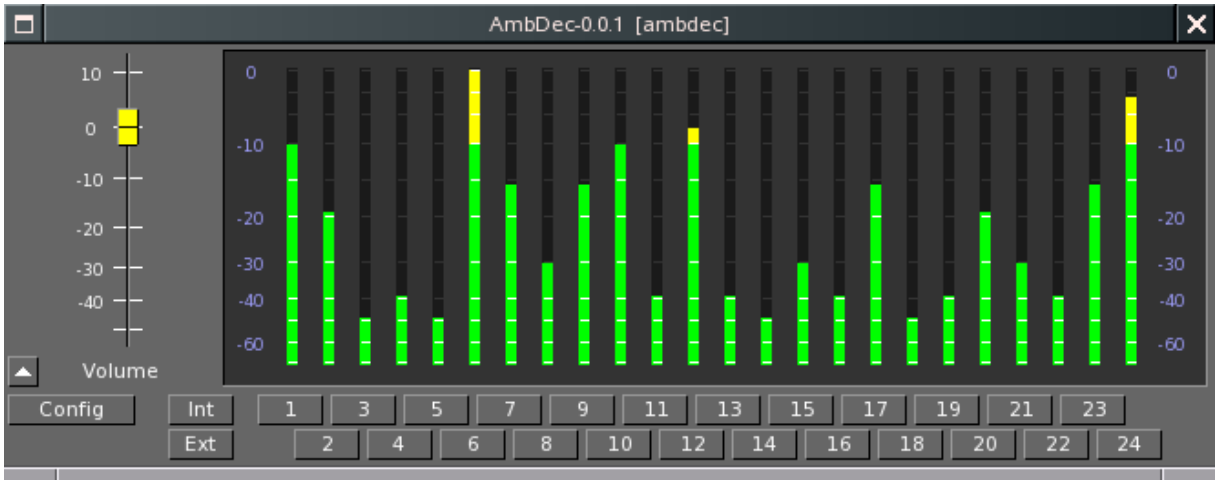
AmbDec - 0.2.0

User Manual

Fons Adriaensen
fons@kokkinizita.net



Contents

1 Introduction

This is the user manual for the 0.2.0 release of AmbDec.

AmbDec is an Ambisonics decoder for up to 24 speakers. It can be used for both horizontal and full 3-D systems of first and second order. The decoding matrices are fully user-configurable. AmbDec has some advanced feautures not found on most decoders:

- **Dual frequency band operation.** Optimal Ambisonics reproduction requires a different decoder for low and high frequencies. This is so mainly because human perception of sound direction uses different mechanisms at low and high frequencies.
  The crossover filters used in AmbDec are phase-aligned — the two outputs are exactly in phase at all frequencies. The combined response is equivalent to a first-order all-pass network, which is very probably inaudible.
- **Speaker distance compensation.** If the speakers are not all at the same distance to the center of the listening area, this must be compensated by including the right delays and gain corrections.
- **Near-field effect compensation.** Ambisonics works by reconstructing the spherical harmonic components of a sound field at the center of the listening area, using all speakers to synthesize each of them. All non-zero order components show a *near-field* effect depending on the ratio *wavelength*/*distance*, and increasing for rising order. One example of this is the well known 'proximity effect' of directional microphones ('directional' is equivalent to 'using non-zero order components'). Except in very large installations, where the speaker distances will be large enough so the listening area is not in the near field, this effect must be compensated for in order to obtain a correct reconstruction of a sound field.

All three features can be selectively enabled or disabled in the configuration.

1.1 Computing decoder matrices

AmbDec provides the audio processing, but does not have any functionality for computing the required matrices. For regular speaker layouts this calculation is relatively simple. 3-D layouts are almost never regular (this would require speaker below floor level in many cases), and the calculation of an optimal decoder gets much more complicated, in particular at medium and high frequencies.

