Ambitools manual

for version 0.2

NTNU, Q2S

1 Introduction

1.1 General

Ambitools is a set of tools for real-time higher order Ambisonics (HOA) encoding and decoding with a graphical user interface. The real-time processing means that the programs operate on a number of live (mono) input signals. The programs use JACK [2] for audio input/output. The programs included are:

Ambienc Real-time Ambisonics encoder.

Ambidec Real-time Ambisonics decoder.

Ldc Real-time loudspeaker distance changer.

This manual does not include information on how to build and install Ambitools. For this information, see the `README` file bundled with the source code.

Familiarity with Ambisonics is assumed throughout this document.

1.2 JACK

JACK [2] is an audio server that provides a way to connect the audio inputs and outputs of programs together. All Ambitools programs start and auto-connect to the JACK server, and will not start if it is not running. Audio connections between programs and inputs/outputs must be done either manually with programs like Qjackctl or Patchage, or via the "Routing" menu (found in all Ambitools programs), which will attempt to auto-connect all applicable inputs/outputs to any desired location.

Each program will process the signals that are connected to its input ports and make the processed signals available on the output ports. If you wish to process a file, you will have to stream this file to the programs with a JACK compatible program capable of doing so.

Ambisonics components are transfered in the usual "WXYZRSTUVKLMNOPQ. . . " order.

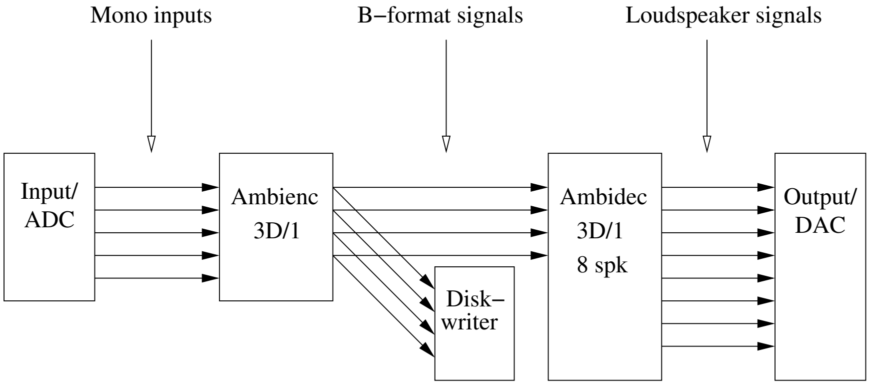


Figure 1: Usage example

1.2.1 Auto-routing

Thanks to the potentially large number of channels HOA might have, Ambitools has possibilites for auto-routing connections. The auto-routing dialog allows the user to specify a string describing target ports to connect to. This string consists of three parts, a program name, a colon delimiter, and the port base name. For example, specifying `ambidec:input_` in the "Auto-route outputs" dialog window of Ambienc would find the first instance of Ambidec, then connect the first output port of Ambienc to a port named `input_1`, the second output port to `input_2` and so on, until all source or target ports are connected.

The auto-routing dialog also has sensible default target strings to pick from.

1.3 Usage example

See figure 1 for an example block diagram of how to use Ambitools. This block diagram will look very much like what you'll see if you use a graphical JACK router program like Patchage. This usage scenario consists of three main parts:

1. Five external mono inputs are connected to Ambienc. These inputs might be anything, for example live microphone inputs or a CD player, and will all be treated as virtual source signals. Ambienc is set to do first order 3D encoding, a format that requires four output channels. All virtual sources are encoded according to the user's choices in Ambienc.

2. The four Ambisonics B-format output channels (W, X, Y and Z) are passed on to two programs: one disk-writer that will capture the Ambisonics signal and write it to disk, and one instance of Ambidec that will decode the Ambisonics signal for monitoring on a local loudspeaker rig.

3. Ambidec is set to do first order 3D decoding of the B-format signal to eight loudspeakers. The eight outputs of Ambidec are routed to the output ports of the sound device, which routes the signal to the loudspeakers via physical outputs.

Notes and tips

General

- The programs will sometimes write error messages on the console if they crash, so in cases of weird behaviour, it is recommended you run them from a console.
- The programs can both write and read their settings via menu entries in the “File” menu. These settings files are stored in plaintext and can be easily generated by other software, for example to generate long loudspeaker or virtual source position lists.

