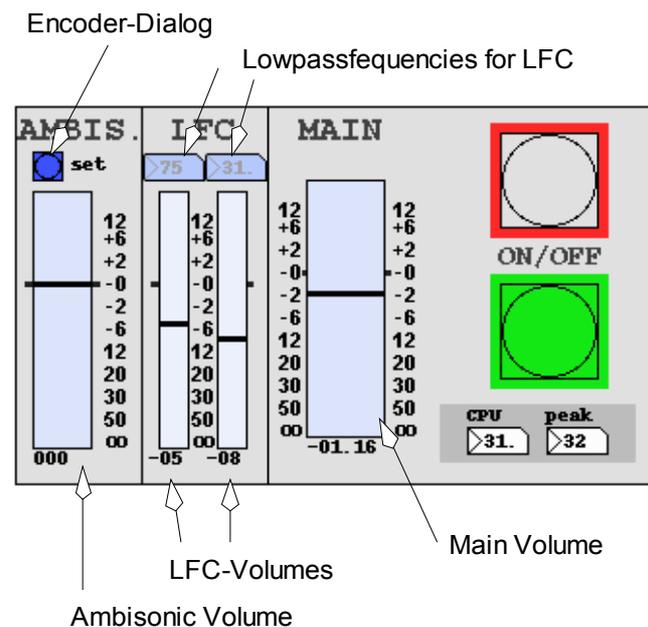


Abbildung 24: Master-Sektion

Bis auf das Main-Volume können

alle Parameter als Szene gespeichert werden.

### Ambisonic-Decoder

Der Ambisonic-Decoder hat einen Gain-Regler, der dem Ambisonic Volume-Regler entspricht. Darüber hinaus kann die Rotation aller Quellen eingestellt werden, um zB. die Gesamtorientierung des Schallfelds im Aufführungsraum zu drehen.

Die Gewichtungen der einzelnen Ambisonic-Ordnungen können verändert werden, um verschiedene Decoder-Strategien realisieren zu können (Basic, in-phase etc, siehe entsprechende Fachliteratur).