A separate application is being developed to assist with the design of irregular and 3-D systems, but using this requires in-depth knowledge of the Ambisonics theory. The techniques used are interactive simulation and genetic search.

If you are planning to build an Ambisonic system, please get in contact with the author who will be happy to assist with the design of a matched AmbDec configuration. The 24 output limit can be modified quite easily if required.

2 Installing and running AmbDec

AmbDec is published under the GPL license. The full text of the license is provided in the file named COPYING which should be provided wich each copy of AmbDec.

2.1 Installing AmbDec from source

The source code for AmbDec is available from <http://www.kokkinizita.net/linuxaudio>. You also need some shared libraries: **libclthreads** version 2.4.0 or later, and **libclxclient** version 3.6.0 or later. Both can be downloaded from the same site. Apart from these, compiling AmbDec requires the JACK and X11 development files to be available.

Compiling and installing AmbDec should be simple enough. First ensure the two libraries are installed (compile and install **libclthreads** first as **libclxclient** depends on it). Unpack the AmbDec tarball, *cd* to the **source** directory, *make* and as root *make install*. The default install directory is **/usr/local/bin**. To change this, just edit the **PREFIX** variable in the Makefile. Copy the *presets* directory to some convenient place, and create an *$HOME/.ambdecrc* (see below).

2.2 Packaging and binary installation

Binary install packages should copy the *presets* directory to some place used for shared configuration files, e.g. */opt/ambdec/presets*, and install an */etc/ambdec.conf* pointing to the the shared presets directory. See the following section for details.

2.3 Running AmbDec

AmbDec reads options from */etc/ambdec.conf* or *$HOME/.ambdecrc*. If the latter file is present (even empty) then the global one is ignored. The option syntax is the one used by the X11 resource manager (as in *.Xdefaults*). The following can be used:

```
Ambdec.config:   path_to_default_configuration_file
Ambdec.presets:  path_to_presets_directory
```

The corresponding command line options are **-c** and **-p**. These will override those in either file. Using **-h** will print version info and list the command line options.

As any audio application AmbDec should be run with the privileges required to create real-time threads and to lock virtual memory. How this is done will depend on your system. Recent distros will require nothing special if the user is a member of the **audio** group. On others you may have to use either **rlimits** or **sudo**.

2.4 Input and output connections

AmbDec is designed as a professional application. It is therefore a JACK client only and it can not be used directly with ALSA, OSS, or any desktop specific sound server.

The number and names of both input and output ports will depend on the exact configuration. When a **new** preset is loaded and applied, all ports will be removed and a new set will be created. Just modifying the current configuration will not touch the JACK connections.

Output ports can be connected automatically if their destination is specified in the preset file. Input ports are never autoconnected.

As AmbDec could be used to drive systems capable of producing high sound levels, the audio code is designed so as to avoid clicks and pops when a preset is loaded or modified, by applying a short fade-out and fade-in around the change. Output ports are connected or disconnected only while silent. Also all audio controls are fully de-clicked or de-zippered.

3 The main window

The main window (fig.1) provides the master volume control, output level metering, and some test facilities. Clicking the Config button opens the configuration window which is discussed in the next section.