JACK

- Using HOA involves a lot of channels, and JACK might be set up to have a lower maximum amount of ports than you need. Adjust this setting to a higher number in the JACK configuration if you notice the programs exiting suddenly with a message saying there are too few ports.
- Since everything currently runs realtime, you should set the latency as low as you can in the JACK server if you want responsive audio. Note that not all Linux distributions are set up well for good low-latency use, so you might get glitches in the sound if latency is set too low. This is not Ambitools’ fault.

2 Ambienc

Ambienc encodes an arbitrary number of sound sources with full real-time positioning possibilities. Both distance attenuation and time delay based on distance (and hence also Doppler effect) can be modelled. Ambienc also supports near-field compensation (NFC) [1] filters.

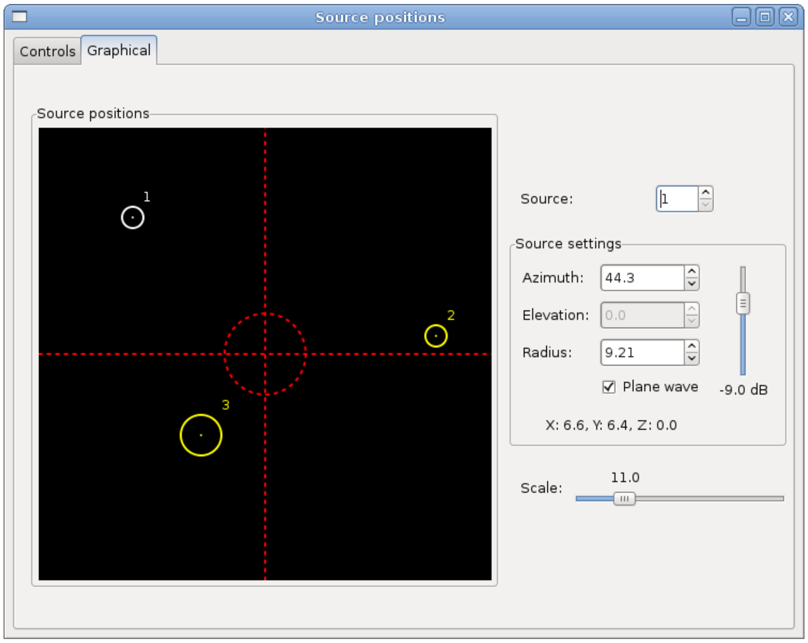
The main Ambienc window has the following adjustable parameters:

Sources Number of virtual sources to encode. This determines the number of input ports Ambienc will have. The current maximum is 128 sources.

Order Ambisonics order to use for encoding. Current maximum is 15th order for 2D Ambisonics and 5th order for 3D.

3D Chooses whether to use 2D or 3D Ambisonics. This does not only affect number of output channels, but also the scaling [1] of the channels which 2D and 3D Ambisonics encodings have in common.

Loudspeaker distance Distance of the loudspeaker array in meters. This needs to be specified for stability reasons when using NFC filters, and need not reflect the reproduction loudspeaker radius, since differing radiuses can be compensated for later (for example by using the Ldc program).



Source: 1

Azimuth: 44.3

Elevation: 0.0

Radius: 9.21

☑ Plane wave

-9.0 dB

X: 6.6, Y: 6.4, Z: 0.0

Scale: 11.0

Figure 2: Graphical source screen

2.1 Source control

Each sound source has a number of parameters belonging to it. These are:

Azimuth Determines the horizontal angle (in degrees) at which the source will be positioned relative to the front of the listener. Positive values equate to an anti-clockwise change in position.

Elevation Determines the elevation angle (in degrees) at which the source will be positioned relative to the horizontal listener plane. This parameter is only enabled for 3D Ambisonics.

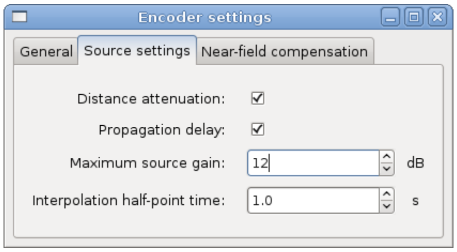
Radius Moves a source further from or closer to the listener position in the direction specified by azimuth and elevation parameters. The radius is specified in meters.

Gain Source gain in decibels.

Plane wave Sets source NFC filters to infinite radius and disables both gain and delay modelling. The radius parameter will have no effect while this switch is enabled.

Source movement resulting from manipulating these parameters will not be instantaneous, but smooth, using an interpolator. The position interpolation time can be changed in the settings dialog.