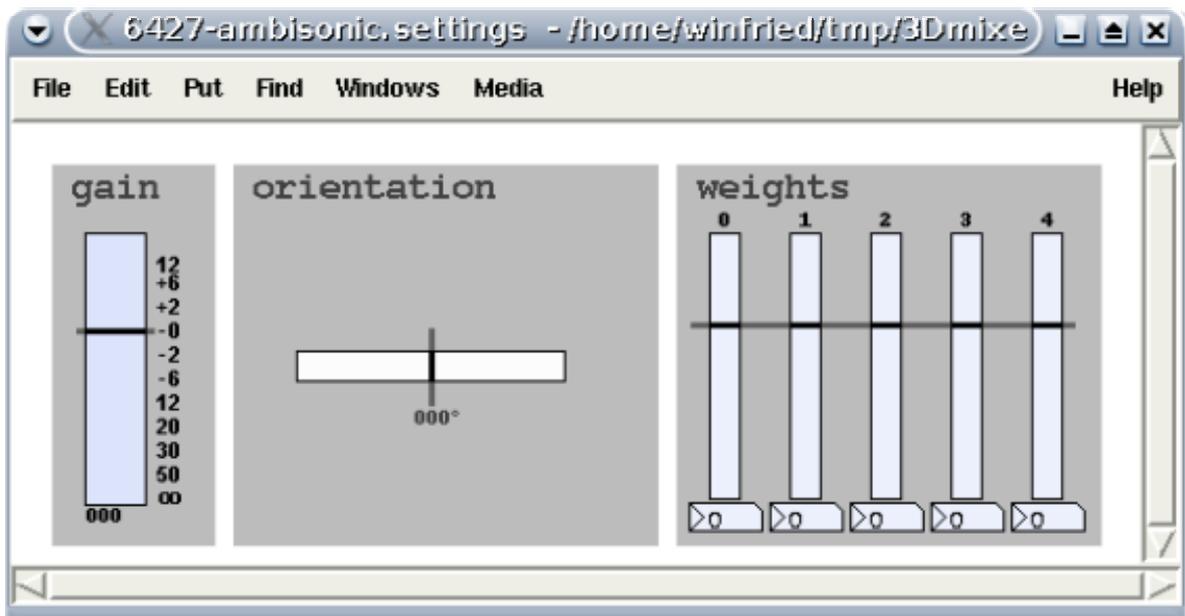


Abbildung 25: Ambisonic-Decoder Dialog

### Bus-Master und -Matrix

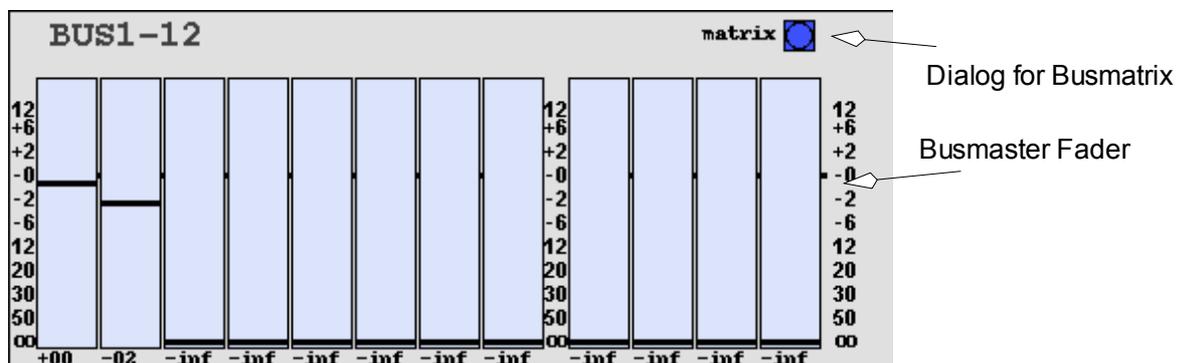


Abbildung 26: Bus-Master Dialog

Die Summenregler für die Busse sind im Bus-Master Dialog zu finden. Durch den *matrix*-Knopf kann der Dialog zur Bus-Matrix geöffnet werden.

In der Bus-Matrix wird das Routing mittels Verstärkungswerten der Busse auf die ADCs eingestellt. Damit kann ein Bus auf mehrere Ausgänge gewichtet verschaltet werden.

	BUS 0	BUS 1	BUS 2	BUS 3	BUS 4	BUS 5	BUS 6	BUS 7	BUS 8	BUS 9	BUS 10	BUS 11	
DAC01	-	-	-	-	-	-	-	-	-	-	-	-	DAC01
DAC02	-	-	-	-	-	-	-	-	-	-	-	-	DAC02
DAC03	-	-	-	-	-	-	-	-	-	-	-	-	DAC03
DAC04	-	-	-	-	-	-	-	-	-	-	-	-	DAC04
DAC05	-	-	-	-	-	-	-	-	-	-	-	-	DAC05
DAC06	-	-	-	-	-	-	-	-	-	-	-	-	DAC06
DAC07	-	-	-	-	-	-	-	-	-	-	-	-	DAC07
DAC08	-	-	-	-	-	-	-	-	-	-	-	-	DAC08
DAC09	-	-	-	-	-	-	-	-	-	-	-	-	DAC09
DAC10	-	-	-	-	-	-	-	-	-	-	-	-	DAC10
DAC11	-	-	-	-	-	-	-	-	-	-	-	-	DAC11
DAC12	-	-	-	-	-	-	-	-	-	-	-	-	DAC12
DAC13	-	-	-	-	-	-	-	-	-	-	-	-	DAC13
DAC14	-	-	-	-	-	-	-	-	-	-	-	-	DAC14
DAC15	-	-	-	-	-	-	-	-	-	-	-	-	DAC15
DAC16	-	-	-	-	-	-	-	-	-	-	-	-	DAC16
DAC17	-	-	-	-	-	-	-	-	-	-	-	-	DAC17
DAC18	-	-	-	-	-	-	-	-	-	-	-	-	DAC18
DAC19	-	-	-	-	-	-	-	-	-	-	-	-	DAC19
DAC20	-	-	-	-	-	-	-	-	-	-	-	-	DAC20
DAC21	-	-	-	-	-	-	-	-	-	-	-	-	DAC21
DAC22	-	-	-	-	-	-	-	-	-	-	-	-	DAC22
DAC23	-	-	-	-	-	-	-	-	-	-	-	-	DAC23
DAC24	-	-	-	-	-	-	-	-	-	-	-	-	DAC24
DAC25	-	-	-	-	-	-	-	-	-	-	-	-	DAC25
DAC26	-	-	-	-	-	-	-	-	-	-	-	-	DAC26