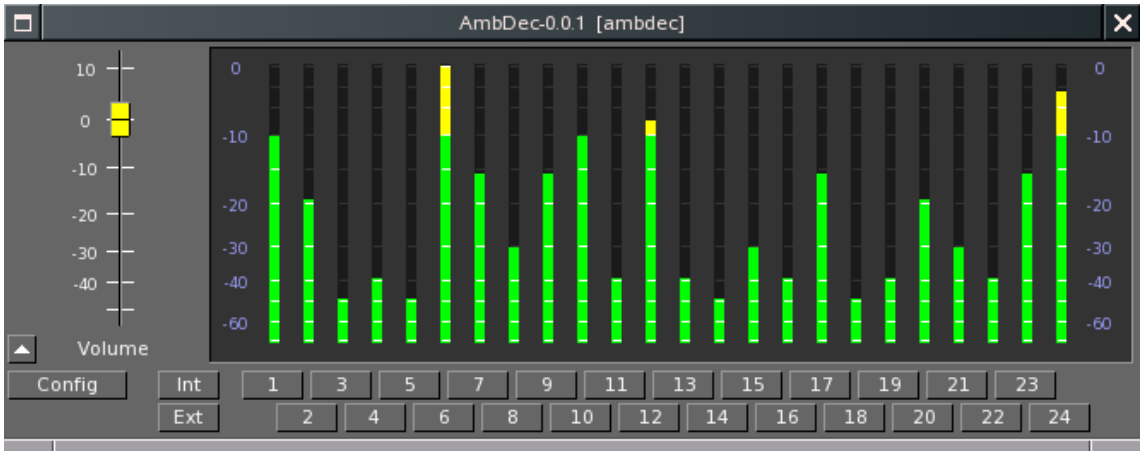


Figure 1: The main window

The buttons below each output level meter will take their labels from the speaker names in the preset file. They can be used in two ways:

- L-click on a button to **solo** the corresponding speaker. An R-click will add/remove a channel to/from the solo selection. Solo state is indicated by the button lighting up in green.
- Shift-click will **mute** the corresponding channel, indicated by the button turning orange. Shift-click again to unmute it.

The Int and Ext buttons will replace the output of the decoder matrices by either an internal or an external test signal. The test signals are sent only to channels selected for solo.

The internal signal (a phase-modulated sine) is used mainly to verify correct connections and levels. The external test input can be used for example when performing impulse response measurements on the speakers.

4 Configuration

The config window (fig.3, next page) is opened by clicking the Config button in the main window. Here you can create, load, modify and save AmbDec configurations. This can be done without any impact on the audio processing — the displayed configuration is transferred to the audio code only when the Apply button is clicked. When the window is opened the current configuration is shown. You can revert to this by using Cancel at any time.

4.1 Matrix coefficient and signal scaling

In order to correctly use some of the options discussed below, the following must be understood. Ambisonic signals are traditionally scaled in one of two different ways. For mathematical analysis it is very convenient to use the *normalised* form, meaning that each spherical harmonic has unity power when integrated over the sphere.
The alternative is the so-called *Furse-Malham* form. This applies some gain factors to the signals so that they all have the same *maximum* level. The only exception is the pressure signal W which, for historical reasons, is attenuated by 3 dB. Practical Ambisonic applications and audio files use the Furse-Malham convention.

AmbDec allows to use either representation, both for the input signals and for the matrix coefficients. If the two settings are different, AmbDec will automatically apply the necessary gain factors. So you can design a decoder matrix using the normalised form (which is usually less confusing, in particular for higher orders), and then use it with signals scaled according to the Furse-Malham convention.

Important note: the *normalised* form used is always the 3-D one, even for an horizontal-only decoder. The reason for this is that all decoders are used in 3-D space — unless your speakers are infinite vertical line sources the rules of physical 3-D space apply, even when analysing a 2-D decoder.

4.2 Creating a new configuration

Clicking the New button brings up the new configuration dialog (fig.2).

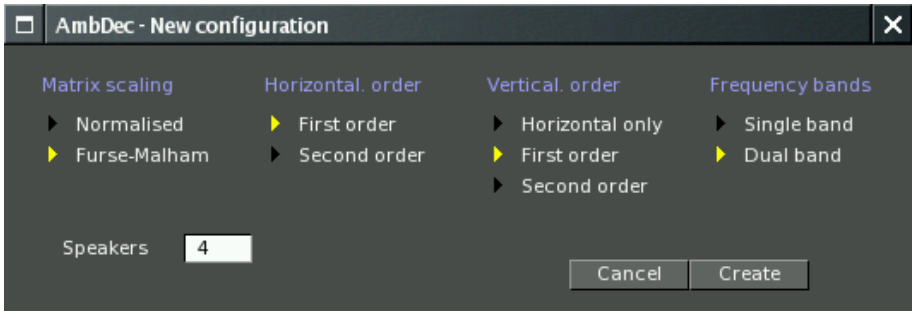


Figure 2: The cew configuration dialog

The choices made here are fixed for any configuration can not be modified later.