There are two source control screens where these attributes can be modified, one based on simple controls in the form of sliders and switches, and one graphical, in



Distance attenuation: ☑

Propagation delay: ☑

Maximum source gain: 12 dB

Interpolation half-point time: 1.0 s

Figure 3: Encoder settings

which you can position sources graphically in a coordinate system. See figure 2 for a screenshot of the latter.

The graphical source screen is a top-down view of the X-Y-plane where sources are represented by circles. The currently selected source will appear white, while all other sources are yellow. The circle representing a source will have a radius corresponding to its current height, with a bigger radius meaning it's higher up on the Z-axis. The loudspeaker array radius is represented by a dotted red circle.

The following mouse buttons are assigned actions:

Left mouse button Grab hold of a source and position it in the X-Y-plane. This will affect both azimuth and radius of the source. If 3D mode is enabled, it will also affect the elevation value.

Right mouse button Grab hold of a source and position it on the Z-axis. This will affect both elevation and radius of the source.

Middle mouse button Reset source position height to 0. Will also reset the source gain to 0 dB when used over the gain control.

In addition, there is also a scale slider, which will let the user adjust the scale of the coordinate system. Bigger values allow sources to be positioned further away from the listener position.

2.2 Settings

Ambienc has the following settings (see figure 3 for screenshot):

Output normalization

This setting can have the following values:

Semi-normalized Components are semi-normalized.

Normalized Components are normalized.

N2D compatible Components are scaled so that the 2D components of a 3D Ambisonics setup are compatible with the 2D "normalized" format, and can be stripped away and used separately for 2D-only decoding. See equation (22) in [1].

Furse-Malham Components are Furse-Malham format, meaning all component signal amplitudes are restrained to the ± 1.0 range, assuming no NFC filters are used. Furse-Malham normalization is only available up to 3rd order.

See [1] for details on normalization.

Distance attenuation

Disables the reciprocal distance to gain relationship for source modelling. If distance attenuation is enabled, a source will have unity gain when placed at the radius of the loudspeaker array.

Propagation delay

Disables the modelling of time delay (and hence Doppler effect) for sources.

Maximum source gain

Maximum gain for a source. Source gain will obey the reciprocal distance to gain relationship for larger radius values, but will be soft-clipped towards this value as the radius gets closer to zero. At which radius the gain starts being limited depends on the setting value.

Interpolation half-point time

The time at which a source's position has reached half-way to the target when moving. The interpolation curve is currently assymetric, so the total interpolation time is more than just double this setting.

The interpolator is currently a third order IIR filter designed to have no overshoot.

Near-field compensation

Enables NFC [1] filters to compensate for the near-field effect of both source and loudspeakers. The result of including this effect is a correct wavefront curvature being synthesized. Basic Ambisonics encoding (no NFC filters) will always yield spherical waves with origin at the loudspeaker array.

For sources placed within the loudspeaker array, an unmodified NFC filter will generate a very big bass boost, so Ambienc currently soft-limits the NFC filter radius to the loudspeaker array radius. Note that this limiting of the radius is only done for the NFC filters, not the rest of Ambienc. Eliminating this bass boost in other ways is a current research topic.

2.3 MIDI support

Ambienc supports controlling the source parameters by MIDI. This MIDI support is enabled by connecting any MIDI device to Ambienc's MIDI port using ALSA's sequencer interface, for example using Qjackctl, Aconnect or Patchage. The MIDI channel to parameter mapping is as follows:

Channel 0	Azimuth
Channel 1	Elevation
Channel 2	Radius
Channel 3	Plane wave switch
Channel 4	Gain

Controller numbers 0–31 will map to sources 1–32. Controller numbers 32–63 are the high-precision counterparts of controllers 0–31. Use these for LSB messages if your MIDI device supports 14 bit controller messages. Only the first 32 sources can be controlled by MIDI.

Notes and tips

- Ambienc uses more processing power for a source while it is being moved around. Quite a lot of sources can be used at the same time when they are stationary.
- 3D Ambisonics is as of yet quite untested!

3 Ambidec

Ambidec decodes Ambisonics signals to an array of arbitrarily placed loudspeakers. Supports loading a bank of FIR filters for spectral shaping of input Ambisonics components.

The main Ambidec window has the following adjustable parameters:

Speakers Number of loudspeakers to decode to. This determines the number of output ports Ambidec will have.

Order Ambisonics order to use for decoding. Current maximum is 15th order.

3D Determines whether to expect a 2D or 3D Ambisonics signal.