Abbildung 27: Busmatrix 12x26 Gains

### Settings und Cues

Es gibt zwei Arten von Daten, Settings und Parameter. Diese unterscheiden sich dadurch, dass die einen globale Einstellungen sind und die anderen für Szenen relevante Einstellungen. Damit können mit *cue PARAMETERS* Szenen gespeichert und abgerufen werden.

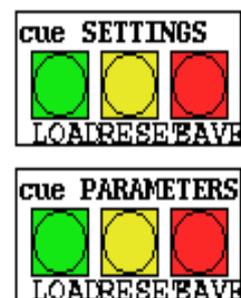


Abbildung 28 Settings

## 2.5 Extensions

Extensions sind Erweiterungen der Master-Sektion. Als Standard gibt es einen einfachen Ambisonic-Hall und einen Soundfileplayer.



Abbildung 29: Extension im Master

Im Extension Dialog befindet sich meist ein Knopf, der ein Editorfenster für die jeweilige Extension beinhaltet. Als Beispiel ist der Ambisonic-Hall in Abbildung 30 dargestellt.

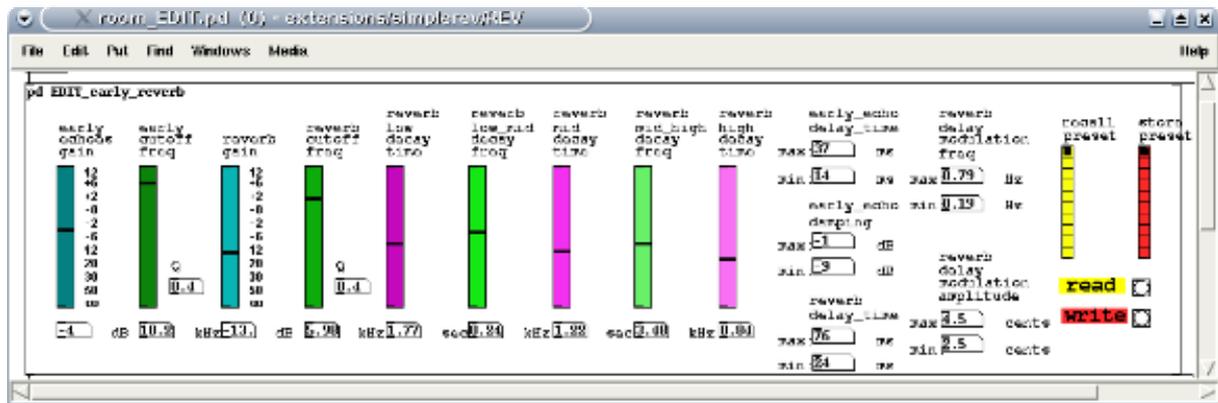


Abbildung 30: Hallparameterdialog des Hall Plugin

## 4. Anwendungsbeispiele

### 4.1 Steuerung des 3D-Mixer mit Samplitude

Zur Steuerung wird die MIDI-Remote Schnittstelle verwendet. Das Setup besteht aus der Verbindung des Samplitude Rechners über MIDI und digitales Audio mit dem 3D-Mixer.

Aus der im Anhang aufgelisteten MIDI-Controller Nummerierung können nun die entsprechenden Werte ausgelesen werden und im Samplitude entsprechende Controller an den Spuren angelegt werden. Günstig ist es, wenn die abgespielten Spuren gleichzeitig die Controller beinhalten.