- **Matrix scaling.** This determines how the matrix coefficients will be interpreted, as explained in the previous section.
- **Horizontal order** Select either 1st or 2nd order.
- **Vertical order.** Select either a horizontal-only decoder, or 1st or 2nd order for the vertical component. You can use 2nd order horizontal with 1st order vertical (requiring six channels).
- **Frequency bands**. Select a single or dual band decoder. Generally much better results can be achieved with a dual band decoder.

Finally select the number of speakers to use, then click Create.

4.3 Run-time configuration options

Apart from the matrices which will be discussed in the next section, the configuration window (fig.3) provides some options that can be modified at any time.

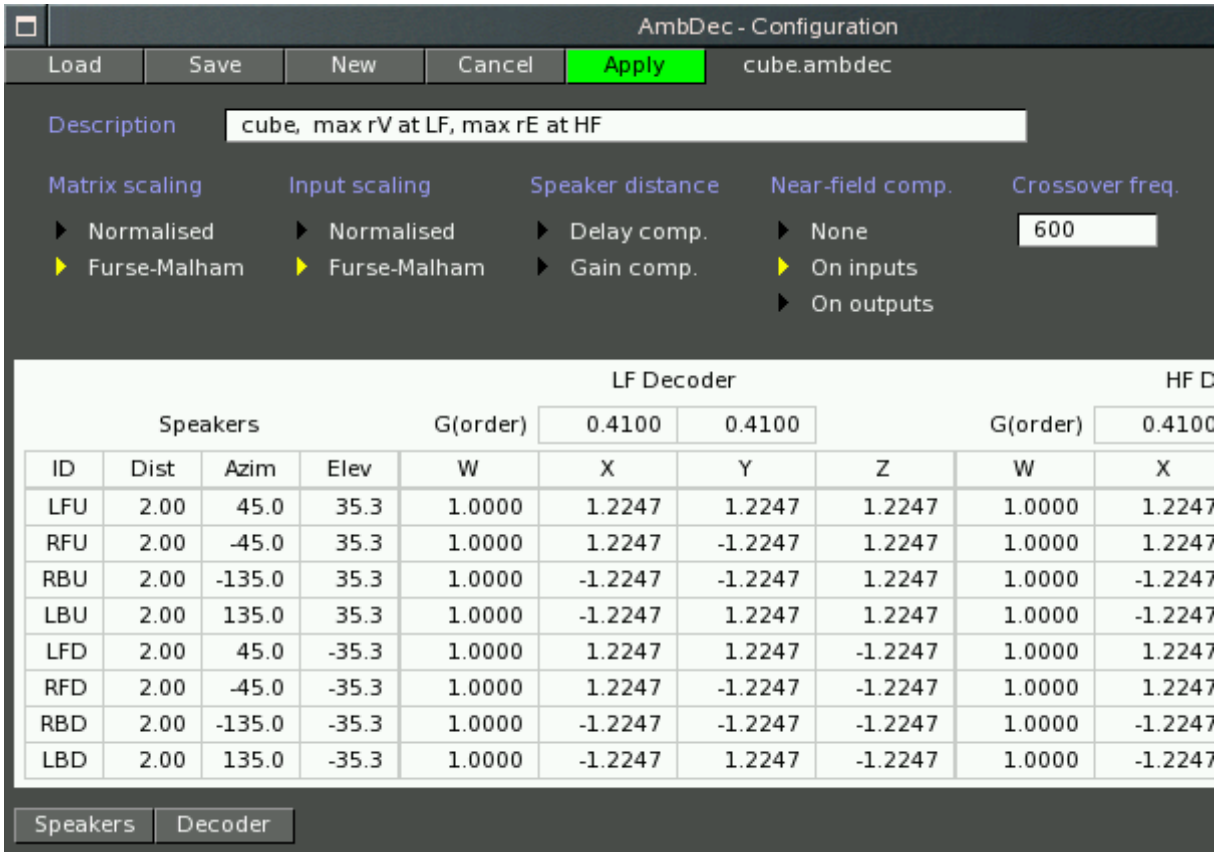


Figure 3: The configuration window, showing decoder matrices

- **Description.** This field can be used to add a short textual comment to the configuration file.

- **Matrix scaling.** This is shown for information only and can not be modified.
- **Input scaling.** Select the scaling of the input signals. This will be the Furse-Malham convention in almost all cases.
- **Delay compensation.** If selected, this will apply the correct delays to all outputs if your speakers are not all at the same distance to the center of the listening area. This option should be selected for all non-regular layouts, unless these delays are already included in for example an IR based speaker equalisation.
- **Gain compensation.** This will compensate for level differences resulting from unequal speaker distance. The output gains are calculated assuming the speakers are perfect point sources, which will in general not be the case. A better alternative is calibrate all speakers to the same SPL at the center using e.g. the amplifier gain controls, or to include the gain corrections in IR based equalisation.