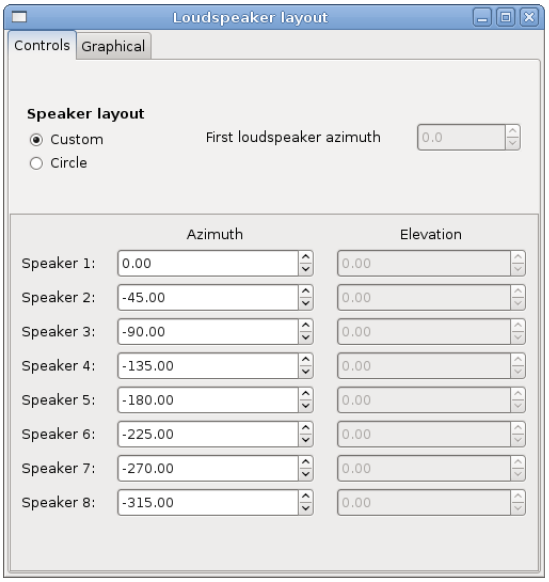
Gain Decoder gain control. Click with middle mouse button to reset to 0 dB.

3.1 Input filters

The “Load filters” entry in the “File” menu will allow the user to load file containing a bank of FIR filters. This is Ambidec’s way of supporting modified high-frequency decoding, usually using shelving filters with correcting gains.

Each filter bank file contains filters for a given order and loudspeaker number. When either of these are changed, the filters will unload themselves. The status bar at the bottom of the main window will show whether filters are currently in use or not.

Matlab code for generating shelving filters for modified high-frequency decoding is included in `matlab/` directory. Do `help createadf` in Matlab for details.



Speaker layout

● Custom

○ Circle

First loudspeaker azimuth 0.0

	Azimuth	Elevation
Speaker 1:	0.00	0.00
Speaker 2:	-45.00	0.00
Speaker 3:	-90.00	0.00
Speaker 4:	-135.00	0.00
Speaker 5:	-180.00	0.00
Speaker 6:	-225.00	0.00
Speaker 7:	-270.00	0.00
Speaker 8:	-315.00	0.00

Figure 4: Loudspeaker layout screen

3.2 Loudspeaker layout

This window allows the loudspeaker array to be set up. All loudspeakers are currently assumed to be at the same radius from the listener. See figure 4 for a screenshot of this window.

Currently, two layouts are supported:

Custom Allows you to set up your own loudspeaker array using the azimuth and elevation controls further down in the window.

Circle Sets up the loudspeakers in a clockwise circle, with the first loudspeaker being placed at the angle specified by the “First loudspeaker azimuth” input box.

The computed decoding matrix is Tikhonov-regularized [3] to assure a stable solution for non-regular loudspeaker layouts. The degree of regularization can be adjusted in the settings.

The loudspeaker layout window has two tabs at the top of the window: one named “controls” and one named “graphical”. The first allows you to position loudspeakers using numerical controls, and the second allows you to view your loudspeaker array in a top down view. This last also has basic positioning possibilities, but is currently meant to be a visualization tool.

3.3 Settings

Ambidec has the following settings:

Input normalization

Determines which format to expect input in. See Ambienc section for details.

Regularization parameter

Which beta factor to use in Tikhonov regularization of decoding matrix. 0 disables regularization, which might lead to unstable matrices for heavily irregular loudspeaker array setups, so use with care.

4 Ldc

Ldc converts the loudspeaker distance used in the input Ambisonics signal to another loudspeaker distance. Every NFC HOA encoded signal needs to be encoded with a reference loudspeaker array radius for low-frequency stability reasons, and this needs to be changed to the user's loudspeaker radius for correct reproduction.

The main Ldc window has the following adjustable parameters:

Order Ambisonics order to use. Current maximum is 15th order.

3D Determines whether to use 2D or 3D Ambisonics. For Ldc, this only affects the number of ports used.

Original distance Radius (in meters) of the loudspeaker array in the input Ambisonics signal.

Infinite Treat the input Ambisonics signal as having loudspeakers placed at infinity. This corresponds to basic encoding (no NFC filters).

New distance Radius (in meters) of the loudspeaker array in the processed output Ambisonics signal.

WARNING: if the new loudspeaker array radius is very much smaller than the old, extreme bass boosts may occur. Use with caution!

References

[1] Jérôme Daniel. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable new ambisonic format. In AES *23rd International Conference*, 2003.

[2] Paul Davis. JACK Audio Connection Kit. `http://www.jackaudio.org/`.

[3] Bård Støfringsdal and U. Peter Svensson. Conversion of discretely sampled sound field data to auralization formats. *Journal of the Audio Engineering Society*, 54(5), May 2006.