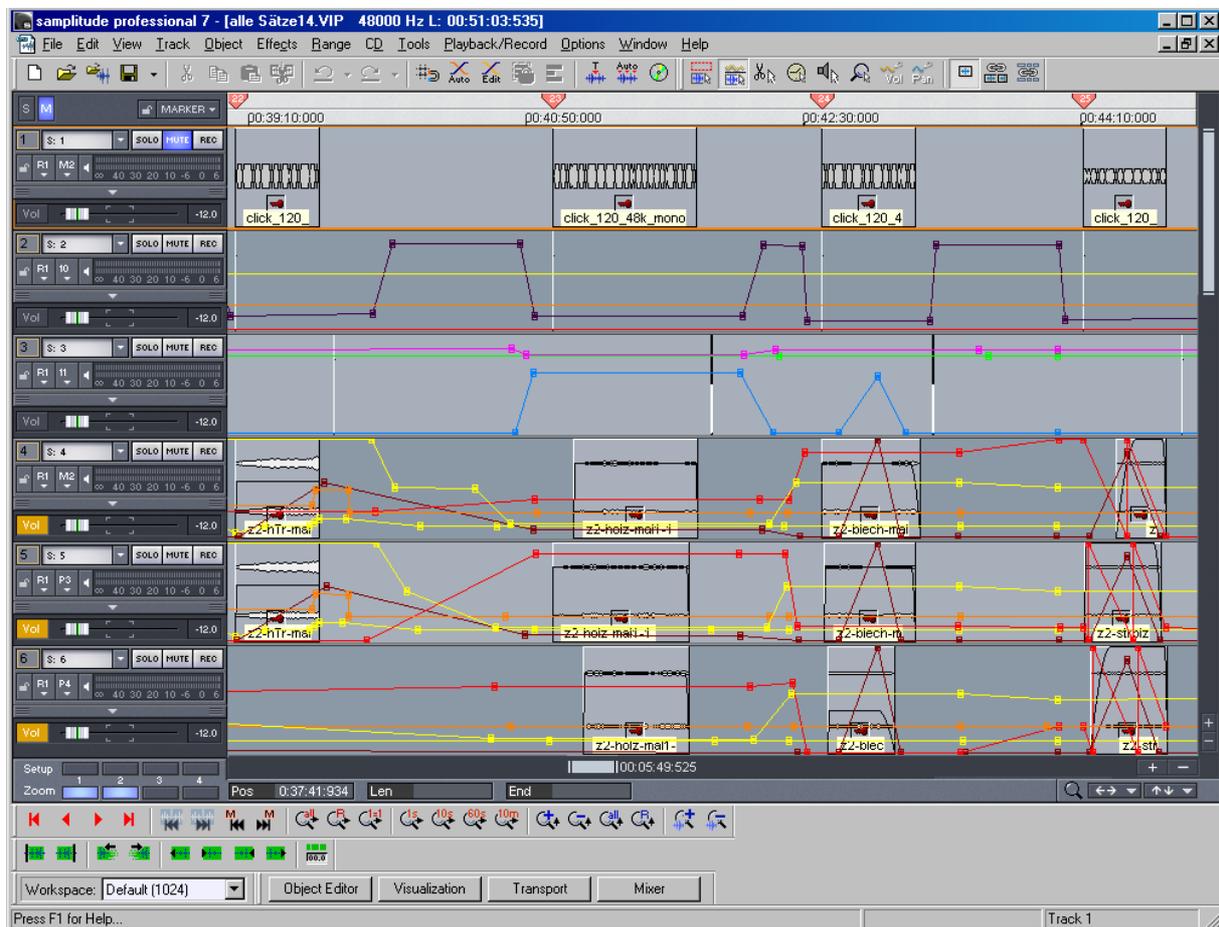


Abbildung 31: Steuerung mit Samplitude, Controller und Audio

Dabei ist zu beachten, dass MIDI eine begrenzte Datenübertragungsrate besitzt und zwar 31 kBit/sec. Daraus ergibt sich bei 3 Bytes pro Controller ein Maximum von ca. 1000 Controllerwerte/sec. Werden nun für jede Spur Volume, Azimuth und Elevation als Steuerparameter verwendet sind das pro Kanal 3 Controllerwerte. In der