- **Near-field compensation.** This compensates for level and phase errors on the non-zero order components resulting from the finite distance of the speakers. If all speakers are (approximately) at the same distance, you can select the On inputs option, else use On outputs which could take more CPU cycles. This option should be enabled for all but very large installations.
- **Crossover frequency.** This is shown for dual-band decoders only. The optimal value depends on the size of your setup, order, and some other factors. A value around 600 Hz will work fine in most cases.

4.4 The speaker matrix

The first four colums of the matrix define the speaker layout. Clicking Speakers in the lower left corner will add the JACK connections column (fig.4). If this is filled in the outputs will be automatically connected to their destination ports, usually the playback ports of a sound card.

Clicking Decoder in the lower left corner will return to the decoder matrices. There may be two such buttons if a dual band decoder is too big to show both matrices at the same time.

The **ID** column sets the names of the output channels. These can consist of up to three characters, and are used on the solo/mute buttons in the main window, and also as a suffix to the names of the JACK output ports.

Dist, Azim and **Elev** set the loudspeaker positions in spherical coordinates as seen from the center of the listening area. Azimuth is measured in the mathematical sense, i.e. positive values are counter-clockwise. Currently the software only uses the distance, but filling in the other coordinates may be required in the future and also helps with understanding or verifying the decoder matrices.

4.5 The decoder matrices

Apart from the basic matrix, there are also per-order gain factors labeled **G(order)**. These can often be used to modify the matrices in interesting ways without having to touch all the cells. For example, for a regular layout you can design a matrix for the $max\ rV$ criterion, and