Spatialisation können Controller-Raten von 50ms/Wert gerade noch differenziert werden. Daraus ergibt sich, dass theoretisch maximal 16 Kanäle parallel gesteuert werden können. Es sollte jedoch eine Sicherheit von 25% eingerechnet werden, wodurch sich damit eine maximal zu steuerbare Anzahl von 12 Kanälen ergibt.

Wird keine so feine Zeitauflösung benötigt kann diese entsprechend erhöht werden. Dies wird am besten in Samplitude eingestellt.

## **4.2 Erweiterbarkeit mit eigenen Patches**

Es kann der DSP-Mixer mit eigenen Patches erweitert werden, entweder in der DSP-Engine oder im GUI oder am besten in beiden. Die Verbindung zwischen GUI und DSP-Engine erfolgt mittels einer net-send-net-receive Paars in beide Richtungen. Wichtig dabei ist es sich an Namensraum der Applikation zu halten. Dieser ist im Anhang aufgelistet.

Ein Beispiel ist die Steuerung von zwei Kanälen über einen Raumwinkel, indem diese als Stereopaar definiert werden. Dabei müssen eigene Namen für die Steuerparameter definiert werden und dann an die entsprechenden Kanäle die Werte für Azimuth, Elevation und eventuell Radius gesendet werden. (siehe Example-Ordner).

## Literatur

Alois Sontacchi, Robert Höldrich: Probleme bei unsymmetrischen Lautsprecherlayouts und der Versuch diese zu kompensieren. IEM Report 9/99

Alois Sontacchi: Neue Ansätze der Schallfeldreproduktion, Dissertation der TU Graz 2001

Johannes Zmölnig, Winfried Ritsch, Alois Sontacchi: "Der IEM CUBE - ein periphones (Re)Produktionssystem", 22.TMT, Produktforum, Jahrestagung des Vereins Deutscher Tonmeister, Hannover, November 2002

Markus Noisternig, Alois Sontacchi, Thomas Musil, Robert Höldrich: "A 3D Ambisonic based Binaural Sound Reproduction System" AES 24th International Conference, 26-28 June 2003, Banff, Canada

Robert Höldrich, Winfried Ritsch, Christopher Frauenberger: Internet Archive for Electronic Music IAEM-iARS internet Audio Rendering System, AES 24th International Conference, 26-28 June 2003, Banff, Canada.

Thomas Musil, Johannes Zmölnig, Markus Noisternig, Alois Sontacchi, Robert Höldrich: AMBISONIC 3D-Beschallungssystem 5.Ordnung für PD, IEM Report 15/2003

Thomas Musil: IEMLIB für PD, IEM Report 12/2003

Johannes Zmölnig, Winfried Ritsch, Alois Sontacchi: "The IEM CUBE", ICAD - International Conference on Auditory Display, July 7-9, 2003, Boston University, Boston, MA

## Anhang

### A. Programmsetup

Die aktuelle Beschreibung hierfür findet sich in einem Dokumentationsordner in der Applikation.

#### Voraussetzungen

- PureData ab Version 0.39
- PD-Libraries sind im Projektordner inkludiert. Es sollten keine systemweit installierten PD-Externals verwendet werden.
- Audioworkstation mit mindestens 26 Kanal Audio I/O
- Betriebssystem Linux oder auch Windows, Max OS-X mit Anpassungen.

Unter Linux wird das Realtime-Scheduling ab der Kernel-Version 2.6.11 unterstützt, sodass eine entsprechend kleine Latenzzeit erreicht werden kann.

#### Versionen, Releases und Download

Das Softwarepaket mit Dokumentation wird auf dem Versionssystem CVS am IEM verwaltet.

Es werden zyklisch aktuelle Version dieser Software herausgegeben und als Pakete angeboten.

Die jeweils aktuellste, jedoch auch experimentelle Version, kann über den CVS-Server bei entsprechenden Rechten bezogen werden. Die Adresse ist *cvs.iem.at* und das Modul heißt *CUBEMixer*, zum Beispiel mit dem Befehl:

```
cvs -d:ext:<user>@cvs.iem.at co CUBEMixer
```

wobei <user> für einen registrierten Benutzer am IEM steht. Die aktuelle Version ist immer am IEM-Server angekündigt.

#### Verzeichnisstruktur

(aktuelle Version siehe README im Basisverzeichnis)

*3D-mixer: for the IEM CUBE and other environments*

---

---

*./pd pure-data patches (and externals)*  
*./doc documentation*  
*./src source-code (for externals)*  
*./data data*

directory-layout:

core modules:

<i>./pd/CUE</i>	- cue-triggering machine
<i>./pd/DSP</i>	- the audio-engine
<i>./pd/GUI</i>	- the Graphical User Interface
<i>./pd/GUI/&lt;name&gt;/</i>	- helper abstractions for the <i>./GUI/&lt;name&gt;.pd abs</i>
<i>./pd/MIDI</i>	- interface to MIDI-hw/sw
<i>./pd/MIDI/faderbox</i>	- interface to MIDI-hw/faderboxes [THINK about the name]
<i>./pd/MIDI/extern</i>	- interface to external MIDI-sequencer
<i>./pd/SEQUENCER</i>	- cue-sequencer

add-on modules: ???

*./pd/plugins/<pluginname>/* - DSP/GUI/MIDI/... part of plugins

helpers:

<i>./pd/abs</i>	- general abstractions
<i>./pd/libs</i>	- libraries (binaries)
<i>./pd/tests</i>	- sandbox for experimenting

## B. Namensraum des 3D-Mixer

Für die Steuerung durch eigens PD-Patches und zur besseren Erweiterbarkeit des Mixer wurde ein Namensraum definiert. Damit gibt es eine eindeutige Bezeichnung aller Steuerparameter und Steuerkanäle in der Software. Die aktuelle Liste liegt im Verzeichnis doc unter den Namen *send\_names.txt* .

### Parameternamen

1. Spalte = send/receive-name

```
===INPUT=====
=====
```

soundfile-player:

-----

jeder soundfile-player (egal wieviele kanäle er hat) hat einen uniq-name <ID>

ausgabe des sfplayers sollte auf <ID>:<channel>\_PLAY erfolgen

der <ID> wird immer prefixed

*\$1\_SFPLAY\_FILE* <soundfile>: lädt <soundfile>

*\$1\_SFPLAY\_CTRL* play|pause|stop: start/pause/stop

*\$1\_SFPLAY\_CTRL* jump <time> : springt zur zeit <time> in [ms]

```
===CHANNEL=====
=====
```

Routing Parameter werden nur am Anfang geladen und eingestellt, und duerfen im Sound einen Knackser verursachen.

(fuer IN-MIX-CH)

-----  
\$1\_IN\_SRC : selected INPUT or AUX-RET (1\_ADC ... 52\_ADC, 1\_AUX ... 8\_AUX)  
\$1\_IN\_DEL : Delay (0 ... 1000 ms)  
\$1\_IN\_INV : Signal-Inverter (+1 or -1)  
\$1\_IN\_AUX\_1\_DST : selected Destination of 1. AUX-SND (1\_AUX ... 8\_AUX) (--> audio-engine:  
[throw~ 1\_AUX\_SND])

Operating Parameter werden im Betrieb staendig nachgeladen und verursachen keine Knackser.