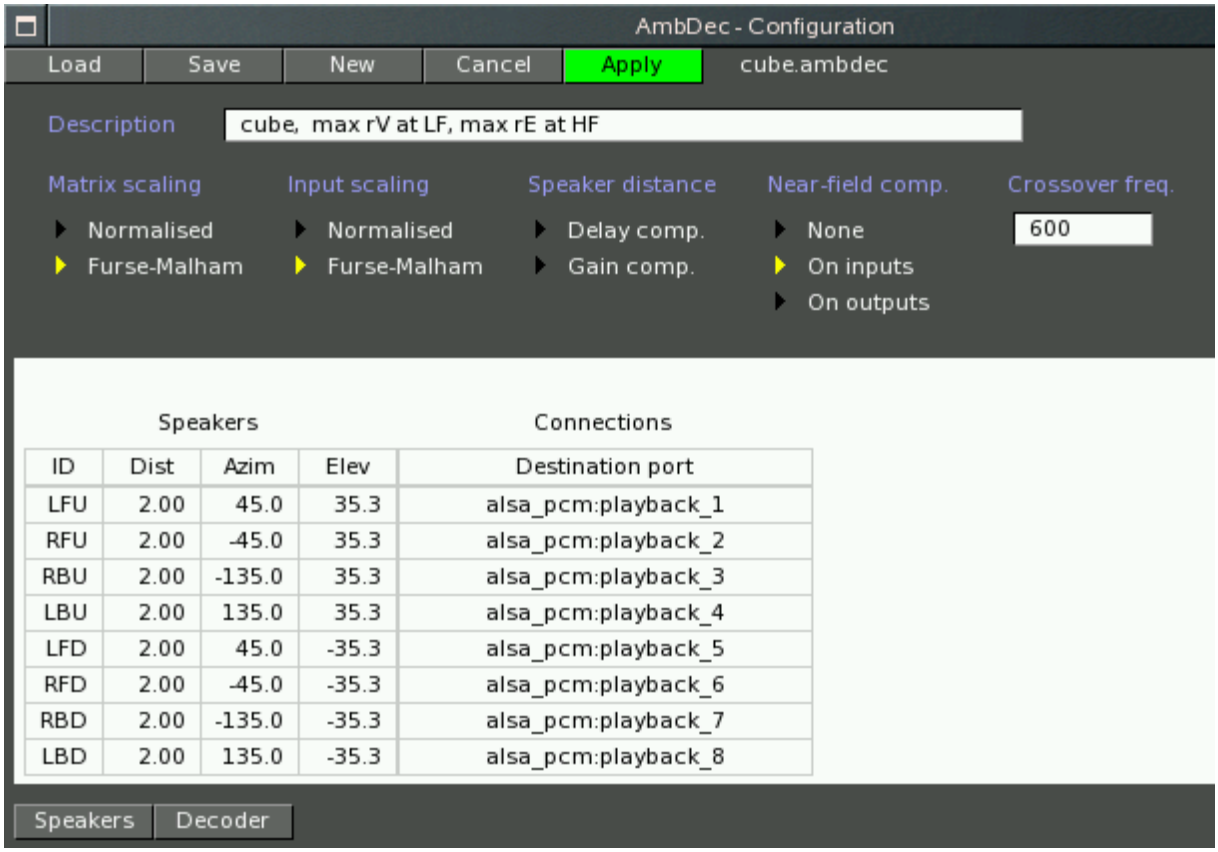


Figure 4: The configuration window, showing JACK connections

then turn it into a *max rE* or *in-phase* one by just modifying the gains for the non-zero order components. The cube decoder shown in fig.3 is an example of this: the LF and HF matrices are identical, only the gain of the first order components has been decreased in the HF part.

4.6 Some editing hints

The matrices can be navigated using Tab, Shift-Tab and the Up and Down arrows. The cells also understand some Emacs-style editing keys. To copy a value from the current cell to another, Shift-Click on the destination. To fill a column, set one cell, then use Shift-Up or Shift-Down to copy up or down. Modified cells are shown in a different color. This will be reset when you either Apply or Cancel the changes.

4.7 Preset files

The configurations are stored in preset files having the *.ambdec* extension. The format is a textual form of OSC, and will be clear enough from reading the examples provided.

The following presets are provided with this release.

- **square-1** A basic first order decoder for a square layout.

- **hexagon-1** First order decoder for a regular hexagon with two speakers in the front and back and one at each side.
- **hexagon-2** First order hexagon with one speaker in the front and back and two at each side.
- **cube** A basic first order 3-D decoder for a regular cube layout.
- **itu5.1-ord2-optim** A second order decoder for the ITU 5.1 surround layout. This one is the result of a genetic search. It includes a 3 dB front dominance.

All these decoders are designed for the *max rV* criterion at LF and for *max rE* at HF, which means they are optimised for a small 'sweet spot'. They can be modified easily by using the **G(order)** gains only.