(fuer IN-MIX-CH)

-----  
\$1\_IN\_ON: on/off (!mute)  
IN\_SOLO: (1 - N) ch-number of exclusive ON; 0: all channels in prev. ON-state  
IN\_SELECT: (1 - N) ch-number of exclusive SELECT; 0: all channels in prev. ON-state

\$1\_IN\_ADJ\_GAIN : adjust the INPUT-Level (0 ... 150 MIDI-dB)

\$1\_IN\_EQ\_ON :  
\$1\_IN\_LOWCUT\_FREQ : in Hz  
\$1\_IN\_LOWCUT\_ON : positive logik (1=on, 0=off)  
\$1\_IN\_HIGHCUT\_FREQ :  
\$1\_IN\_HIGHCUT\_ON :  
\$1\_IN\_EQ\_LOW\_TRANS\_FREQ :  
\$1\_IN\_EQ\_LOW\_GAIN :  
\$1\_IN\_EQ\_MID2\_FREQ :  
\$1\_IN\_EQ\_MID2\_Q :  
\$1\_IN\_EQ\_MID2\_GAIN :  
\$1\_IN\_EQ\_MID1\_FREQ : (> MID2\_FREQ)  
\$1\_IN\_EQ\_MID1\_Q :  
\$1\_IN\_EQ\_MID1\_GAIN :  
\$1\_IN\_EQ\_HIGH\_TRANS\_FREQ :  
\$1\_IN\_EQ\_HIGH\_GAIN :

\$1\_IN\_AUX\_1\_PRE : Pre-Fader or Post-Fader of 1. AUX-SND (0 or 1) (default=0=post; 1=pre)

\$1\_IN\_AUX\_1\_GAIN : Gain of 1. AUX-SND (0 .. 150 MIDI-dB)

\$1\_IN\_FADER\_GAIN : Main-Gain of fader (0 .. 150 MIDI-dB)

\$1\_IN\_SPATIALIZATION : 0, 1,... (0==ambisonic, 1==direct)

\$1\_IN\_AMBI\_RAD :

\$1\_IN\_AMBI\_AZM : in degree

\$1\_IN\_AMBI\_ELV : in degree

\$1\_IN\_BUS\_1\_GAIN : Gain of 1.(1..16) BUS-SND (0 or 150 MIDI-dB)

\$1\_IN\_LFC\_GAIN : Gain of subwoofer(0 or 150 MIDI-dB)

====OUTPUT=====

Operating Parameter werden im Betrieb staendig nachgeladen und verursachen keine Knackser.  
(fuer MAIN-OUT-Gain)

-----

\$1\_OUT\_BUS\_GAIN: Direct-Output-Level fuer BUS1(1..16) (0 ... 150 MIDI-dB)

\$1\_OUT\_LFC\_GAIN: gesamt-lautstaerke fuer subwoofer-kanal 1(1..2)

\$1\_OUT\_LFC\_HIGHCUT\_FREQ: cutoff frequency fuer subwoofer-kanal 1(1..2)

OUT\_AMBI\_GAIN: gesamt-lautstaerke fuer ambisonic-decoder

OUT\_MASTER\_GAIN: master-gain (alles!)

====SUBOUT=====

=====

AMBI\_ORDER\_WEIGHT : 1 1 1 1 1 ...

bus-routing-matrix::

BUS\_ROUTE: matrix 16 52 .... (werte in MIDI-db)

element n m gain

\$1\_OUT\_LFC\_BUS\_DST: OUT-Destination fuer subwoofer-kanal 1(1..2)

(1\_OUT .. 52\_OUT)

Operating Parameter werden im Betrieb staendig nachgeladen und verursachen keine Knackser.  
(fuer mehrere Module)

===TALKBACK=====

\$1\_IN\_VU\_SRC: 0=input("adc in"), 1=pre, 2=post  
\$1\_IN\_VU\_LEVEL: dsp->gui talkback

VU\_BRIDGE\_SRC : <symbol>

Nach VU kann folgen:

- 0=IN\_ADC : routet VU-Meter zum Eingang nach Input-Gain;
  - 1=IN\_PRE : routet VU-Meter vor dem Fader, nach dem Effekt;
  - 2=IN\_POST : routet VU-Meter nach dem Fader;
  - 3=AUX\_SUM : routet VU-Meter nach dem Aux-Sum-Fader;
  - 4=AMBI\_SUM : routet VU-Meter nach dem Ambi-Gain und Decoder;
  - 5=BUS\_SUM : routet VU-Meter nach Main-Out-Gain;
  - 6=OUT\_DAC : routet VU-Meter unmittelbar vor dac~;
- dem VU-Meter-Werte-Paerchen wird noch der Kanal-Index vorgehaengt.

===GENERAL=====

DSP: 1/0 audio-engine on/off (pd dsp \$1)  
DSP\_CPU <mean> <peak>: cputime auf dem dsp

===CUE=====

CUE\_PARAMETERS\_LOAD: filename (symbol) das geladen werden soll  
CUE\_PARAMETERS\_SAVE: filename (symbol) in das "parameter" gespeichert werden  
CUE\_PARAMETERS\_RESET: setzt den CUE-speicher jedes parameters auf ""; wenn direkt danach die parameter gespeichert werden, ist das file leer! um "default"-werte zu laden sollte ein eigener "defaults"-CUE existieren!  
CUE\_SETTINGS\_LOAD: filename (symbol) das geladen werden soll  
CUE\_SETTINGS\_SAVE: filename (symbol) in das "setting" gespeichert werden  
CUE\_SETTINGS\_RESET: setzt den CUE-speicher jedes settings auf ""; wenn direkt danach die setting gespeichert werden, ist das file leer! um "default"-werte zu laden sollte ein eigener "defaults"-CUE existieren!

note: beim laden wird nicht zwischen "parameter" und "setting" unterschieden !

===EXTENSIONS=====

--SIMPLE\_REV-----

EXT\_SIMPLEREV\_ON: 0=off, 1=on

EXT\_SIMPLEREV\_PARAM: parameterliste... (see patch simplerevedit)

EXT\_SIMPLEREV\_SRC: AUX SUM Source eg: 1\_AUX

### **MIDI-Controller zur Fernsteuerung**

Die meisten Controller bilden Gruppen zu je 8 Parameter. Werden mehr als 8 Controller für eine Steuerelementgruppe gebraucht (z.b. 16 in-channels) wird der MIDI-kanal erhöht:

z.b.:

1\_IN\_FADER\_GAIN: 105, ch 1

2\_IN\_FADER\_GAIN: 106, ch 1

9\_IN\_FADER\_GAIN: 105, ch 2

MIDI-Werte:

IN\_FADER\_GAIN: 105-112

IN\_SOLO : 65-72

IN\_ON : 73-80

IN\_SELECT : 33-40

IN\_LFC\_GAIN : 1-8

IN\_AMBI\_AZM : 97-104

IN\_AMBI\_ELV : 89-96

IN\_AMBI\_RAD : 81-88

OUT\_BUS\_GAIN : 113-120

OUT\_AMBI\_GAIN : 121

OUT\_LFC\_GAIN : 122-123

OUT\_MASTER\_GAIN : 124

### **C. Belegung Fader- und Drehreglerboxen**

Für den CUBE wurden die MIDI-Controller Behringer BCF2000 und Bcr2000 und sind eigens für den 3D-Mixer umprogrammiert. Dazu wurden Presets erstellt. Die Presets in im Ordner Hardware abespeichert.

Preset-1: inchannels 1-8

Preset-2: inchannels 9-16  
Preset-3: inchannels 17-24  
Preset-4: inchannels 25-32  
Preset-5: BUS-OUT  
Preset-6: MASTER-OUT

#### **D. Anwendungsbeispiel im IEM CUBE**

Technische Daten:

I/OKanäle: 26 digitale (3x ADAT + AES/EBU oder SPDIF)

Kanalzüge: 16 mit Parametrischen EQ, 4 Plugins, Ambisonic oder Direkt, Volume, Subkanal

Ambisonic Master: 4.Ordnung

Busmaster: 12 Kanal Bus

Direkt: Matrix der Eingangskanäle

Plugins: 4 Plugins Standard + optionale Erweiterung

Master Sektions: Ambisonic mit Drehung und Gewichtung der Ordnungen.

Erweiterbarkeit: Plugins, Extensions, PD-Patches